

บทที่ 5

การพัฒนาโปรแกรมจัดการบนจอภาพให้กับโปรแกรมประยุกต์

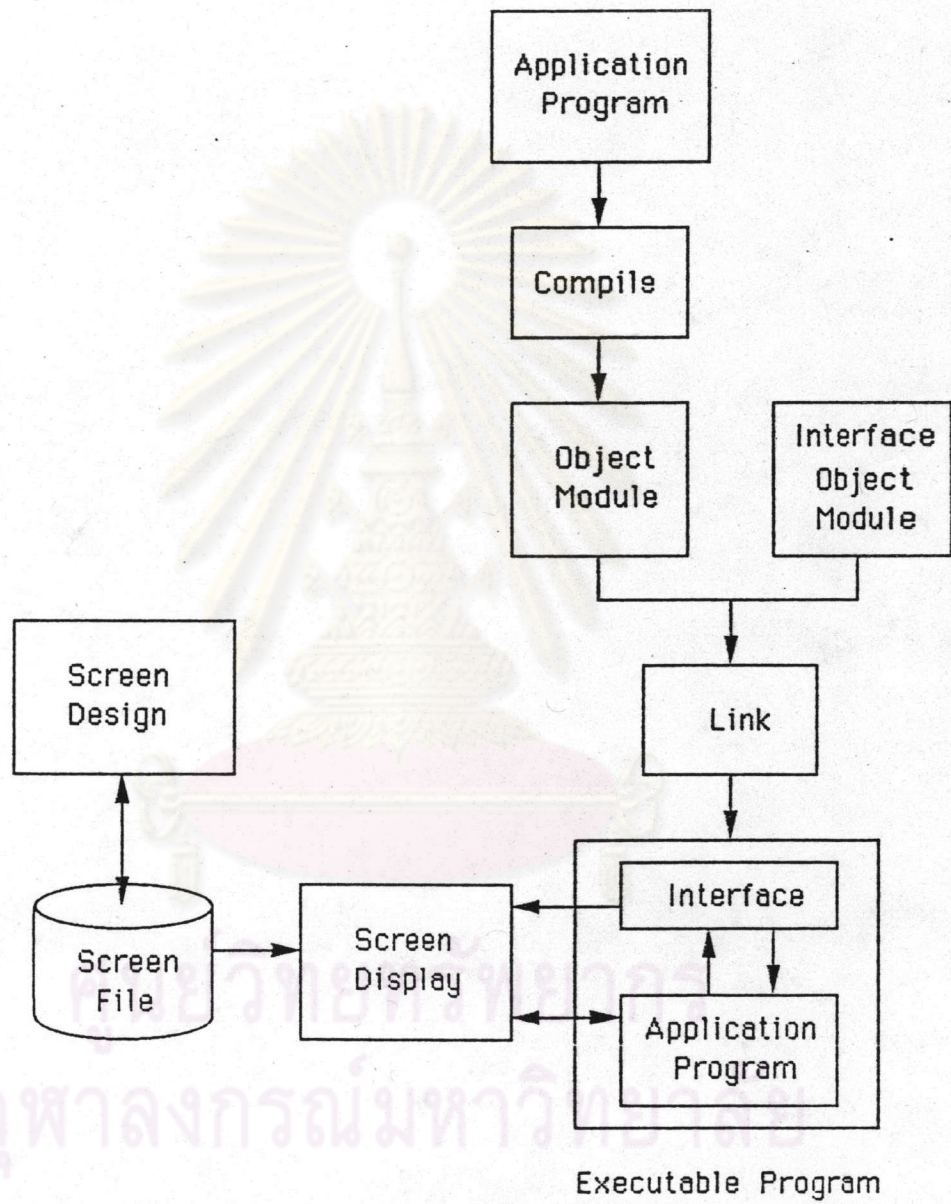
การพัฒนาโปรแกรมแบ่งเป็น 4 โปรแกรมดังนี้

1. โปรแกรมออกแบบจอภาพ
2. โปรแกรมออกรายงานรายละเอียดของแฟ้มข้อมูลของจอภาพ
3. โปรแกรมแสดงบนจอภาพ
4. โปรแกรมเชื่อมโยง

แต่ละโปรแกรมมีการทำงานที่สัมพันธ์กันดังนี้



ศูนย์วิจัยและพัฒนา
จุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 5.1 แสดงการทำงานของโปรแกรมจัดการหน้าจอภาพ

5.1 โปรแกรมออกแบบจอภาพ

เป็นโปรแกรมที่ใช้ในการสร้างหรือแก้ไขจอภาพที่จะแสดงตามผู้ใช้ต้องการ โดยให้ผู้ใช้กำหนดรายละเอียดดังนี้

5.1.1 ชื่อแฟ้มข้อมูลของจอภาพที่ต้องการสร้างหรือแก้ไข มีความยาวได้ไม่เกิน 8 ตัวอักษร

5.1.2 การกำหนดบนจอภาพ ได้แก่การกำหนดข้อความที่ต้องการให้ปรากฏบนจอภาพ ณ ตำแหน่งใด ๆ บนจอภาพ มีลักษณะการใช้งานเหมือนกับโปรแกรมบรรณาธิการจอภาพดังนี้

5.1.2.1 การเคลื่อนย้ายเคอร์เซอร์ไปยังตำแหน่งใด ๆ บนจอภาพ กำหนดให้ใช้ปุ่มบนแป้นพิมพ์ดังนี้

→ (ลูกศรชี้ทางขวา) ใช้เลื่อนเคอร์เซอร์ไปทางขวา 1 ตัวอักษร

← (ลูกศรชี้ทางซ้าย) ใช้เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตัวอักษร

↑ (ลูกศรชี้ขึ้น) ใช้เลื่อนเคอร์เซอร์ขึ้น 1 บรรทัด

↓ (ลูกศรชี้ลง) ใช้เลื่อนเคอร์เซอร์ลง 1 บรรทัด

Home ใช้เลื่อนเคอร์เซอร์ไปต้นบรรทัด

End ใช้เลื่อนเคอร์เซอร์ไปตัวอักษรสุดท้ายของบรรทัด

Ctrl + Home ใช้เลื่อนเคอร์เซอร์ไปตำแหน่งมุมบนซ้ายสุดของจอภาพ

Ctrl + End ใช้เลื่อนเคอร์เซอร์ไปตำแหน่งมุมล่างขวาสุดของจอภาพ

Return ใช้เลื่อนเคอร์เซอร์ไปยังต้นบรรทัดถัดไป

Alt + F9 ใช้เลื่อนเคอร์เซอร์ไปยังตำแหน่งที่มีการกดปุ่ม F9 ครั้งสุดท้าย

Alt + F10 ใช้เลื่อนเคอร์เซอร์ไปยังตำแหน่งที่มีการกดปุ่ม F10 ครั้งสุดท้าย

ท้าย

5.1.2.2 การกำหนดและแก้ไขข้อความ กำหนดให้ใช้ปุ่มบนแป้นพิมพ์ดังนี้

Backspace ใช้ลบตัวอักษรที่อยู่หน้าเคอร์เซอร์

Del ใช้ลบตัวอักษรที่เคอร์เซอร์อยู่แล้วเลื่อนตัวอักษรที่เหลือทางขวาเข้ามา

มา

Ins ใช้กำหนดสภาวะการป้อนข้อมูลให้อยู่ในสภาวะแทรก หรือยกเลิก

สภาวะแทรก

Alt + C ใช้ตัดลอกข้อความกลุ่มหนึ่งไปยังตำแหน่งใด ๆ บนจอภาพ

Alt + D ใช้ลบข้อความกลุ่มหนึ่งออกจากจอภาพ

Alt + M ใช้ย้ายข้อความกลุ่มหนึ่งไปยังตำแหน่งใด ๆ บนจอภาพ

Alt + B ใช้ติกรอบล้อมรอบบริเวณที่กำหนด

5.1.2.3 การกำหนดขนาดของจอภาพ กำหนดให้ใช้ปุ่มฟังก์ชันดังนี้

F5 ใช้กำหนดตำแหน่งมุมบนซ้ายสุดของจอภาพ

F6 ใช้กำหนดตำแหน่งมุมล่างขวาสุดของจอภาพ

5.1.3 การกำหนดรายละเอียดของเขตข้อมูล ได้แก่การกำหนดตำแหน่งของเขตข้อมูลบนจอภาพ รวมทั้งคุณสมบัติของเขตข้อมูลนั้น

5.1.3.1 การกำหนดคุณสมบัติของเขตข้อมูล กำหนดให้ใช้ปุ่มฟังก์ชันดังนี้

F9 ใช้กำหนดตำแหน่งเริ่มต้นของเขตข้อมูล

F10 ใช้กำหนดตำแหน่งสุดท้ายของเขตข้อมูล

F7 ใช้กำหนดคุณสมบัติของเขตข้อมูล หลังจากที่ถูกปุ่ม F9 และ F10 แล้ว ผู้ใช้สามารถกำหนดได้ดังนี้

1. ชื่อเขตข้อมูล (Field Name) มีความยาวได้ไม่เกิน 8 ตัวอักษร ชื่อเขตข้อมูลจะต้องตรงกับชื่อของตัวแปรในโปรแกรมประยุกต์ที่กำหนดให้เป็นเขตข้อมูลที่มีการส่งผ่านค่าระหว่างจอภาพกับ โปรแกรมประยุกต์

2. ลักษณะเฉพาะของการแสดง (Attribute) ได้แก่ แสดงแบบปกติ การเน้น สัญลักษณ์ภาพตรงข้าม การกระพริบ และขีดเส้นใต้

3. ประเภทของเขตข้อมูล (Type) ได้แก่ เขตข้อมูลนำเข้า เขตข้อมูลส่งออกและเขตข้อมูลข้อความ ในแต่ละประเภทสามารถกำหนดคุณสมบัติเพิ่มเติมได้ดังนี้

3.1 คุณสมบัติของเขตข้อมูลนำเข้า ประกอบด้วย

3.1.1 ประเภทของเขตข้อมูลที่ป้อน (Datatype)

เป็นการกำหนดการตรวจสอบตัวอักษรที่ป้อน ได้แก่

Alphabetic หมายถึงตัวอักษร A-Z

หรือ a-z

Alphanumeric หมายถึงตัวอักษร A-Z

หรือ a-z หรือตัวเลข 0-9

Integer หมายถึงตัวเลข 0-9 หรือ

เครื่องหมายบวก (+) หรือลบ (-)

Float หมายถึงตัวเลข 0 - 9 หรือ

เครื่องหมายบวก (+) หรือลบ (-) หรือจุดทศนิยม (.) หรือตัวอักษร E e D หรือ d

3.1.2 การจัดข้อความ (Justification)

เป็นการกำหนดลักษณะการป้อนข้อมูลจากตำแหน่งซ้ายสุด หรือขวาสุดของเขตข้อมูล

3.1.3 เขตข้อมูลที่ต้องป้อนข้อมูล (Input

Required) เป็นการกำหนดว่าเขตข้อมูลนั้นจะต้องมีการป้อนตัวอักษรอย่างน้อย 1 ตัวอักษร

- 3.1.4 เขตข้อมูลที่ต้องป้อนข้อมูลให้ครบทุกตัวอักษร (Must Fill) เป็นการกำหนดว่าเขตข้อมูลนั้นจะต้องมีการป้อนข้อมูลให้ครบทุกตัวอักษร
- 3.1.5 การเลื่อนเคอร์เซอร์ไปยังเขตข้อมูลถัดไป (Auto Skip) เป็นการกำหนดให้เคอร์เซอร์ถูกเลื่อนไปยังเขตข้อมูลถัดไปทันทีที่ป้อนข้อมูลในเขตข้อมูลปัจจุบันครบทุกตัวอักษร
- 3.1.6 การคัดลอกข้อมูล (Copy Value) เป็นการกำหนดให้คัดลอกค่าของเขตข้อมูลจากโปรแกรมประยุกต์ไปเก็บไว้ในบัฟเฟอร์ของการป้อนข้อมูลที่ป้อนก่อนการแสดงผลจอภาพ แล้วแสดงค่านี้นบนจอภาพด้วย
- 3.1.7 การไม่แสดงตัวอักษรที่ป้อน (No Echo) เป็นการกำหนดไม่ให้แสดงตัวอักษรที่ป้อนให้ปรากฏบนจอภาพ
- 3.1.8 การจบการแสดงผลจอภาพปัจจุบัน (Screen Exit) เป็นการกำหนดว่าถ้ามีการเลื่อนเคอร์เซอร์ออกจากเขตข้อมูลนั้นด้วยวิธีการใด ๆ ก็ตาม จะทำให้การทำงานของจอภาพที่กำลังแสดงอยู่จบลงทันที
- 3.1.9 การเปลี่ยนตัวอักษรเป็นตัวใหญ่ (Uppercase) เป็นการกำหนดให้เปลี่ยนตัวอักษรภาษาอังกฤษที่ป้อนจากตัวพิมพ์เล็กให้เป็นตัวพิมพ์ใหญ่
- 3.1.10 การเลื่อนดูข้อมูลที่ละบรรทัด (Scroll) เป็นการกำหนดให้สามารถเลื่อนเคอร์เซอร์เพื่อดูข้อมูลในเขตข้อมูลที่เป็นตัวแประะยะได้พร้อมกันหลายเขตข้อมูล โดยดูข้อมูลขึ้นลงได้ที่ละบรรทัด และที่ละหน้าพร้อมกัน
- 3.1.11 การป้องกันการป้อนข้อมูล (Protect) เป็นการกำหนดไม่ให้ป้อนข้อมูลในเขตข้อมูลนั้น
- 3.1.12 การตรวจสอบค่าข้อมูลภายในช่วงที่กำหนด เป็นการกำหนดให้ตรวจสอบค่าข้อมูลเลขจำนวนเต็มหรือเลขจำนวนจริงที่ป้อนว่าอยู่ภายในช่วงค่าต่ำสุดและค่าสูงสุดตามที่กำหนดหรือไม่
- 3.1.13 การตรวจสอบกับค่าที่เป็นไปได้ (Possible Check) เป็นการกำหนดให้ตรวจสอบค่าข้อมูลตัวอักษรที่ป้อนว่าอยู่ในกลุ่มของรายการของค่าที่เป็นไปได้ (Possible List) หรือไม่ การกำหนดรายการของค่าที่เป็นไปได้จะให้ผู้ใช้งานได้สูงสุด 100 ค่า แต่ละค่ายาวได้สูงสุด 60 ตัวอักษร
- 3.1.14 การกำหนดส่วนข้อความช่วยเหลือ เป็นการกำหนดข้อความที่จะแสดงในบรรทัดสุดท้ายของจอภาพ เพื่ออธิบายหรือขยายความเกี่ยวกับเขตข้อมูลนั้น ๆ ข้อความที่กำหนดจะยาวได้สูงสุด 80 ตัวอักษร

3.2 คุณสมบัติของเขตข้อมูลส่งออก ประกอบด้วยการจัดข้อความ ซึ่งเป็นการกำหนดลักษณะการแสดงผลให้อยู่ชิดซ้ายหรือขวาของเขตข้อมูล

3.3 คุณสมบัติของเขตข้อมูลข้อความ ประกอบด้วยการกำหนดเขตข้อมูลข้อความให้เป็นเมนูที่ถูกเลือกได้ (Selection Field) เป็นการกำหนดกลุ่มของเขตข้อมูลข้อความให้เป็นเมนูที่สามารถเลือกได้ โดยใช้วิธีการเลือกแบบการเลื่อนเคอร์เซอร์ไปยังรายการที่ต้องการ หรือการกดตัวอักษรตัวแรกของแต่ละรายการในเมนู การกำหนดกลุ่มของข้อความให้เป็นเมนูอาจให้อยู่ในแนวตั้งหรือแนวนอนก็ได้ คุณสมบัตินี้ผู้ใช้ต้องกำหนดรายละเอียดเพิ่มเติมดังนี้

3.3.1 ชื่อตัวแปรเลขจำนวนเต็มสำหรับเก็บสถานะของการเลือกเมนู (State Variable) มีค่าได้ตั้งแต่ 1, 2, 3, ... ซึ่งหมายถึงการเลือกเมนูในรายการที่ 1, 2, 3, ... ตามลำดับ ตัวแปรที่กำหนดนี้จะต้องถูกกำหนดในโปรแกรมประยุกต์ และถือเสมือนเป็นเขตข้อมูลหนึ่งในจอภาพที่มีการส่งผ่านค่าด้วย

3.3.2 ตัวอักษรตัวแรกของรายการ (First Character) เป็นการระบุตัวอักษรตัวแรกหรือตัวอักษรใด ๆ หนึ่งตัวอักษรที่ใช้แทนแต่ละรายการในเมนูที่จะให้ผู้ใช้กดเพื่อเลือกรายการนั้น ๆ

5.1.3.2 การแสดงขอบเขตของเขตข้อมูลทั้งหมดที่กำหนดบนจอภาพ ใช้ปุ่ม F8 โดยจะแสดงขอบเขตด้วยสัญลักษณ์พิเศษที่มีจำนวนเท่ากับความยาวของเขตข้อมูล และแสดงลักษณะเฉพาะของการแสดงด้วย

5.1.3.3 การแก้ไขเขตข้อมูลที่กำหนด ผู้ใช้ต้องเลื่อนเคอร์เซอร์ไปอยู่ในเขตข้อมูลที่ต้องการแก้ไข แล้วกด F7 เพื่อแสดงจอภาพของการกำหนดเขตข้อมูลพร้อมทั้งรายละเอียดที่กำหนดไว้ก่อนหน้านี้ หลังจากนั้นจึงแก้ไขส่วนใด ๆ ได้ตามต้องการ

5.1.3.4 การลบเขตข้อมูลที่กำหนด ผู้ใช้ต้องกระทำเช่นเดียวกับการแก้ไขข้อมูล แต่ให้ลบชื่อเขตข้อมูลออกโดยใช้ปุ่ม Spacebar แทนการแก้ไขส่วนใด ๆ

5.1.4 การกำหนดรายละเอียดของจอภาพ ประกอบด้วย

5.1.4.1 การกำหนดชื่อเขตข้อมูลที่จะให้เคอร์เซอร์ประจำอยู่ (Cursor Field) เมื่อมีการแสดงจอภาพของเขตข้อมูลนั้น

5.1.4.2 การกำหนดชื่อเขตข้อมูลส่งออกที่ให้ เป็นเขตข้อมูลสำหรับแสดงข่าวสาร (Message Field) จากโปรแกรมประยุกต์

5.1.4.3 การกำหนดชื่อแฟ้มข้อมูลของจอภาพที่จะถูกแสดงถัดไป (Next Screen) เมื่อจบการทำงานในจอภาพที่กำลังแสดงอยู่

5.1.4.4 การกำหนดชื่อแฟ้มข้อมูลของจอภาพที่จะถูกแสดงก่อนหน้า (Previous Screen) เมื่อจบการทำงานในจอภาพที่กำลังแสดงอยู่

เมื่อผู้ใช้กำหนดรายละเอียดตามความต้องการแล้ว โปรแกรมจะให้เก็บข้อมูลลงในแฟ้มข้อมูลภายใต้ชื่อที่กำหนดและมีส่วนขยายเป็น .SCN ในกรณีที่เป็นการแก้ไขจอภาพ โปรแกรมจะอ่านรายละเอียดของจอภาพที่เคยกำหนดไว้จากแฟ้มข้อมูลของจอภาพมาแสดงบนจอภาพ แล้วให้ผู้ใช้แก้ไขในแต่ละส่วนตามต้องการ

5.2 โปรแกรมออกรายงานรายละเอียดของแฟ้มข้อมูลของจอภาพ

การทำงานของโปรแกรมคือ จะอ่านรายละเอียดจากแฟ้มข้อมูลของจอภาพตามชื่อแฟ้มข้อมูลที่ระบุ แล้วพิมพ์รายงานออกทางจอภาพหรือเครื่องพิมพ์ตามที่ผู้ใช้เลือก โดยเรียงตามลำดับดังนี้

- 5.2.1 ข้อความที่ปรากฏบนจอภาพ จะมีข้อความและตำแหน่งเหมือนกับที่ปรากฏบนจอภาพ
- 5.2.2 รายละเอียดของจอภาพ ได้แก่ ชื่อแฟ้มข้อมูลของจอภาพ ขนาด จำนวนเขตข้อมูลในจอภาพ และคุณสมบัติอื่น ๆ
- 5.2.3 รายละเอียดของเขตข้อมูล ได้แก่ ชื่อเขตข้อมูล ความยาว ตำแหน่งบนจอภาพ และคุณสมบัติอื่น ๆ

5.3 โปรแกรมแสดงจอภาพ

เป็นโปรแกรมที่ใช้ในการจัดการจอภาพที่สร้างด้วยโปรแกรมออกแบบจอภาพให้กับโปรแกรมประยุกต์ที่เรียกใช้จอภาพ โดยในโปรแกรมประยุกต์จะบรรจุฟังก์ชันจัดการบนจอภาพไว้ให้โปรแกรมประยุกต์เรียกใช้ ประกอบด้วยการทำงานที่สำคัญคือ

1. การติดตั้งโปรแกรมในหน่วยความจำ
2. การบรรจุและปฏิบัติการโปรแกรม
3. ฟังก์ชันเริ่มต้นการทำงานหลังจากถูกเรียกใช้จากโปรแกรมประยุกต์
4. การจบการทำงานของโปรแกรม
5. ฟังก์ชันจัดการบนจอภาพ

มีรายละเอียดดังนี้

5.3.1 การติดตั้งโปรแกรมในหน่วยความจำ

เป็นการบรรจุโปรแกรมแสดงจอภาพให้ติดตั้งในหน่วยความจำ ประกอบด้วยขั้นตอนการทำงานดังนี้

5.3.1.1 การบรรจุโปรแกรมและกำหนดค่า เริ่มต้นจากพารามิเตอร์ที่ส่งมา
กับการประมวลผล โปรแกรมที่บรรทัดคำสั่ง มีรูปแบบของการกำหนดดังนี้

[drive:] SCDP progname [/Ssize] [/Msize] [/Psize] [/Nnumber]

- drive หมายถึง หน่วยขั้วงานแม่เหล็ก เช่น A: B:
- SCDP หมายถึง ชื่อ โปรแกรมแสดงจอภาพ
- progname หมายถึง ชื่อ โปรแกรมประยุกต์ที่ เรียก ใช้เพิ่มข้อมูลของจอภาพ
มีส่วนขยายเป็น .EXE และจะต้องระบุส่วนขยายนี้ตามหลัง
ชื่อ โปรแกรมด้วย
- /Ssize หมายถึงการกำหนดเนื้อที่ของหน่วยความจำ ให้เป็นบัฟเฟอร์
สำหรับเก็บรายละเอียดของจอภาพและ เขตข้อมูลของจอภาพที่
กำลังแสดงอยู่ รวมถึงบัฟเฟอร์สำหรับเก็บข้อมูลที่ป้อนเข้ามา
ของเขตข้อมูลนำเข้าด้วย size หมายถึงจำนวนกิโล ไบต์
ที่กำหนดให้กับบัฟเฟอร์ มีขนาดได้ตั้งแต่ 2-30 กิโล ไบต์
- /Msize หมายถึงการกำหนดเนื้อที่ของหน่วยความจำ ให้เป็นบัฟเฟอร์
สำหรับเก็บเพิ่มข้อมูลของจอภาพไว้ในหน่วยความจำตลอดการ
ทำงานของโปรแกรมประยุกต์ size หมายถึงจำนวนกิโล
ไบต์ที่กำหนดให้กับบัฟเฟอร์ มีขนาดได้ตั้งแต่ 0-30 กิโล ไบต์
- /Psize หมายถึงการกำหนดเนื้อที่ของหน่วยความจำ ให้เป็นบัฟเฟอร์
สำหรับเก็บรายละเอียดของตัวแปรใน โปรแกรมประยุกต์ที่
สัมพันธ์กับเขตข้อมูลในจอภาพ size หมายถึงจำนวนกิโล ไบต์
ที่กำหนดให้กับบัฟเฟอร์ มีขนาดได้ตั้งแต่ 2-30 กิโล ไบต์
- /Nnumber หมายถึงการกำหนดจำนวนเพิ่มข้อมูลของจอภาพที่จะถูกเก็บให้
อยู่ในหน่วยความจำตลอดการทำงาน ของโปรแกรมประยุกต์
number หมายถึงจำนวนเพิ่มข้อมูล กำหนด ได้ตั้งแต่ 0-99

5.3.1.2 การติดตั้งบัฟเฟอร์ โปรแกรมจะจองเนื้อที่ในหน่วยความจำให้
กับบัฟเฟอร์ตามขนาดที่กำหนดในแต่ละพารามิเตอร์

5.3.1.3 การกำหนดตำแหน่ง เริ่มต้นการทำงานของฟังก์ชันจัดการบนจอภาพ
เป็นการหาตำแหน่ง เริ่มต้นทำงานเมื่อ โปรแกรมประยุกต์ เรียก ใช้ฟังก์ชัน แล้วบรรจุค่า เชกเมนต์
และออฟเซตของตำแหน่งนั้นลงในตารางอินเทอร์รับต์ เวคเตอร์หมายเลข 60h โดย ใช้ฟังก์ชันของ
ระบบหมายเลข 25h

5.3.1.4 การกำหนดเส้นทางสำหรับค้นหาโปรแกรมประยุกต์และเพิ่มข้อมูลของจอภาพ เป็นการกำหนดไดเรคทอรีที่จะให้โปรแกรมแสดงจอภาพค้นหาโปรแกรมประยุกต์หรือเพิ่มข้อมูลของจอภาพที่เก็บอยู่ในไดเรคทอรีต่าง ๆ โดยใช้คำสั่ง SET ในเอ็มเอส-ดอสขณะอยู่ที่บรรทัดคำสั่งก่อนปฏิบัติการโปรแกรมประยุกต์

5.3.2 การบรรจุและปฏิบัติการโปรแกรม

โปรแกรมแสดงจอภาพจะบรรจุและปฏิบัติการโปรแกรมประยุกต์ตามชื่อที่ระบุในบรรทัดคำสั่ง โดยเรียกใช้ฟังก์ชันของระบบหมายเลข 4Bh ประกอบด้วยการทำงานดังนี้

5.3.2.1 กำหนดขนาดของหน่วยความจำของโปรแกรมแสดงจอภาพใหม่ให้มีขนาดเท่ากับที่จำเป็นต้องใช้จริง โดยใช้ฟังก์ชันของระบบหมายเลข 4Ah

5.3.2.2 กำหนดบล็อกของพารามิเตอร์เพื่อปฏิบัติการ และกำหนดชื่อของโปรแกรมประยุกต์ให้ถูกต้องตามข้อกำหนดของฟังก์ชันของระบบ

5.3.2.3 เก็บบันทึกสถานะเดิมของรีจิสเตอร์ทุกตัวของโปรแกรมแสดงจอภาพไว้ในสแตคของโปรแกรมนั้น รวมทั้งเก็บค่าสแตกเชกเมนต์และตัวชี้สแตกขณะนั้นไว้ในเนื้อที่ส่วนหนึ่งที่กำหนดอยู่ในเชกเมนต์ของโปรแกรม

5.3.2.4 บรรจุและปฏิบัติการโปรแกรมประยุกต์ และส่งการทำงานให้กับโปรแกรมประยุกต์

5.3.2.5 ในระหว่างที่โปรแกรมประยุกต์ทำงาน โปรแกรมสามารถเรียกใช้ฟังก์ชันการทำงานของโปรแกรมแสดงจอภาพได้ โดยเรียกผ่านโปรแกรมเชื่อมโยงของโปรแกรมแต่ละภาษา หลังจากนั้นจึงส่งการทำงานจากโปรแกรมประยุกต์ไปให้ฟังก์ชันจัดการบนจอภาพทำงาน

5.3.3 ฟังก์ชันเริ่มต้นการทำงานหลังจากถูกเรียกใช้จากโปรแกรมประยุกต์

เป็นฟังก์ชันซึ่งรับการทำงานมาจากโปรแกรมประยุกต์ ประกอบด้วยการทำงาน

5.3.3.1 เปลี่ยนแปลงการใช้สแตกของโปรแกรมประยุกต์มาใช้สแตกของโปรแกรมแสดงจอภาพ โดยใช้ค่าสแตกเชกเมนต์และตัวชี้สแตกที่เก็บไว้ในเชกเมนต์ของโปรแกรมแสดงจอภาพ

5.3.3.2 เก็บบันทึกสถานะเดิมของรีจิสเตอร์ทุกตัวของโปรแกรมประยุกต์

5.3.3.3 ตรวจสอบพารามิเตอร์ที่ส่งมาจากโปรแกรมประยุกต์ว่าเป็นฟังก์ชันการทำงานใด

5.3.3.4 ส่งการทำงานพร้อมกับพารามิเตอร์ไปให้กับฟังก์ชันนั้น ๆ หลังจากฟังก์ชันทำงานเสร็จก็จะคืนสถานะเดิมของรีจิสเตอร์ทุกตัว พร้อมทั้งเปลี่ยนแปลงการใช้สแตกกลับมาใช้สแตกของโปรแกรมประยุกต์ตามเดิม แล้วส่งการทำงานกลับคืนให้กับโปรแกรมประยุกต์

5.3.4 การจบการทำงานของโปรแกรม

5.3.4.1 เมื่อโปรแกรมประยุกต์ทำงานเสร็จ ก็จะคืนสภาวะเดิมของรีจิสเตอร์ทุกตัว รวมทั้งเปลี่ยนแปลงการใช้สแตกมาใช้สแตกของโปรแกรมแสดงจอภาพ แล้วส่งการทำงานกลับคืนให้กับโปรแกรมแสดงจอภาพ

5.3.4.2 โปรแกรมแสดงจอภาพจะคืนเนื้อที่หน่วยความจำทั้งหมดที่จองไว้เป็นบัฟเฟอร์ต่าง ๆ ให้กับหน่วยความจำ แล้วจบการทำงานของโปรแกรมด้วยการดึงเอาโปรแกรมออกจากหน่วยความจำและคืนการทำงานให้กับระบบต่อไป

5.3.5 ฟังก์ชันจัดการบนจอภาพ

เป็นฟังก์ชันซึ่งถูกเรียกใช้จากโปรแกรมประยุกต์เพื่อทำหน้าที่เกี่ยวกับการแสดงและจัดการจอภาพ โดยเรียกผ่านโปรแกรมเชื่อมโยง พร้อมทั้งส่งค่าพารามิเตอร์ต่าง ๆ ให้กับแต่ละฟังก์ชัน ประกอบด้วยฟังก์ชันต่อไปนี้

5.3.5.1 ฟังก์ชันแสดงจอภาพ เป็นฟังก์ชันซึ่งใช้ในการแสดงข้อความและเขตข้อมูลที่กำหนดในแฟ้มข้อมูลของจอภาพออกทางจอภาพ พร้อมทั้งควบคุมการทำงานของบนจอภาพ การติดต่อระหว่างผู้ใช้กับจอภาพ และระหว่างโปรแกรมประยุกต์กับจอภาพที่แสดง

5.3.5.1.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน DISPLAY

ชื่อแฟ้มข้อมูลของจอภาพ หรือไม่ระบุก็ได้

5.3.5.1.2 การทำงานของฟังก์ชัน

ก) อ่านรายละเอียดของจอภาพที่กำหนดจากแฟ้มข้อมูลของจอภาพ แล้วแสดงเนื้อหาข้อความ (ถ้ามี) บนจอภาพ พร้อมทั้งเก็บรายละเอียดของจอภาพและเขตข้อมูล (ถ้ามี) ไว้ในบัฟเฟอร์ของรายละเอียดของจอภาพที่กำลังแสดง โดยแยกเก็บเป็นบัฟเฟอร์ของจอภาพ คุณสมบัติของเขตข้อมูลนำเข้า เขตข้อมูลส่งออก และเขตข้อมูลข้อความในการอ่านรายละเอียดของแฟ้มข้อมูล ฟังก์ชันจะต้องตรวจสอบก่อนว่าแฟ้มข้อมูลนั้นถูกเก็บไว้ในบัฟเฟอร์สำหรับเก็บแฟ้มข้อมูลไว้ในหน่วยความจำหรือไม่ ถ้ามีก็จะอ่านรายละเอียดของแฟ้มข้อมูลนั้นจากบัฟเฟอร์ แต่ถ้าไม่มีจะอ่านแฟ้มข้อมูลจากสื่อบันทึกข้อมูลภายนอก

ข) ตรวจสอบว่าในจอภาพนั้นมีการกำหนดเขตข้อมูลนำเข้าหรือไม่ ถ้ามีก็จะมีภารกิจของเนื้อหาส่วนหนึ่งของบัฟเฟอร์ของรายละเอียดของจอภาพที่กำลังแสดงเพื่อใช้เป็นบัฟเฟอร์สำหรับเก็บบันทึกค่าข้อมูลที่ผู้ใช้ป้อนเข้ามาในเขตข้อมูลนั้น

ค) จัดการเขตข้อมูลต่าง ๆ บนจอภาพ (ถ้ามี) โดยตรวจสอบคุณสมบัติของเขตข้อมูลนั้นจากบัฟเฟอร์ของจอภาพที่กำลังแสดง การจัดการเขตข้อมูลส่งออกจะเป็นการอ่านค่าของเขตข้อมูลจากตัวแปรในโปรแกรมประยุกต์ที่สัมพันธ์กับเขตข้อมูลนั้นแล้วแสดง

ค่าข้อมูลบนจอภาพตามคุณสมบัติที่กำหนด ในการเข้าถึงค่าของเขตข้อมูล ฟังก์ชันจะตรวจสอบหาตำแหน่งของตัวแปรที่เก็บข้อมูลนั้น ได้จากบัพเฟอร์ที่เก็บตัวแปรทั้งหมด แล้วอ่านค่าข้อมูล ณ ตำแหน่งนั้นตามคุณสมบัติของตัวแปรที่กำหนดไว้

ส่วนการจัดการเขตข้อมูลนำเข้าและเขตข้อมูลข้อความที่กำหนดเป็นเมนู จะเป็นการรอรับการป้อนข้อมูลหรือการเลือกเมนูจากผู้ใช้ พร้อมทั้งมีการตรวจสอบและจัดการกับเขตข้อมูลที่ป้อนให้เป็นไปตามคุณสมบัติที่กำหนดดังที่ได้กล่าวไปแล้ว เช่น การตรวจสอบตัวอักษรที่ป้อน การตรวจสอบข้อมูลที่ป้อนว่าอยู่ในช่วงค่าต่ำสุดหรือสูงสุดที่กำหนดหรือไม่ โดยมีลำดับการป้อนข้อมูลและการเลือกเมนูจากซ้ายไปขวาและจากบนลงล่างตามลำดับ เริ่มตั้งแต่ตำแหน่งมุมบนซ้ายสุดของจอภาพไปเรื่อย ๆ

ง) เมื่อผู้ใช้งานต้องการจบการทำงานของฟังก์ชันนี้จะกระทำได้โดยการกดปุ่ม F2 F3 หรือ Esc บนแป้นพิมพ์ การกดปุ่ม F2 หรือ F3 จะทำให้มีการส่งค่าข้อมูลนำเข้าจากบัพเฟอร์ และผลลัพธ์จากการเลือกเมนูไปเก็บในตัวแปรในโปรแกรมประยุกต์ที่สัมพันธ์กับเขตข้อมูลบนจอภาพนั้น ส่วนการกดปุ่ม Esc จะไม่มีการส่งค่าไปยังเขตข้อมูลใด ๆ หลังจากนั้นจึงจบการแสดงผลจอภาพด้วยการส่งการทำงานกลับคืนไปยังโปรแกรมประยุกต์ พร้อมทั้งส่งค่ารหัสแสดงความผิดพลาดกลับไปด้วย โดยไม่มีการลบล้างค่าต่าง ๆ ที่อยู่ในบัพเฟอร์ และไม่มีการลบข้อมูลบนจอภาพด้วย

ในการส่งผ่านค่าข้อมูลฟังก์ชันจะต้องตรวจสอบหาตำแหน่งและคุณสมบัติของตัวแปรจากบัพเฟอร์ของตัวแปรทั้งหมด และตรวจสอบภาษาคอมพิวเตอร์ของโปรแกรมประยุกต์ก่อน แล้วจึงส่งค่าข้อมูลจากบัพเฟอร์ไปเก็บในตัวแปรตามคุณสมบัติที่ตรวจสอบ ทั้งนี้เนื่องจากการเก็บค่าข้อมูลแต่ละประเภทในหน่วยความจำ เช่น ตัวอักษร เลขจำนวนเต็ม มีวิธีการเก็บไม่เหมือนกัน และการเก็บค่าข้อมูลบางประเภท เช่น ตัวอักษรในภาษาคอมพิวเตอร์ แต่ละภาษาก็มีวิธีการเก็บไม่เหมือนกันอีกด้วย [5,6,7]

จ) ในกรณีที่โปรแกรมประยุกต์เรียกใช้ฟังก์ชัน DISPLAY โดยไม่ระบุชื่อแฟ้มข้อมูลของจอภาพ หมายถึง ให้แสดงรายละเอียดของจอภาพเดิมที่ถูกแสดงก่อนหน้านี้ด้วยฟังก์ชัน DISPLAY และฟังก์ชันจะทำหน้าที่ เชื่อมโยงกับผู้ใช้และโปรแกรม โดยอ้างอิงถึงคุณสมบัติของจอภาพและเขตข้อมูลจากค่าที่ยังคงอยู่ในบัพเฟอร์

ฉ) ถ้าในบัพเฟอร์ต่าง ๆ ไม่มีที่ว่างพอที่จะเก็บรายละเอียดของจอภาพที่กำลังแสดง หรือไม่มีแฟ้มข้อมูลของจอภาพอยู่ ฟังก์ชันจะส่งรหัสความผิดพลาด (return code) ที่ไม่เท่ากับ 0 กลับไป

ช) การเคลื่อนย้ายเคอร์เซอร์และการควบคุมจอภาพเมื่อมีการแสดงจอภาพ ในฟังก์ชันได้กำหนดการใช้ปุ่มต่าง ๆ บนแป้นพิมพ์ให้ผู้ใช้ เลือกควบคุมการทำงานบนจอภาพและการเคลื่อนย้ายเคอร์เซอร์ดังนี้

7.1 การเคลื่อนย้ายเคอร์เซอร์ภายในเขตข้อมูล

- ใช้เลื่อนเคอร์เซอร์ไปทางขวา 1 ตัวอักษร
- ← ใช้เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตัวอักษร
- ↓ ใช้เลื่อนจอภาพลงทีละ 1 บรรทัด สำหรับเขต

ข้อมูลที่เป็นตัวแประะเรย์

- ↑ ใช้เลื่อนจอภาพขึ้นทีละ 1 บรรทัด สำหรับเขต

ข้อมูลที่เป็นตัวแประะเรย์

- PgDn ใช้เลื่อนจอภาพลงทีละ 1 จอภาพ สำหรับ

เขตข้อมูลที่เป็นตัวแประะเรย์

- PgUp ใช้เลื่อนจอภาพขึ้นทีละ 1 จอภาพ สำหรับ

เขตข้อมูลที่เป็นตัวแประะเรย์

- Home ใช้เลื่อนเคอร์เซอร์ไปตำแหน่งแรกของเขต

ข้อมูล

- End ใช้เลื่อนเคอร์เซอร์ไปตำแหน่งสุดท้ายของ

เขตข้อมูล

7.2 การเคลื่อนย้ายเคอร์เซอร์ระหว่างเขตข้อมูล

- ↑ , Shift + Tab ใช้เลื่อนเคอร์เซอร์ไปยังเขต

ข้อมูลก่อนหน้า

- ↓ , Tab , Return ใช้เลื่อนเคอร์เซอร์ไปยัง

เขตข้อมูลถัดไป

7.3 การป้อนและแก้ไขข้อมูลในเขตข้อมูลนำเข้า

- Backspace ใช้ลบตัวอักษรที่อยู่หน้าเคอร์เซอร์

- Del ใช้ลบตัวอักษรที่มีเคอร์เซอร์อยู่ แล้วเลื่อนตัว

อักษรที่เหลือเข้ามา

- Ins ใช้กำหนดสถานะการป้อนข้อมูลให้อยู่ในสถานะ

แทรกหรือยกเลิกสถานะแทรก

7.4 การจบการแสดงผลจอภาพ

- F2 ใช้จบการแสดงผลจอภาพ แล้วแสดงผลจอภาพถัดไป

ถ้ามีการกำหนด ในกรณีนี้จะยังไม่ส่งการทำงานกลับคืนไปยังโปรแกรมประยุกต์

- F3 ใช้จบการแสดงผลจอภาพ แล้วแสดงผลจอภาพก่อน

หน้าถ้ามีการกำหนด ซึ่งในกรณีนี้จะยังไม่ส่งการทำงานกลับคืนไปยังโปรแกรมประยุกต์

Esc ใช้จบการแสดงผล โดยไม่ส่งค่าข้อมูลนำ
เข้าไปเก็บในตัวแปรในโปรแกรมประยุกต์

7.5 การแสดงส่วนขอความช่วยเหลือสำหรับชุดข้อมูล
ใช้ปุ่มฟังก์ชัน F1

7.6 ตัวแปรแสดงสาเหตุของการจบการแสดงผล
เป็นตัวแปรซึ่งโปรแกรมประยุกต์สามารถใช้ตรวจสอบได้ว่า ผู้ใช้จบการแสดงผลด้วยการกดปุ่ม
ใดบนแป้นพิมพ์ โดยในโปรแกรมประยุกต์จะต้องกำหนดตัวแปรนี้ไว้ในส่วนการแสดงผลรู้จัก
เช่นเดียวกับตัวแปรที่กำหนดเป็นชุดข้อมูลที่มีการส่งผ่านค่าระหว่างโปรแกรมประยุกต์กับจอภาพ
ตัวแปรที่ใช้แสดงสาเหตุมี 2 ตัวแปรได้แก่

7.6.1 ตัวแปร ZKBD เป็นตัวแปรเลขจำนวน
เต็มขนาด 2 ไบต์ ใช้เก็บรหัสสแกน (scan code) และรหัสแอสกี (ASCII code) ของปุ่มที่
ใช้จบการทำงานของจอภาพ

7.6.2 ตัวแปร ZEXT เป็นตัวแปรตัวอักษร
ขนาด 3 ไบต์ ใช้เก็บคำสั่งที่ใช้จบการแสดงผลดังนี้
คำสั่ง END หมายถึงการใช้ปุ่ม F2 หรือ F3
ในการจบการแสดงผล

คำสั่ง ESC หมายถึงการใช้ปุ่ม Esc ในการ
จบการแสดงผล

5.3.5.2 ฟังก์ชันกำหนดตำแหน่งของเคอร์เซอร์ เป็นฟังก์ชันซึ่งใช้ในการ
บังคับให้เคอร์เซอร์เลื่อนไปอยู่ในเขตข้อมูลนำเข้าที่กำหนดในจอภาพที่จะถูกแสดงต่อไปด้วยฟังก์ชัน
แสดงผลจอภาพ (DISPLAY) โดยเขตข้อมูลนำเข้านั้นจะต้องเป็นเขตข้อมูลที่ถูกกำหนดในจอภาพนั้นด้วย

5.3.5.2.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน CURSOR

ชื่อเขตข้อมูลนำเข้า

5.3.5.2.2 การทำงาน จะกำหนดแฟล็กเพื่อให้ฟังก์ชันแสดง
จอภาพรู้ว่ามีการเรียกใช้ฟังก์ชัน CURSOR มาก่อน

5.3.5.3 ฟังก์ชันกำหนดให้แสดงข่าวสาร ใช้ในการแสดงผลข้อมูลซึ่งเก็บอยู่
ในเขตข้อมูลข่าวสาร (Message Field) จากโปรแกรมประยุกต์ ในจอภาพที่จะถูกแสดงต่อไป
ด้วยฟังก์ชันแสดงผลจอภาพ (DISPLAY) โดยเขตข้อมูลข่าวสารนั้นจะต้องถูกกำหนดอยู่ในจอภาพนั้นด้วย

5.3.5.3.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน SENDMSG

5.3.5.3.2 การทำงาน จะกำหนดแฟลกเพื่อแสดงให้ฟังก์ชัน แสดงจอภาพรู้ว่าการเรียกใช้ฟังก์ชัน SENDMSG มาก่อน

5.3.5.4 ฟังก์ชันเสียงเตือน

5.3.5.4.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน ALARM

5.3.5.4.2 การทำงาน ทำให้เครื่องคอมพิวเตอร์ส่งเสียง ด้วยการส่งตัวอักษรที่มีค่า 07h ออกทางจอภาพ

5.3.5.5 ฟังก์ชันลบทั้งจอภาพ

5.3.5.5.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน CLEAR

5.3.5.5.2 การทำงาน ทำให้ข้อมูลทั้งจอภาพถูกลบด้วยการ ล้างจอภาพให้เป็นช่องว่าง (space) ทั้งหมด

5.3.5.6 ฟังก์ชันกำหนดแฉีกข้อมูลของจอภาพไว้ในหน่วยความจำ ใช้เก็บ บันทึกรายละเอียดของแฉีกข้อมูลของจอภาพที่กำหนดไว้ในบัฟเฟอร์ตลอดการทำงานของ โปรแกรมประยุกต์ หรือจนกว่าจะมีการยกเลิกการกำหนดแฉีกข้อมูลนั้นในหน่วยความจำ หลังจาก ใช้ฟังก์ชันนี้ เมื่อโปรแกรมประยุกต์เรียกใช้ฟังก์ชันการแสดงผลจอภาพ ฟังก์ชันนี้จะอ่านรายละเอียด ของแฉีกข้อมูลของจอภาพจากบัฟเฟอร์ที่กำหนดแทนการอ่านจากสื่อบันทึกข้อมูลภายนอก

5.3.5.6.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน SBUFFER

ชื่อแฉีกข้อมูลของจอภาพ

5.3.5.6.2 การทำงาน

ก) หาขนาดของแฉีกข้อมูลของจอภาพ

ข) ตรวจสอบที่ว่างในบัฟเฟอร์สำหรับเก็บแฉีกข้อมูลของจอภาพ ว่ามีเนื้อที่เหลือพอที่จะเก็บรายละเอียดของแฉีกข้อมูลที่กำหนดหรือไม่ และตรวจสอบจำนวนแฉีก ข้อมูลที่เก็บในบัฟเฟอร์ด้วยว่าเกินค่าที่กำหนดไว้หรือไม่

ค) ถ้าในบัฟเฟอร์มีที่ว่างเหลือพอและจำนวนแฉีกข้อมูลที่เก็บ อยู่ในบัฟเฟอร์ไม่เกินค่าที่กำหนด ก็จะอ่านรายละเอียดของแฉีกข้อมูลจากสื่อบันทึกข้อมูลเข้ามาเก็บ

ไว้ในบัพเฟอร์

ง) ถ้าไม่สามารถเก็บบันทึกรายละเอียดของแฟ้มข้อมูลได้จะส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับไป

5.3.5.7 ฟังก์ชันลบแฟ้มข้อมูลของจอภาพออกจากหน่วยความจำ ใช้ลบรายละเอียดทั้งหมดของทุกแฟ้มข้อมูลของจอภาพที่เก็บอยู่ในบัพเฟอร์ ถ้าโปรแกรมประยุกต์ต้องการเรียกใช้จอภาพที่เคยเก็บไว้ในบัพเฟอร์ ฟังก์ชันแสดงจอภาพจะไม่สามารถอ่านรายละเอียดของแฟ้มข้อมูลของจอภาพจากบัพเฟอร์ได้อีก

5.3.5.7.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน SRESET

5.3.5.7.2 การทำงาน จะลบล้างข้อมูลทั้งหมดในบัพเฟอร์ที่ใช้เก็บรายละเอียดของจอภาพ

5.3.5.8 ฟังก์ชันกำหนดตัวแปรเดี่ยว เป็นฟังก์ชันที่ใช้กำหนดรายละเอียดของตัวแปรเดี่ยวในโปรแกรมประยุกต์ที่มีชื่อตรงกับชื่อเขตข้อมูลบนจอภาพ เพื่อให้โปรแกรมแสดงจอภาพรู้จักและสามารถเข้าถึงข้อมูลของตัวแปรนั้นได้ เมื่อมีการกำหนดตัวแปรรายละเอียดต่าง ๆ จะถูกเก็บบันทึกไว้ในบัพเฟอร์ของตัวแปรทั้งหมดตลอดการทำงานของโปรแกรม หรือจนกว่าจะมีการยกเลิกตัวแปรนั้นในบัพเฟอร์

5.3.5.8.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน VDEFINE

ชื่อตัวแปร

ประเภทของข้อมูลที่เก็บในหน่วยความจำ ได้แก่

C หมายถึงตัวอักษร

I หมายถึงเลขจำนวนเต็ม

FO - F9 หมายถึงเลขจำนวนจริงที่มีเลขหลังจุดทศนิยม

0-9 ตำแหน่งตามลำดับ

ตำแหน่งที่อยู่ของตัวแปรในโปรแกรมประยุกต์ ได้แก่ ค่าเซกเมนต์และออฟเซตของตำแหน่ง

ความยาวที่ใช้เก็บค่าของตัวแปรในหน่วยความจำ คือ

เลขจำนวนเต็มแบบ short ใช้เนื้อที่ 2 ไบต์

เลขจำนวนเต็มแบบ long ใช้เนื้อที่ 4 ไบต์

เลขจำนวนจริงแบบ single precision ใช้เนื้อที่ 4
 ไบต์
 เลขจำนวนจริงแบบ double precision ใช้เนื้อที่ 8
 ไบต์
 ตัวอักษร มีความยาวเท่ากับ ความยาวของตัวแปรที่กำหนด
 ใน โปรแกรมประยุกต์

5.3.5.8.2 การทำงาน

- ก) คำนวณหาตำแหน่งในบัพเฟออร์ที่จะใช้เก็บตัวแปรที่กำหนด โดยใช้ชื่อตัวแปร เป็นคีย์ในการคำนวณหาตำแหน่ง
- ข) ตรวจสอบตำแหน่งที่คำนวณ ได้ว่ามีที่ว่างหรือไม่ ถ้าว่าง จะเก็บบันทึกรายละเอียดของตัวแปรลงในบัพเฟออร์ในตำแหน่งนั้น แล้วกำหนดแฟล็กใหม่เพื่อแสดง สถานภาพว่าไม่ว่าง แต่ถ้าตำแหน่งนั้นไม่ว่างก็จะค้นหาตำแหน่งถัดไปที่ว่างที่ละตำแหน่งจนกว่าจะพบ
- ค) ถ้าในบัพเฟออร์ไม่มีที่ว่างที่จะเก็บตัวแปรใด ๆ ได้อีกก็จะ ส่งรหัสความผิดพลาดที่ไม่เท่ากับ 0 กลับไป

5.3.5.9 ฟังก์ชันกำหนดตัวแปรอะเรย์ เป็นฟังก์ชันที่ใช้ในการกำหนด ตัวแปรอะเรย์ เช่นเดียวกับฟังก์ชันการกำหนดตัวแปรเดี่ยว

5.3.5.9.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน VDEFARR

ชื่อตัวแปรอะเรย์

ประเภทของข้อมูลที่เก็บตัวแปรในหน่วยความจำ มีรายละเอียด เช่นเดียวกับฟังก์ชันกำหนดตัวแปรเดี่ยว

ตำแหน่งที่อยู่ของตัวแปรใน โปรแกรมประยุกต์

ความยาวที่ใช้เก็บค่าของตัวแปรในหน่วยความจำ มีรายละเอียด เช่นเดียวกับฟังก์ชันกำหนดตัวแปรเดี่ยว

จำนวนสมาชิกของตัวแปรอะเรย์

5.3.5.9.2 การทำงาน เหมือนกับฟังก์ชันกำหนดตัวแปรเดี่ยว
 ทุกประการ

5.3.5.10 ฟังก์ชันลบตัวแปรออกจากบัพเฟออร์ ใช้ลบได้ทั้งตัวแปรเดี่ยวและ ตัวแปรอะเรย์ออกจากบัพเฟออร์ของชื่อตัวแปรทั้งหมด

5.3.5.10.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน VDELETE

ชื่อตัวแปร

5.3.5.10.2 การทำงาน จะค้นหาตำแหน่งของตัวแปรที่กำหนดไว้แล้วในบัพเฟอร์ โดยใช้ชื่อตัวแปรเป็นคีย์ในการคำนวณ แล้วกำหนดแฟล็กแสดงสถานภาพใหม่ให้เป็นหนึ่ง พร้อมทั้งลบรายละเอียดของตัวแปรออกจากบัพเฟอร์

5.3.5.11 ฟังก์ชันลบทุกตัวแปรออกจากบัพเฟอร์ ใช้ลบตัวแปรทุกตัวทั้งตัวแปรเดี่ยวและตัวแปรอะเรย์ออกจากบัพเฟอร์ของตัวแปร

5.3.5.11.1 พารามิเตอร์ที่ใช้

ชื่อฟังก์ชัน VRESET

5.3.5.11.2 การทำงาน จะกำหนดแฟล็กแสดงสถานภาพให้เป็นหนึ่งในทุกตำแหน่งในบัพเฟอร์ และลบรายละเอียดของทุกตัวแปรออกจากบัพเฟอร์ด้วย

5.4 โปรแกรมเชื่อมโยง

5.4.1 ลักษณะของโปรแกรมเชื่อมโยง เป็นโปรแกรมซึ่งทำหน้าที่เป็นตัวกลางเชื่อมโยงโปรแกรมประยุกต์ภาษาใด ๆ ให้สามารถเรียกใช้และส่งพารามิเตอร์ให้กับฟังก์ชันจัดการบนจอภาพได้ ในการปฏิบัติการโปรแกรมประยุกต์จะต้องรวมโปรแกรมเชื่อมโยงเข้าเป็นส่วนหนึ่งของโปรแกรมประยุกต์ ด้วยการเชื่อมโยงออบเจคโมดูลของโปรแกรมประยุกต์เข้ากับออบเจคโมดูลของโปรแกรมเชื่อมโยง ภายในโปรแกรมเชื่อมโยงแบ่งออกเป็น 3 โปรแกรมย่อยคือ

5.4.1.1 โปรแกรมย่อยสำหรับใช้กับฟังก์ชันทั่วไป เช่น DISPLAY ALARM จะต้องส่งผ่านพารามิเตอร์ดังนี้

- ก) ชื่อฟังก์ชันและพารามิเตอร์ของฟังก์ชัน
- ข) ความยาวของชื่อฟังก์ชันและพารามิเตอร์รวมกัน
- ค) ตำแหน่งที่อยู่ของตัวแปรรับรหัสความผิดพลาดในโปรแกรม

ประยุกต์

5.4.1.2 โปรแกรมย่อยสำหรับใช้กับฟังก์ชันกำหนดตัวแปรเดี่ยว คือ VDEFINE จะต้องส่งผ่านพารามิเตอร์ดังนี้

- ก) ชื่อฟังก์ชันและพารามิเตอร์ของฟังก์ชัน
- ข) ความยาวของชื่อฟังก์ชันและพารามิเตอร์รวมกัน
- ค) ตำแหน่งที่อยู่ของตัวแปรรับรหัสความผิดพลาดในโปรแกรม

ประยุกต์

- ง) ตำแหน่งที่อยู่ของตัวแปรเดี่ยวในโปรแกรมประยุกต์
- จ) ความยาวของเนื้อหาในหน่วยความจำที่ใช้เก็บค่าข้อมูลของ

ตัวแปรเดี่ยว

5.4.1.3 โปรแกรมย่อยสำหรับใช้กับฟังก์ชันกำหนดตัวแปรอะเรย์ คือ VDEFARR จะต้องส่งผ่านพารามิเตอร์ดังนี้

- ก) ชื่อฟังก์ชันและพารามิเตอร์ของฟังก์ชัน
- ข) ความยาวของชื่อฟังก์ชันและพารามิเตอร์รวมกัน
- ค) ตำแหน่งที่อยู่ของตัวแปรรับรหัสความผิดพลาดในโปรแกรม

ประยุกต์

- ง) ตำแหน่งที่อยู่ของตัวแปรอะเรย์ในโปรแกรมประยุกต์
- จ) ความยาวของเนื้อหาในหน่วยความจำที่ใช้เก็บค่าข้อมูลของ

ตัวแปรอะเรย์

- ฉ) จำนวนสมาชิกของตัวแปรอะเรย์

5.4.2 การทำงานของโปรแกรมเชื่อมโยง ในการทำงานของโปรแกรมเชื่อมโยง โปรแกรมจะต้องเตรียมเนื้อหาที่หน่วยความจำส่วนหนึ่งให้เป็นบล็อกข้อมูล (data block) สำหรับเก็บพารามิเตอร์ที่ส่งมาจากโปรแกรมประยุกต์ เพื่อให้ฟังก์ชันจัดการจอภาพสามารถอ้างถึงพารามิเตอร์แต่ละตัวได้จากบล็อกข้อมูลนี้ บล็อกข้อมูลที่มีความยาว 32 ไบต์ ภายในประกอบด้วยข้อมูลตามลำดับดังนี้

- ก) รหัสแทนภาษาของโปรแกรมประยุกต์ ได้แก่ 1 แทนโปรแกรมภาษาซี 2 แทนโปรแกรมภาษาฟอร์แทรนและ 3 แทนโปรแกรมภาษาปาสคาล ใช้เนื้อหาในบล็อกข้อมูล 2 ไบต์
- ข) รหัสแทนประเภทของโปรแกรมย่อยที่ใช้ ได้แก่ 1 แทนโปรแกรมย่อยที่ใช้กับฟังก์ชันทั่วไป 2 แทนโปรแกรมย่อยที่ใช้กับฟังก์ชันกำหนดตัวแปรเดี่ยวและ 3 แทนโปรแกรมย่อยที่ใช้กับฟังก์ชันกำหนดตัวแปรอะเรย์ ใช้เนื้อหาในบล็อกข้อมูล 2 ไบต์
- ค) ชื่อฟังก์ชันการจัดการบนจอภาพและพารามิเตอร์ มีความยาวเท่ากับ 20 ไบต์

ง) ตำแหน่งอ้างอิงตัวแปรในโปรแกรมประยุกต์ในลักษณะฟาร์พอยน์เตอร์ (far pointer) คือประกอบด้วยค่าเซกเมนต์และออฟเซตของตำแหน่งของตัวแปร พารามิเตอร์นี้จะใช้กับโปรแกรมย่อยกำหนดตัวแปรเดี่ยวและตัวแปรอะเรย์ และใช้เนื้อหาในบล็อกข้อมูล 4 ไบต์

- จ) จำนวนไบต์ที่ใช้ในการเก็บค่าข้อมูลของตัวแปรในหน่วยความจำ พารามิเตอร์นี้จะใช้กับโปรแกรมย่อยกำหนดตัวแปรเดี่ยวและตัวแปรอะเรย์ และใช้เนื้อหาในบล็อกข้อมูล 2 ไบต์

จ) จำนวนสมาชิกของตัวแปรอระเรย์ พารามิเตอร์นี้จะใช้กับโปรแกรมย่อย กำหนดตัวแปรอระเรย์เท่านั้น และใช้เนื้อที่ในบล็อกข้อมูล 2 ไบต์

1	2		3		4	5	6
2	2		20		4	2	2

ไบต์

ภาพที่ 5.2 โครงสร้างของบล็อกข้อมูล

เมื่อโปรแกรมประยุกต์เรียกใช้ฟังก์ชันจัดการบนจอภาพโดยผ่านโปรแกรมเชื่อมโยง โปรแกรมเชื่อมโยงจะตรวจสอบการติดตั้งฟังก์ชันการจัดการบนจอภาพในหน่วยความจำก่อน โดยตรวจสอบข้อมูลในตำแหน่งก่อนจุดเริ่มต้นการทำงานของอินเตอร์พรีตหมายเลข 60h ซึ่งต้องมีค่าเท่ากับ SCDP แล้วจึงบรรจุพารามิเตอร์ต่าง ๆ ลงในบล็อกข้อมูลที่เตรียมไว้ จากนั้นจะส่งการทำงานไปยังจุดเริ่มต้นการทำงานของฟังก์ชันการจัดการบนจอภาพโดยการเรียกใช้อินเตอร์พรีตหมายเลข 60h ด้วยคำสั่งซึ่งเขียนเป็นภาษาแอสเซมบลีคือ int 60h เมื่อการทำงานของฟังก์ชันสิ้นสุดลงก็จะส่งการทำงานกลับคืนมายังโปรแกรมประยุกต์ พร้อมทั้งส่งรหัสแสดงความผิดพลาดกลับมาที่ตัวแปรที่รับรหัสความผิดพลาดด้วย โดยจะมีค่าเท่ากับ 0 ถ้าฟังก์ชันทำงานปกติ และมีค่าไม่เท่ากับ 0 ถ้าการทำงานของฟังก์ชันไม่ปกติ

5.4.3 โปรแกรมเชื่อมโยงสำหรับโปรแกรมประยุกต์แต่ละภาษา ในการเรียกใช้ฟังก์ชันใด ๆ จากโปรแกรมประยุกต์ การส่งพารามิเตอร์จากโปรแกรมประยุกต์ให้กับโปรแกรมเชื่อมโยงจะส่งผ่านมาทางสแต็ก ลำดับในการเก็บพารามิเตอร์แต่ละตัวลงสแต็กจะถูกกำหนดโดยตัวแปลภาษา ซึ่งตัวแปลของแต่ละภาษาจะมีการกำหนดลำดับไม่เหมือนกัน [5,6,7] ดังนั้นการพัฒนาโปรแกรมเชื่อมโยงจำเป็นต้องพัฒนาโปรแกรมขึ้นมาใช้เฉพาะสำหรับโปรแกรมประยุกต์แต่ละภาษา ในที่นี้ได้พัฒนาโปรแกรมเชื่อมโยง 3 โปรแกรมเพื่อใช้กับโปรแกรมประยุกต์ภาษาซี ปาสคาล และฟอร์แทรน ที่ใช้ตัวแปลภาษาของบริษัทไมโครซอฟต์ดังนี้

5.4.3.1 โปรแกรมเชื่อมโยงสำหรับโปรแกรมไมโครซอฟต์ซี

การพัฒนาโปรแกรมเชื่อมโยงสำหรับโปรแกรมไมโครซอฟต์ซี

กำหนดให้มีการเรียกใช้งานจากโปรแกรมประยุกต์ในลักษณะฟังก์ชัน การส่งพารามิเตอร์มาทางสแต็กจะขึ้นกับโมเดลของโปรแกรมประยุกต์ที่เลือกใช้ด้วย โมเดลของโปรแกรมไมโครซอฟต์ซีมีทั้งหมด 5 โมเดลคือ small compact medium large และ huge แต่ละโมเดลจะมีการส่งพารามิเตอร์แตกต่างกัน ในที่นี้ได้พัฒนาโปรแกรมเชื่อมโยงให้ใช้ได้กับโปรแกรมประยุกต์

โมเดล small เท่านั้น ตำแหน่งอ้างอิงต่าง ๆ จะถูกอ้างในลักษณะเนียร์พอยน์เตอร์ (near pointer) คืออ้างเพียงค่าออฟเซต ส่วนค่าเซกเมนต์จะอ้างได้จากรีจิสเตอร์ DS สำหรับโปรแกรมย่อยที่จะเรียกใช้ได้ในลักษณะของฟังก์ชันประกอบด้วย

5.4.3.1.1 โปรแกรมย่อย scdpc ใช้สำหรับฟังก์ชันการจัดการทั่วไป มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
scdpc (parm, len, rc);
```

โดยที่

parm คือข้อความที่เป็นชื่อฟังก์ชัน

len คือความยาวของข้อความที่เป็นชื่อฟังก์ชัน

rc คือตำแหน่งที่อยู่ของตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์

ตัวอย่างการใช้งาน

```
int rc;
```

```
scdpc ("display screen", 14, &rc);
```

5.4.3.1.2 โปรแกรมย่อย scdpcv ใช้สำหรับฟังก์ชันการกำหนดตัวแปรเดี่ยว มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
scdpcv (parm, len, rc, var, vlen);
```

โดยที่

parm คือข้อความที่เป็นชื่อฟังก์ชัน

len คือความยาวของข้อความที่เป็นชื่อฟังก์ชัน

rc คือตำแหน่งที่อยู่ของตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์

var คือตำแหน่งที่อยู่ของตัวแปรเดี่ยว

vlen คือความยาวที่ใช้เก็บค่าข้อมูลของตัวแปรเดี่ยว

ตัวอย่างการใช้งาน

```
int rc, number;
```

```
scdpcv ("vdefine number i", 16, &rc, &number, 2);
```

5.4.3.1.3 โปรแกรมย่อย scdpcva ใช้สำหรับฟังก์ชันการกำหนดตัวแปรอะเรย์ มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
scdpcva (parm, len, rc, var, vlen, nelem);
```

โดยที่

parm คือข้อความที่เป็นชื่อฟังก์ชัน

len คือความยาวของข้อความที่เป็นชื่อฟังก์ชัน

rc คือตำแหน่งที่อยู่ของตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์

var คือตำแหน่งที่อยู่ของตัวแปรอะเรย์

vlen คือความยาวที่ใช้เก็บค่าข้อมูลของตัวแปรอะเรย์

nelem คือจำนวนสมาชิกของตัวแปรอะเรย์

ตัวอย่างการใช้งาน

```
int rc, id[20];
```

```
scdpcva ("vdefarr id i", 12, &rc, &id[0], 2, 20);
```

5.4.3.2 โปรแกรมเชื่อมโยงสำหรับโปรแกรมไมโครซอฟต์แวร์แทน

การพัฒนาโปรแกรมเชื่อมโยงสำหรับโปรแกรมไมโครซอฟต์แวร์แทน

ได้กำหนดให้มีการเรียกใช้งานจากโปรแกรมประยุกต์ในลักษณะของโปรแกรมย่อย (subroutine) การส่งพารามิเตอร์จะต้องอ้างอิงตำแหน่งต่าง ๆ ในลักษณะพาร์พอยน์เตอร์เท่านั้นคืออ้างอิงทั้งค่าเชกเมนต์และออฟเซต สำหรับโปรแกรมย่อยที่จะเรียกใช้งานในลักษณะของโปรแกรมย่อยประกอบด้วย

5.4.3.2.1 โปรแกรมย่อย SCDF ใช้สำหรับฟังก์ชันการจัด

การทั่วไป มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
CALL SCDF (PARAM, LEN, RC)
```

โดยที่

PARAM คือตัวแปรเก็บข้อความที่เป็นชื่อฟังก์ชัน

LEN คือตัวแปรเก็บความยาวของข้อความที่เป็นชื่อฟังก์ชัน

RC คือตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์

ตัวอย่างการใช้งาน

```
CHARACTER*20 PARM
INTEGER RC, LENGH
PARM = 'DISPLAY SCREEN'
LENGH = 14
CALL SCDF (PARM, LENGH, RC)
```

5.4.3.2.2 โปรแกรมย่อย SCDFV ใช้สำหรับฟังก์ชันการกำหนดตัวแปรเดี่ยว มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
CALL SCDFV (PARM, LEN, RC, VAR, VLEN)
```

โดยที่

PARM คือตัวแปรเก็บข้อความที่เป็นชื่อฟังก์ชัน
 LEN คือตัวแปรเก็บความยาวของข้อความที่เป็นชื่อฟังก์ชัน
 RC คือตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์
 VAR คือตำแหน่งที่อยู่ของตัวแปรเดี่ยว
 VLEN คือตัวแปรเก็บความยาวของค่าข้อมูลของตัวแปรเดี่ยว

ตัวอย่างการใช้งาน

```
CHARACTER*20 PARM
INTEGER RC, LENGH, LENGH1, NUMBER
PARM = 'VDEFINE NUMBER I'
LENGH = 16
LENGH1 = 2
CALL SCDFV (PARM, LENGH, RC, NUMBER, LENGH1)
```

5.4.3.2.3 โปรแกรมย่อย SCDFVA ใช้สำหรับฟังก์ชันการกำหนดตัวแปรอะเรย์ มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
CALL SCDFVA (PARM, LEN, RC, VAR, VLEN, NELEM)
```

โดยที่

PARM คือตัวแปรเก็บข้อความที่เป็นชื่อฟังก์ชัน

LEN คือตัวแปรเก็บความยาวของข้อความที่เป็นชื่อฟังก์ชัน
 RC คือตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์
 VAR คือตำแหน่งที่อยู่ของตัวแปรอะเรย์
 VLEN คือตัวแปรเก็บความยาวของค่าข้อมูลของตัวแปรอะเรย์
 NELEM คือตัวแปรเก็บจำนวนสมาชิกของตัวแปรอะเรย์

ตัวอย่างการใช้งาน

```
CHARACTER*20 PARM
INTEGER RC, LENGTH, LENGTH1, LENGTH2, ID(20)
PARM = 'VDEFARR ID I'
LENGTH = 12
LENGTH1 = 2
LENGTH2 = 20
CALL SCDFVA (PARM, LENGTH, RC, NUMBER, LENGTH1)
```

5.4.3.3 โปรแกรมเชื่อมโยงสำหรับโปรแกรมไมโครซอฟต์ปาสคาล

การพัฒนาโปรแกรมเชื่อมโยงสำหรับโปรแกรมไมโครซอฟต์ปาสคาล

กำหนดให้มีการเรียกใช้งานจากโปรแกรมประยุกต์ในลักษณะของโพรซีเจอร์ (Procedure) ซึ่งผู้ใช้ต้องกำหนดเป็นโพรซีเจอร์ซึ่งอยู่ภายนอกด้วยคำสั่ง EXTERN ในโปรแกรมประยุกต์ด้วย การส่งพารามิเตอร์จะต้องอ้างอิงตำแหน่งต่าง ๆ ในลักษณะของพาร์พอยน์เตอร์เท่านั้น สำหรับโปรแกรมย่อยที่จะเรียกใช้งานประกอบด้วย

5.4.3.3.1 โปรแกรมย่อย SCDPP ใช้สำหรับฟังก์ชันการจัดการทั่วไป มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```
SCDPP (PARM, RC);
```

โดยที่

PARM คือตัวแปรเก็บข้อความที่เป็นชื่อฟังก์ชัน
 RC คือตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์

ตัวอย่างการใช้งาน

```
VAR PARM : LSTRING[80];
RC : INTEGER;
```

```

    PARM := 'DISPLAY SCREEN';
    SCDPP (PARM, RC);
  
```

5.4.3.3.2 โปรแกรมย่อย SCDPPV ใช้สำหรับฟังก์ชันการกำหนดตัวแปรเดี่ยว มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```

    SCDPPV (PARM, RC, VAR, VLEN);
  
```

โดยที่

PARM คือตัวแปรเก็บข้อความที่เป็นชื่อฟังก์ชัน
 RC คือตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์
 VAR คือตำแหน่งที่อยู่ของตัวแปรเดี่ยว
 VLEN คือตัวแปรเก็บความยาวของค่าข้อมูลของตัวแปรเดี่ยว

ตัวอย่างการใช้งาน

```

    VAR PARM : LSTRING(80);
        RC, NUMBER, LENGTH1 : INTEGER;
    PARM := 'VDEFINE NUMBER I';
    LENGTH1 := 2;
    SCDPPV (PARM, RC, ADS NUMBER, LENGTH1);
  
```

5.4.3.3.3 โปรแกรมย่อย SCDPPVA ใช้สำหรับฟังก์ชันการกำหนดตัวแปรอะเรย์ มีวิธีการเรียกใช้และส่งพารามิเตอร์ไปทางสแตกดังนี้

```


    SCDPPVA (PARM, RC, VAR, VLEN, NELEM);
  
```

โดยที่

PARM คือตัวแปรเก็บข้อความที่เป็นชื่อฟังก์ชัน
 RC คือตัวแปรรับรหัสความผิดพลาดในโปรแกรมประยุกต์
 VAR คือตำแหน่งที่อยู่ของตัวแปรอะเรย์
 VLEN คือตัวแปรเก็บความยาวของค่าข้อมูลของตัวแปรอะเรย์
 NELEM คือตัวแปรเก็บจำนวนสมาชิกของตัวแปรอะเรย์

ตัวอย่างการใช้งาน

```
VAR PARM : LSTRING(80);  
    RC, LNGTH1, LNGTH2 : INTEGER;  
    ID : ARRAY [1..20] OF INTEGER;  
PARM := 'VDEFARR-ID I';  
LNGTH1 := 2;  
LNGTH2 := 20;  
SCDPPVA (PARM, RC, ADS ID, LNGTH1, LNGTH2);
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย