



บทที่ 3

โครงร่างโปรแกรม

3.1 วัตถุประสงค์

ในบทนี้จะกล่าวถึงโครงร่างของโปรแกรม File Editor และ Screen Generator โดยจะกล่าวถึงข้อกำหนดของโปรแกรมเหล่านี้ ข้อกำหนดของฟอร์ม โครงสร้างของข้อมูลที่เกี่ยวข้องกับฟอร์ม และที่เกี่ยวข้องกับระเบียบการเก็บข้อมูล จากการพิจารณาองค์ประกอบพื้นฐานที่ได้กล่าวถึงในบทที่ 2

หัวข้อในบทนี้จะแบ่งออกเป็น 3 ส่วน คือ

- ก. ข้อกำหนดและโครงสร้างข้อมูล
- ข. โปรแกรม File Editor
- ค. โปรแกรม Screen Generator

ส่วนที่เกี่ยวข้องกับข้อกำหนด และโครงสร้างข้อมูล จะกล่าวถึงข้อกำหนดของฟอร์ม โครงสร้างข้อมูลของฟอร์มที่ใช้ และโครงสร้างของข้อมูลภายในแฟ้มข้อมูล ส่วนของข้อกำหนดการใช้งานจะกล่าวถึงข้อกำหนดต่าง ๆ ในการใช้งานของโปรแกรม File Editor และ Screen Generator ที่เป็นพื้นฐานในการออกแบบโปรแกรม

โครงร่างของโปรแกรมจะกล่าวถึงหน้าที่หลัก, ส่วนประกอบต่าง ๆ, การทำงาน และการเรียกใช้ของโปรแกรม File Editor และ Screen Generator

ก่อนที่จะกล่าวถึงส่วนต่าง ๆ เหล่านี้จะได้กล่าวถึงหน้าที่หลักอย่างย่อ ๆ ของ File Editor และ Screen Generator ก่อน

File Editor หมายถึง โปรแกรมส่วนจัดการแฟ้มข้อมูล โดยกำหนดฟอร์มและโครงสร้างข้อมูล รวมทั้งชื่อแฟ้มข้อมูลที่ต้องการจัดการ ซึ่งจะทำการแสดงข้อมูลที่ละเอียดเป็นเลื่อนตำแหน่งไปที่ช่องเติมข้อมูลต่าง ๆ ได้ ตรวจสอบความเป็นไปได้ของข้อมูล ค้นหา แก้ไข และจัดเก็บข้อมูล โดยจัดการแฟ้มข้อมูลไม่จำกัดจำนวน ทีละแฟ้มข้อมูล

Screen Generator หมายถึง โปรแกรมส่วนจัดการสร้างฟอร์ม โดยการวางตำแหน่งข้อความต่าง ๆ ภายในฟอร์มนั้นแบบ Full Screen Edit แล้วจัดเก็บข้อมูลของฟอร์มนั้น (ลงเพิ่มข้อมูล) เพื่อให้ File Editor ใช้เป็นสื่อแสดงข้อมูลของระเบียบต่อไป

3.2 ข้อกำหนด และโครงสร้างข้อมูล

ในหัวข้อนี้จะกล่าวถึงข้อกำหนดของโปรแกรม File Editor และ Screen Generator ที่เป็นข้อกำหนดในการใช้งาน และข้อกำหนดของฟอร์มที่เป็นตัวกลางระหว่างโปรแกรม File Editor และผู้ใช้ ซึ่งสร้างโดยใช้ Screen Generator ตลอดจนกล่าวถึงโครงสร้างข้อมูลของฟอร์ม และของระเบียบที่เก็บข้อมูลนั้น

3.2.1 ข้อกำหนดในการใช้งาน

ข้อกำหนดในการใช้งานของโปรแกรม File Editor จะเป็นตัวกำหนดสิ่งจำเป็นที่ต้องมีประกอบกัน ซึ่งเป็นพารามิเตอร์ของระบบนั่นเอง

- ใช้จัดการกับเพิ่มข้อมูลได้ครั้งละหนึ่งเพิ่ม ซึ่งชื่อเพิ่มข้อมูลนี้เป็นไปตามข้อกำหนดของ MS-DOS ซึ่งเป็นระบบปฏิบัติการ (Operating System) ของเครื่องไมโครคอมพิวเตอร์ตระกูลไอบีเอ็ม-พีซี ข้อกำหนดนี้คือ มีได้ 8 ตัวอักษร และต่อหลังจุดได้อีก 3 ตัว หรือจะมี path ด้วยก็ได้

- ต้องมีฟอร์มสำหรับแสดง หรือรับข้อมูล คู่กับเพิ่มข้อมูลนี้
- ต้องระบุชนิดข้อมูลของช่องรับข้อมูลในฟอร์ม
- ชุดของข้อกำหนดทั้ง 3 ข้อที่กล่าวแล้วข้างต้น มีจำนวนเท่าใดก็ได้ โดยที่หนึ่งชุดจัดการกับเพิ่มข้อมูลหนึ่งเพิ่ม

ข้อกำหนดของโปรแกรม Screen Generator เป็นดังนี้

- ใช้สร้างเพิ่มข้อมูลที่เก็บคุณสมบัติของฟอร์มต่าง ๆ ได้ไม่จำกัดจำนวน โดยที่แต่ละฟอร์มจะมี 2 เพิ่มข้อมูลสำหรับเป็นข้อมูลเพื่อการแก้ไข และข้อมูลของฟอร์มที่ล้นขนาดแล้ว

- การสร้างและแก้ไข จะแยกกันทำงาน ทำให้สามารถสร้างฟอร์มหนึ่งจากการแก้ไขอีกฟอร์มหนึ่งได้

- การกำหนดชื่อฟอร์มก็เป็นเช่นเดียวกับการกำหนดชื่อเพิ่มข้อมูล แต่ส่วนที่อยู่หลังจุด 3 ตัว ถ้ามิได้กำหนดไว้ โปรแกรมจะให้เป็น '.SCN' สำหรับข้อมูลเพื่อการแก้ไข และ '.SEP' สำหรับข้อมูลฟอร์มที่ลคขนาดแล้ว ซึ่งโปรแกรม File Editor จะใช้ข้อมูลจากฟอร์มที่ลคขนาดแล้ว

3.2.2 ข้อกำหนดของฟอร์ม

เนื่องจากฟอร์มที่ใช้แสดงบนจอภาพ โดยทั่วไปจะประกอบด้วยข้อความถาวร ช่องใส่ข้อมูลและตำแหน่งแสดงข้อมูลพิเศษ สำหรับการจัดวางตำแหน่ง สามารถวางที่ตำแหน่งใดก็ได้ตามขนาดจอภาพที่กำหนดไว้ (ขนาด 25 บรรทัด และ 79 คอลัมน์) แต่ในการทำงานของ File Editor ได้มีข้อจำกัดทางการวางตำแหน่งที่ใช้แสดงข้อมูลพิเศษเอาไว้ ดังนั้นในส่วนนี้จะกล่าวถึงข้อกำหนดของฟอร์มในส่วนที่จำกัดโดย File Editor ตัวอย่างดังรูปที่ 3.1

ทะเบียนประวัตินิติเก่า

> ลำดับที่ >

>

รหัส : :

ชื่อ : :

ที่ทำงาน : :

: :

โทรศัพท์ : :

>

[ESC] = เลิก

[F2] = บันทึก

รูปที่ 3.1 ฟอร์มทะเบียนประวัตินิติเก่า

ส่วนของข้อความถาวร เป็นข้อความที่ช่วยให้ผู้ใช้สามารถใส่ข้อมูลได้ง่าย หรือสามารถสอบถามได้ง่าย เพราะจะกำหนดให้ตรงกับคุณสมบัติของช่องรับข้อมูลถัดมา หรือใช้ หน้าจอที่ของแป้นคีย์ต่าง ๆ ส่วนนี้สามารถจะอยู่ที่ตำแหน่งใดก็ได้บนจอภาพ แต่ไม่เกินขอบเขตที่กำหนด

ส่วนของช่องใส่ข้อมูล เป็นส่วนที่อยู่ระหว่างเครื่องหมาย : : มีขนาดได้ตั้งแต่ 1 ตัวอักษร ถึงไม่เกินหนึ่งบรรทัด โดยปกติส่วนนี้จะอยู่บริเวณกลางจอภาพ ต้องเว้นที่ว่างไว้สำหรับ ตำแหน่งแสดงข้อมูลพิเศษ

ส่วนตำแหน่งแสดงข้อมูลพิเศษ เป็นตำแหน่งที่กำหนดด้วยเครื่องหมาย > ใช้แสดงข้อมูลที่สำคัญบางอย่าง ในที่นี้ควรมีอย่างน้อย 4 ตำแหน่ง คือ แสดงสถานะการลบ (delete status) ลำดับรายการ (record number) และข้อมูลเพื่อการโต้ตอบอีก 2 ตำแหน่ง หรือใช้แสดงข้อมูลที่ไม่ต้องการให้ผู้ใช้เปลี่ยนแปลงได้

การแบ่งขอบเขตการแสดงตำแหน่งข้อมูลต่าง ๆ เป็นดังตัวอย่างรูปที่ 3.2

ข้อมูลค่าแรงพนักงานรายวัน

> ลำดับที่ >

>

รหัส > แผนก >

ชื่อ >

วันทำงาน : : วัน

ค่าแรง > บาท

>

[ESC] = เลิก [F2] = บันทึก

รูปที่ 3.2 การแบ่งขอบเขตของฟอร์ม

แบ่งหน้าจอภาพออกเป็น 3 ส่วน ส่วนบน และส่วนล่างเป็นข้อความถาวร และตำแหน่งแสดงข้อมูลพิเศษ 4 ตำแหน่งคือ ตำแหน่งแรกแสดงสถานะการลบ, ตำแหน่งถัดมาแสดงลำดับรายการ, ตำแหน่งที่ 3 แสดงข้อความพิเศษ 1, และตำแหน่งที่ 4 แสดงข้อความพิเศษ 2

ที่ส่วนกลางของจอภาพ ตรงกับระเบียบในเพิ่มข้อมูลค่าแรงพนักงานรายวัน ที่ประกอบด้วยข้อมูลย่อยคือ สถานะการลบ, รหัส, แผนก, ชื่อ, วันทำงาน และค่าแรง สิ่งที่เกิดค่าแรง แทนที่จะเป็นช่องรับข้อมูลก็เป็นตำแหน่งแสดงข้อมูลพิเศษแทน เพื่อให้ผู้ใช้ดูได้อย่างเดียว แก้ไขไม่ได้

3.2.3 ชนิดของช่องใส่ข้อมูลในฟอร์ม

ช่องใส่ข้อมูลในฟอร์มแต่ละช่อง จะมีชนิดข้อมูลควบคุมกันอยู่ 9 ชนิด คือ

- ข้อความอังกฤษ (S = String)
- ข้อความไทย (T = Thai)
- ตัวอักษรใหญ่ (อังกฤษ) (U = Upper case)
- เลขจำนวนเต็ม (I = Integer)
- เลขจำนวนจริง (R = Real)
- เลขทศนิยม 2 ตำแหน่ง (\$ = Real with two decimal places)
- ตัวอักษร 1 ตัว (C = Single character)
- ค่าตรรกะ (B = Boolean)
- แบบวันที่ (F = Date form dd/mm/yy)

3.2.4 ชนิดของตำแหน่งแสดงข้อความพิเศษ

เช่นเดียวกับชนิดของช่องใส่ข้อมูลในฟอร์ม เพียงแค่ 3 ตำแหน่งแรก และตำแหน่งสุดท้าย โปรแกรมเป็นผู้กำหนดชนิดเอง สัญลักษณ์ที่ใช้แทนชนิดข้อมูลจะเป็นอักษรตัวเล็ก คือ s, t, u, i, r, @, c, b, f โดยมี @ เหมือนกับ \$

3.2.5 โครงสร้างข้อมูลส่วนที่เกี่ยวข้องกับฟอร์ม

โครงสร้างข้อมูลของฟอร์มที่เก็บในแฟ้มข้อมูลมี 2 ประเภท คือ ประเภทเพื่อแก้ไข และประเภทลดขนาดเพื่อใช้งาน ข้อมูลของฟอร์มประกอบด้วย

- จำนวนของข้อความถาวร (N)
- ตำแหน่ง (X, Y) พร้อมข้อความถาวรทั้งหมด N ชุด
- จำนวนช่องเติมข้อมูล (M)
- ตำแหน่งเริ่มต้น (X, Y) และความกว้างช่อง (L) จำนวน M ชุด
- จำนวนตำแหน่งแสดงข้อมูลพิเศษ (K)
- ตำแหน่งเริ่มต้น (X, Y) จำนวน K ชุด

การเก็บข้อมูลทั้ง 2 ประเภทเป็นดังรูปที่ 3.3 และ 3.4

บรรทัด (X, Y)	ข้อความทั้งบรรทัด	} จำนวนเท่ากับบรรทัดที่มี ข้อความปรากฏอยู่
⋮	⋮	

รูปที่ 3.3 โครงสร้างข้อมูลของฟอร์มประเภทเพื่อแก้ไข

N			
X, Y	ข้อความถาวร		
	⋮	} N ชุด	
M			
X, Y	L	} M ชุด	
	⋮		
K		} K ชุด	
X, Y			
	⋮		

รูปที่ 3.4 โครงสร้างข้อมูลของฟอร์มประเภทลดขนาดเพื่อใช้งาน

ทั้ง 2 ประเภทแยกกันเพื่อผลทางความเร็วในการแก้ไขฟอร์ม และลดขนาดเพื่อนำไปใช้งาน

3.2.6 โครงสร้างข้อมูลของระเบียบ

โครงสร้างข้อมูลของระเบียบประกอบด้วย ข้อมูลชนิดต่างๆ ตามหัวข้อย่อย

3.2.3 แต่ข้อมูลจริงในแต่ละระเบียบ จะมีเพียงเท่าที่มีช่องเติมข้อมูลในฟอร์มสำหรับเพิ่มข้อมูลนั้นๆ ขนาดของข้อมูลมี 2 แบบ คือ แบบที่มีความยาวแน่นอน และแบบที่ขึ้นอยู่กับความยาวช่องเติมข้อมูล

ข้อมูลที่มีความยาวแน่นอน ได้แก่

- เลขจำนวนเต็ม มีขนาด 2 ไบต์
- เลขจำนวนจริง และ เลขทศนิยม 2 ตำแหน่ง มีขนาด 6 ไบต์
- ตัวอักษร และค่าตรรกะมีขนาด 1 ไบต์
- แบบวันที่มีขนาด 9 ไบต์

ข้อมูลชนิดที่ขึ้นอยู่กับความยาวของช่องเติมข้อมูล ได้แก่

- ข้อความอังกฤษ และตัวอักษรใหญ่ มีขนาดเท่ากับความยาวของช่อง +1
 - ข้อความไทย มีขนาดตามสูตร $\text{int}(1.2 * \text{ความยาวของช่อง}) + 4$
- การเรียงลำดับข้อมูลชนิดต่างๆ ในแต่ละระเบียบให้เรียงตามช่องเติม

ข้อมูลที่ปรากฏบนฟอร์ม และ ไบต์แรกสุดเป็นชนิดตรรกะเพื่อเก็บสถานะการลบ

จากตัวอย่างทะเบียนประวัตินิติศีกา มีโครงสร้างของข้อมูลดังรูปที่ 3.5 สามารถแทนได้ด้วยรหัสบอกโครงสร้างข้อมูล และชนิดข้อมูลในฟอร์ม 'U7T20T30T30S12' หรือจากตัวอย่างข้อมูลค่าแรงพนักงาน มีโครงสร้างข้อมูลเป็นดังรูปที่ 3.6 รหัสบอกโครงสร้างข้อมูล และชนิดข้อมูลในฟอร์มเป็น 's7s10s30I@'

สถานะการลบ	รหัส	ชื่อ	ที่ทำงาน 1	ที่ทำงาน 2	โทรศัพท์
------------	------	------	------------	------------	----------

รูปที่ 3.5 โครงสร้างข้อมูลทะเบียนประวัตินิติศีกา

สถานะการลบ	รหัส	แผนก	ชื่อ	วันทำงาน	ค่าแรง
------------	------	------	------	----------	--------

รูปที่ 3.6 โครงสร้างข้อมูลค่าแรงพนักงาน

รหัสอักษรตัวเล็กตัวใหญ่ เป็นไปตามข้อกำหนดในหัวข้อย่อย 3.2.3 และ 3.2.4 ตัวเลขที่ตามหลังรหัส S,U หรือ T ใช้บอกความกว้างของช่องรับข้อมูล และคำนวณหาความยาวข้อมูลในระเบียน

โครงสร้างข้อมูลที่อยู่ในแฟ้มข้อมูล เป็นชนิดเดียวกับโครงสร้างแฟ้มข้อมูลที่เกิดจากการเขียนโปรแกรมด้วยภาษาปาสคาล ทำให้สามารถใช้แฟ้มข้อมูลเหล่านั้นกับโปรแกรมอื่นที่เขียนขึ้นด้วยภาษาปาสคาลได้โดยง่าย เพียงแต่กำหนดโครงสร้างข้อมูลในโปรแกรมภาษาปาสคาลนั้นให้ตรงกับรหัสที่กำหนดขึ้น เช่น ทะเบียนประวัติคนลี้ภัย เมื่อเขียนด้วยภาษาปาสคาลจะเป็นดังนี้

```
Type OldStud = record
```

```
delstatus : boolean ;
name      : string[27] ;
addr 1    : string[39] ;
addr 2    : string[39] ;
telephone : string[12] ;
```

```
end;
```

การเก็บข้อมูลชนิด string ในปาสคาลต้องเพิ่มตัวเก็บความยาวของ string นั้นอีก 1 ไบต์ เช่น ชื่อเป็นชนิด T20 ความยาวของข้อมูลในระเบียนจะเป็น 28 ไบต์ เท่ากับ string[27]

หมายเหตุ ในทางปฏิบัติ ข้อมูล "ชื่อ" ซึ่งเป็นชนิด T20 อาจจะมีขนาดช่องรับข้อมูลไม่เท่ากับ 20 ก็ได้ แต่จำนวนตัวอักษรทั้งหมด ที่รับได้จะมีขนาดที่จุลงใน String[27] เท่านั้น ส่วนเกิน (ถ้ามี) จะถูกตัดทิ้งไป

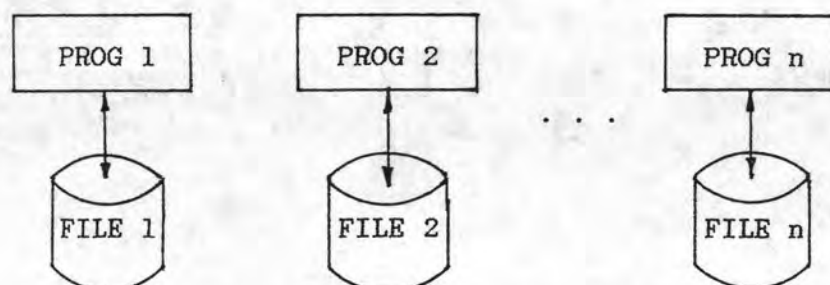
3.3 File Editor

ในหัวข้อนี้จะกล่าวถึง หน้าที, ขั้นตอนการทำงาน, และการใช้งานของ File Editor โดยอาศัยการออกแบบที่ได้เลือกไว้ในบทที่ 2 และข้อกำหนดของโปรแกรม ตามหัวข้อ 3.2

3.3.1 หน้าที

File Editor เป็นเครื่องมืออย่างหนึ่ง สำหรับระบบฐานข้อมูลที่สร้างขึ้น เพื่อให้สามารถจัดการกับเพิ่มข้อมูลตามหน้าที่ที่กำหนด ซึ่งสามารถทำหน้าที่พื้นฐานของระบบฐานข้อมูลได้ คือ ค้นหา เพิ่มเติม แก้ไข และลบข้อมูลที่ต้องการได้ เครื่องมือนี้สามารถใช้กับเพิ่มข้อมูลได้หลายแบบ เพียงแค่เปลี่ยนพารามิเตอร์ที่ควบคุมเท่านั้น โดยก่อนอื่นจะขอกกล่าวถึงการจัดการเพิ่มข้อมูลต่างๆ ไปเสียก่อน

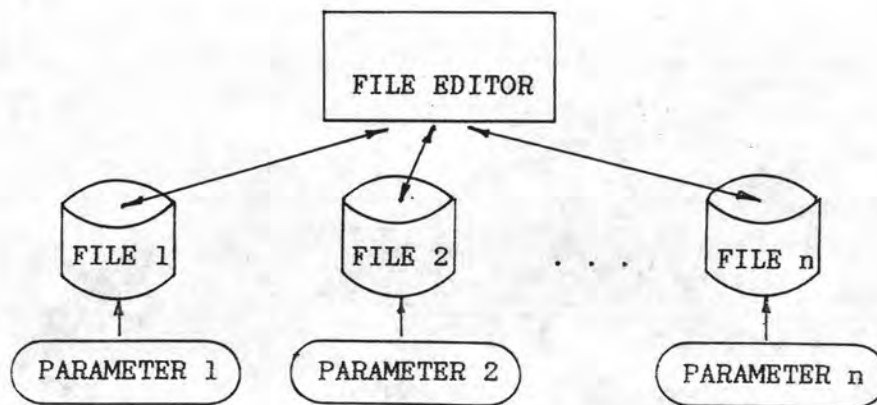
ในภาษารุ่นที่ 3 เช่น COBOL, Pascal หรือ C การพัฒนาโปรแกรมจากภาษาเหล่านี้ เมื่อต้องการจัดการกับเพิ่มข้อมูลจำนวนมากหลายแบบ จะเกิดความยุ่งยาก เนื่องจากภาษาเหล่านี้ ต้องกำหนดโครงสร้างข้อมูลระเบียบที่แน่นอนสำหรับแต่ละเพิ่มข้อมูล และการแสดงข้อมูลแต่ละระเบียบทางจอภาพตามตำแหน่งที่ต้องการ ก็ต้องมีโปรแกรมที่ซับซ้อนขึ้นเพื่อจัดวางตำแหน่ง โปรแกรมที่เขียนขึ้นก็ยิ่งใช้เฉพาะกับเพิ่มข้อมูลที่ได้กำหนดไว้แล้วเท่านั้น เมื่อต้องการใช้กับข้อมูลโครงสร้างอื่น ต้องแก้ไข และคอมไพล์ (Compile) โปรแกรมนั้นใหม่ ดังนั้นเมื่อต้องการใช้หลายเพิ่มข้อมูล ทำให้ต้องมีโปรแกรมเฉพาะควบคุมแต่ละเพิ่มข้อมูล ซึ่งเป็นการเปลืองเนื้อที่ในหน่วยความจำ ทั้งที่โปรแกรมเหล่านี้มีลักษณะการทำงานที่เหมือนกัน ต่างกันเพียงโครงสร้างข้อมูลของแต่ละระเบียบเท่านั้น ตามรูปที่ 3.7



รูปที่ 3.7 โปรแกรมจัดการเพิ่มข้อมูลเฉพาะ

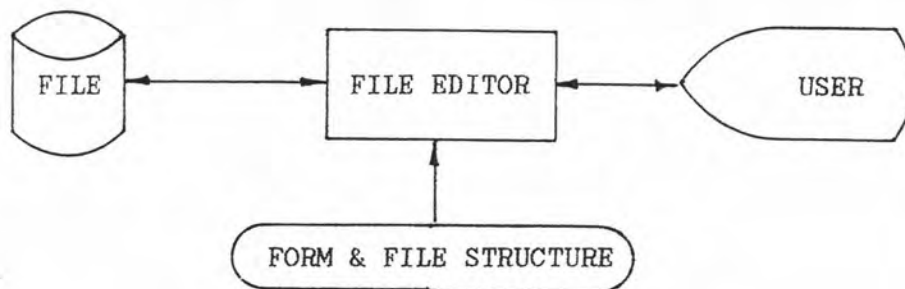
ดังตัวอย่างเช่น ในการเก็บข้อมูลเงินเดือนพนักงาน ต้องมีแฟ้มข้อมูลหลายแบบ เช่น แฟ้มข้อมูลฝ่ายบุคคล ซึ่งประกอบด้วย รหัสประจำตัวพนักงาน, ชื่อพนักงาน และแฟ้มข้อมูลฝ่ายบัญชี ซึ่งประกอบด้วย รหัสประจำตัวพนักงาน, เงินเดือนที่ได้รับ, ภาษีที่หัก เป็นต้น การจัดการกับแฟ้มข้อมูลเหล่านี้ต้องมีโปรแกรมเฉพาะ เช่น โปรแกรมที่ควบคุมแฟ้มข้อมูลฝ่ายบุคคล ก็ต้องมีโครงสร้างของระเบียบตามแฟ้มข้อมูลฝ่ายบุคคลนั้น และโปรแกรมที่ควบคุมแฟ้มข้อมูลฝ่ายบัญชี ก็ต้องมีโครงสร้างระเบียบตามแฟ้มข้อมูลฝ่ายบัญชีนั้น ทั้งที่โปรแกรมทั้ง 2 มีลักษณะการทำงานที่คล้ายกัน เช่น สามารถแสดงข้อมูล, เพิ่มเติม, แก้ไข หรือลบข้อมูลในแต่ละระเบียบได้เช่นเดียวกัน แต่แยกเป็น 2 โปรแกรม และถ้ามีแฟ้มข้อมูลมากขึ้นก็ต้องมีโปรแกรมเฉพาะแต่ละแฟ้มข้อมูลมากขึ้นเช่นเดียวกัน ขั้นตอนในการเปลี่ยนแปลงโปรแกรมและทำการคอมไพล์ โปรแกรมเหล่านี้ใหม่ก็ต้องใช้เวลามากขึ้นตามไปด้วย

ความไม่สะดวกในการจัดการแฟ้มข้อมูลจำนวนมากด้วยภาษารุ่นที่ 3 นี้เอง จึงทำให้มีการออกแบบโปรแกรมอันหนึ่งที่สามารถจัดการกับแฟ้มข้อมูลแตกต่างกันออกไป เพียงแต่บอกเป็นพารามิเตอร์เฉพาะแฟ้มข้อมูลนั้น ๆ ซึ่งเป็นหน้าที่หลักของ File Editor นั้นเอง ตามรูปที่ 3.8



รูปที่ 3.8 โปรแกรมจัดการแฟ้มข้อมูลทั่วไป

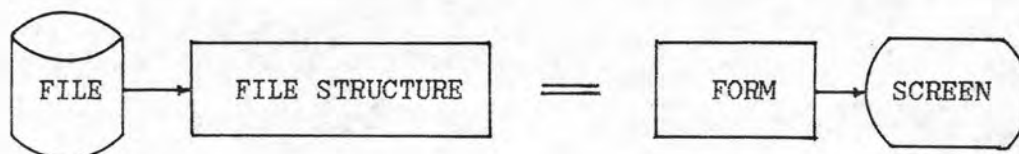
ในการจัดรูปแบบการออกจอภาพก็เป็นหน้าที่อีกอย่างหนึ่งของ File Editor โดยกำหนดให้มีฟอร์มหนึ่ง แทนแฟ้มข้อมูลหนึ่ง เช่นกัน ดังนั้นในพารามิเตอร์หนึ่งจึงประกอบด้วย ฟอร์ม และโครงสร้างข้อมูลของระเบียบในแฟ้มข้อมูลที่มีความสัมพันธ์กัน ดังนั้นในการจัดการเก็บแฟ้มข้อมูลหนึ่งจะเป็นดังรูปที่ 3.9



รูปที่ 3.9 การจัดการเพิ่มข้อมูลด้วย File Editor

ส่วน USER คือส่วนผู้ใช้โปรแกรมที่ต้องการจัดการกับเพิ่มข้อมูล จะมีโปรแกรม File Editor เป็นตัวจัดการ โดยมีฟอร์มเป็นตัวกำกับหน้าจอภาพ

ส่วน FORM & FILE STRUCTURE เป็นพารามิเตอร์ของโปรแกรม File Editor รูปแบบการแสดงผลฟอร์มเป็นอย่างไรก็ได้ ตามที่ได้กำหนดล่วงหน้า โดย FILE STRUCTURE เป็นตัวบอกโครงสร้างระเบียบภายในเพิ่มข้อมูล ซึ่งทั้ง 2 ส่วนต้องสัมพันธ์กัน เนื่องจากส่วนฟอร์มเป็นการกำหนดการแสดงผลบนจอภาพ ส่วนโครงสร้างระเบียบเป็นการกำหนดชนิดข้อมูลย่อยในระเบียบภายในเพิ่มข้อมูล ตัวอย่าง ทะเบียนประวัตินิติศีกษา และข้อมูลค่าแรงพนักงาน ในหัวข้อ 3.2



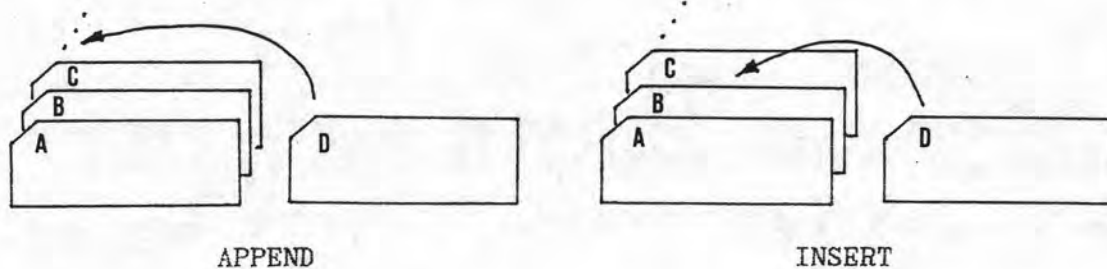
รูปที่ 3.10 แสดงความสัมพันธ์ระหว่างฟอร์มกับเพิ่มข้อมูล

ส่วนของโปรแกรม File Editor จะให้มีความสามารถในการติดต่อกับเพิ่มข้อมูลเท่านั้น ในส่วนความสามารถในการแสดงผล และแก้ไขข้อมูลของระเบียบที่แสดงภายในฟอร์มนั้น จะเป็นส่วนของโปรแกรม Screen Editor ซึ่งจะกล่าวถึงในหัวข้อ 4.3

การติดต่อกับแฟ้มข้อมูลของ File Editor แบ่งออกเป็น 3 ส่วนคือ

- ก. ส่วนรับและแก้ไขข้อมูล
- ข. ส่วนแสดงและค้นหาข้อมูล
- ค. ส่วนลบข้อมูล

ส่วนรับข้อมูล เป็นการเพิ่มเติมข้อมูลแต่ละระเบียบเข้าไปภายในแฟ้มข้อมูลที่กำหนดให้ การเพิ่มเติมข้อมูลแต่ละระเบียบทำได้ 2 วิธีคือ เพิ่มเติมข้อมูลที่ท้ายแฟ้มข้อมูล (Append) และ เพิ่มเติมข้อมูลที่ส่วนใดๆ ของแฟ้มข้อมูล (Insert) การเพิ่มเติมข้อมูลที่ท้ายแฟ้มข้อมูล เป็นการเพิ่มเติมข้อมูลที่ระเบียบตามลำดับ เลียนแบบการทำงานจัดแฟ้มข้อมูลแบบคนทำ ที่ทำโดยการกรอกข้อมูลที่ต้องการเพิ่มลงในแบบฟอร์มเปล่า แล้วนำแบบฟอร์มนั้นต่อท้ายระเบียบอื่นๆ ส่วนการเพิ่มเติมข้อมูลที่ส่วนใดๆ ของแฟ้มข้อมูล ทำได้โดยการกรอกข้อมูลที่ต้องการเพิ่มลงในแบบฟอร์มเปล่า แล้วนำฟอร์มนั้นไปแทรกระหว่างระเบียบใดๆ ซึ่งเป็นการจัดลำดับข้อมูลแต่ละระเบียบไปด้วย ตามตัวอย่าง มีแฟ้มข้อมูลหนึ่งที่มีข้อมูล A, B และ C เมื่อต่อท้ายข้อมูล D ลงไปอีก ในแฟ้มข้อมูลนั้นจะเป็น A, B, C และ D หรือ ถ้าเป็นการแทรก โดยนำข้อมูล D แทรกลงระหว่างข้อมูล B และ C ลำดับข้อมูลในแฟ้มข้อมูลจะเป็น A, B, D และ C ดังรูปที่ 3.11.



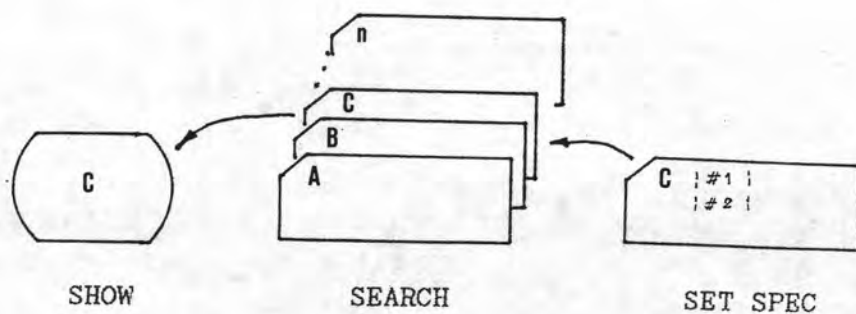
รูปที่ 3.11 แสดงการรับข้อมูล

ลำดับของข้อมูลในแฟ้มข้อมูล โดยปกติแล้วจะ ไม่มีความสำคัญในการทำงาน นัก เนื่องจากในระบบฐานข้อมูลแบบสัมพันธ์ (Relational Database) ไม่คำนึงถึงลำดับของข้อมูลแต่คำนึงถึงข้อมูลภายในมากกว่า ซึ่งลำดับข้อมูลนั้นสามารถเรียงลำดับใหม่ได้ โดยใช้โปรแกรมอื่นๆ จัดการเพิ่มเติม การแทรกมีประโยชน์เมื่อต้องการเรียงลำดับข้อมูลเอง

การแทรกในที่นี้จึงเป็นการแทรกข้อมูลลงในแฟ้มข้อมูลจริง (Physical Insert) ฟอรั่มว่างที่กล่าวถึงนี้ในโปรแกรม File Editor จะจัดให้เป็นหน้าท้ายสุดต่อจากระเบียนสุดท้ายของแฟ้มข้อมูล

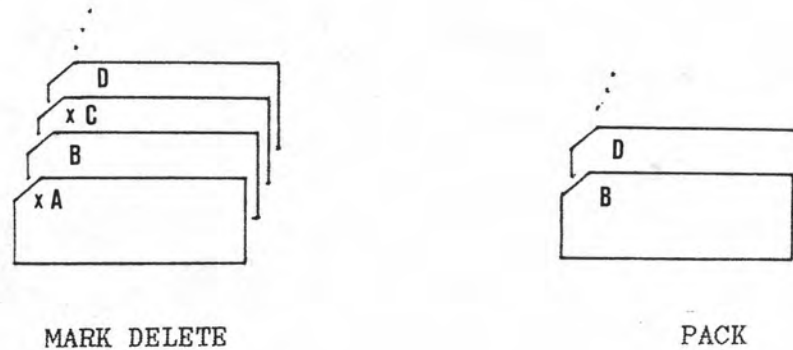
สำหรับการแก้ไขข้อมูลนั้น ก็เสมือนว่านำฟอรั่มเดิมที่มีอยู่แล้วในแฟ้มข้อมูลนั้นออกมา เมื่อแก้ไขเสร็จแล้วก็จะใส่แฟ้มข้อมูลนั้นกลับไปเคิม การนำฟอรั่มเดิมที่มีอยู่แล้วออกมานี้อาจจะโดยวิธีการสังคั่นหาหรือใช้การพลิกหน้าหาคุที่ละหน้าก็ได้

ส่วนแสดงและค้นหาข้อมูล เป็นส่วนช่วยในการจัดการแฟ้มข้อมูลให้เป็นไปอย่างสะดวก โดยปกติจะแสดงระเบียนสุดท้ายของแฟ้มข้อมูลเมื่อเรียกใช้โปรแกรม File Editor แล้วสามารถสังคั่นหน้าขึ้นลง เพื่อดูข้อมูลที่ละหน้า หรือให้กระโดดไปแสดงข้อมูลหน้าแรกหรือหน้าสุดท้ายก็ได้ โดยแต่ละหน้าหมายถึงแต่ละระเบียนในแฟ้มข้อมูลนั้น ในการค้นหาข้อมูล เป็นการสั่งให้แสดงระเบียนในแฟ้มข้อมูลที่มีคุณสมบัติเหมือนที่กำหนดไว้ โดยจะเริ่มค้นหาตั้งแต่หน้าที่แสดงอยู่ไปจนถึงหน้าสุดท้าย โดยการเปรียบเทียบข้อมูลในแต่ละระเบียน กับข้อมูลที่กำหนดให้ ถ้าตรงตามเงื่อนไขที่กำหนดทั้งหมดก็จะหยุดค้นที่ระเบียนที่ตรงตามเงื่อนไขนั้น



รูปที่ 3.12 แสดงการค้นหาข้อมูล

ส่วนลบข้อมูล เป็นการลบข้อมูลที่ไม่ต้องการออก มี 2 ชนิด คือ ชนิดทำเครื่องหมายว่าถูกลบไปแล้ว แต่ยังไม่หายไปจากแฟ้มข้อมูล และชนิดที่ลบออกไปจากแฟ้มข้อมูลเลย ชนิดที่ทำเครื่องหมายไว้จะสามารถนำข้อมูลกลับคืนมาได้ แต่ถ้าลบออกไปจากแฟ้มข้อมูลแล้วจะไม่สามารถนำข้อมูลกลับคืนมาได้ การลบระเบียนออกจากแฟ้มข้อมูลจะลบเฉพาะระเบียนที่ทำเครื่องหมายไว้เท่านั้น ดังรูปที่ 3.13 เมื่อทำเครื่องหมายที่ระเบียน A และ C



รูปที่ 3.13 แสดงการลบข้อมูล

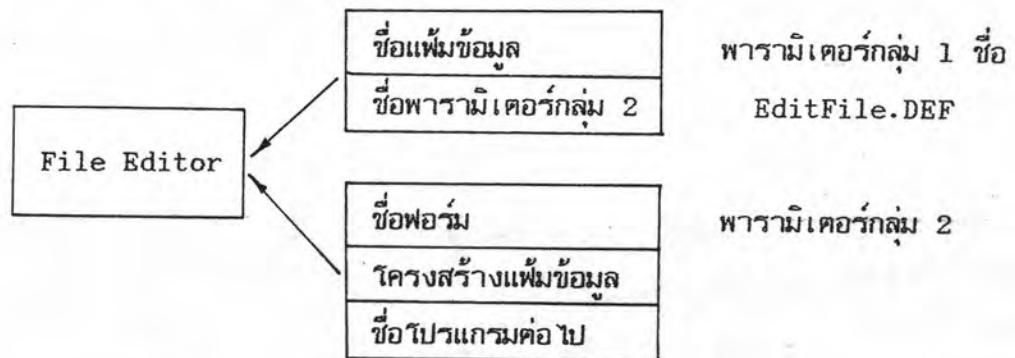
จากหน้าที่การทำงานของ File Editor กำหนดเป็นขั้นตอนการทำงานได้ดังนี้

3.3.2 ขั้นตอนการทำงานของ File Editor

0. อ่านตารางรหัสของระบบ
1. อ่านพารามิเตอร์ ชื่อเพิ่มข้อมูล, โครงสร้างเพิ่มข้อมูล, ชื่อฟอร์ม และชื่อโปรแกรมอื่นที่จะทำงานต่อไป
2. กำหนดโครงสร้างของระเบียบจากข้อมูลโครงสร้างเพิ่มข้อมูล
3. ถ้ามีเพิ่มข้อมูลที่กำหนด อ่านระเบียบสุดท้ายมาแสดงในฟอร์ม ถ้าไม่มีเพิ่มข้อมูลให้สร้างเพิ่มข้อมูลนั้นขึ้น และแสดงฟอร์มว่าง
4. จัดสถานะแวดล้อม และเรียกใช้ Screen Editor ในการแก้ไขระเบียบนั้น
5. ตรวจสอบการทำงานขึ้นต่อไปจากรหัสที่ส่งออกจาก Screen Editor ซึ่งเป็นแป้นควบคุมการทำงานที่นอกเหนือจากการควบคุมภายในของ Screen Editor
6. ทำงานตามการควบคุมที่กำหนดให้ แล้วไปทำขั้นที่ 4 อีก จนกว่าจะมีคำสั่งจบการทำงานของ File Editor
7. ถ้ามีชื่อโปรแกรมที่จะทำงานต่อไปก็ไปทำงานโปรแกรมนั้น มิฉะนั้นจบการทำงาน

คำอธิบายของการทำงานแต่ละขั้นตอนจะเป็นดังนี้

1. อ่านพารามิเตอร์ของ FILE EDITOR ซึ่งประกอบด้วย ชื่อเพิ่มข้อมูล, โครงสร้างเพิ่มข้อมูล, ชื่อฟอร์ม และชื่อโปรแกรมอื่นที่จะทำงานต่อไป (ถ้ามี) พารามิเตอร์ทั้ง 4 จะแบ่งออกเป็น 2 กลุ่ม คือ กลุ่มแรกประกอบด้วยชื่อเพิ่มข้อมูล และชื่อพารามิเตอร์ของกลุ่ม 2 สำหรับกลุ่ม 2 ประกอบด้วย ชื่อฟอร์ม, โครงสร้างเพิ่มข้อมูล และชื่อโปรแกรมที่จะทำงานต่อไป เรียงตามลำดับ พารามิเตอร์เหล่านี้เป็น Text File เก็บในเพิ่มข้อมูลชื่อ EDITFILE.DEF ดังรูปที่ 3.14 ถ้าไม่กำหนดจะให้ผู้ใช้อื่นเองที่ละตัวทางจอภาพ



รูปที่ 3.14 การจัดพารามิเตอร์ของ File Editor

2. หลังจากอ่านพารามิเตอร์แล้ว จะนำโครงสร้างเพิ่มข้อมูลมากำหนดโครงสร้างของระเบียบภายในเพิ่มข้อมูล จากการศึกษาโครงสร้างของระเบียบให้ข้อมูลย่อตัวแรกภายในระเบียบ เป็นสถานะการลบชนิดครรกะ และภายในฟอร์มกำหนดให้มีตำแหน่งเดิมข้อความพิเศษ 4 ตำแหน่ง โดย 3 ตำแหน่งแรก (คือ ตำแหน่งแสดงสถานะการลบ, ตำแหน่งแสดงหมายเลขระเบียบ และตำแหน่งแสดงข้อความบอกหมายเลขระเบียบที่ถูกแทรก) จะอยู่ก่อนหน้าช่องรับข้อมูล และมีตำแหน่งแสดงข้อมูลพิเศษอยู่ภายหลังช่องรับข้อมูลทั้งหมด ภายในระเบียบจริงจะมีสถานะการลบชนิดครรกะเพิ่มที่ตำแหน่งแรก จึงมีการคำนวณขนาด, ชนิด, ตำแหน่งเริ่มต้น และความยาวของข้อมูลย่อแต่ละตัวของระเบียบ ตัวอย่างเช่น พารามิเตอร์โครงสร้างเพิ่มข้อมูลของผลสอบนิสิต เป็น 'S7T23IC' โครงสร้างของข้อมูลที่เตรียมไว้สำหรับ Screen Editor จะเป็น 'S7T23ICBISS' และตารางกำหนดโครงสร้างของข้อมูลจะเป็นดังรูปที่ 3.15

ชนิดข้อมูล	ตำแหน่ง เริ่มต้น	ความยาว
B	1	1
S	2	8
T	10	31
I	41	2
C	43	1

ความยาว = 44 ไบต์

รูปที่ 3.15 ตารางกำหนดโครงสร้างของข้อมูล

และสร้างตัวเก็บผลชั่วคราวในหน่วยความจำจำนวน 4 ชุดขนาดเท่ากับความยาวของระเบียบ ใช้เป็นที่เก็บผลชั่วคราวของระเบียบที่ใช้งานอยู่ (Work), ระเบียบที่ส่งให้เก็บสำหรับการซ้ำข้อมูล (Copy), ระเบียบสำหรับการค้นหา (Spec) และระเบียบสำหรับช่วยในการสลับข้อมูล (Backup) รายละเอียดของตัวเก็บผลชั่วคราวเหล่านี้จะได้อธิบายต่อไป

3. จากพารามิเตอร์ชื่อเพิ่มข้อมูล ทำการเปิดเพิ่มข้อมูลนั้น ถ้าไม่มีเพิ่มข้อมูลที่กำหนดก็ให้สร้างขึ้นมาใหม่ ระเบียบเริ่มต้นเป็นระเบียบว่าง มิฉะนั้นให้เลื่อนตำแหน่งตัวชี้ระเบียบในเพิ่มข้อมูลนั้นไปที่ระเบียบท้ายสุด เพื่ออ่านระเบียบสุดท้ายเข้ามา

4. จัดสถานะแวดล้อมให้แก่ Screen Editor คือชื่อฟอร์ม, ลักษณะโครงสร้างข้อมูลที่ตรงกับฟอร์ม และจัดข้อมูลภายในระเบียบให้ตรงกับลักษณะข้อมูลที่ใช้ใน Screen Editor รวมทั้งกำหนดข้อความพิเศษที่ต้องการแสดง สำหรับเพิ่มข้อมูลใหม่จะแสดงระเบียบว่างมีหมายเลขระเบียบเป็น 1 และอยู่ในสถานะเพิ่มเติมระเบียบใหม่ ถ้าแสดงระเบียบสุดท้ายก็จะอยู่ในสถานะแก้ไขระเบียบนั้น ให้อยู่ในการควบคุมของ Screen Editor

5. เมื่อออกจากการควบคุมของ Screen Editor (มีคำสั่งนอกการควบคุมของ Screen Editor ก็จะส่งการควบคุมนั้นให้ File Editor ทำงานต่อ โดยตรวจสอบจากรหัสควบคุมการทำงานเพื่อไปทำตามการทำงานต่าง ๆ ที่ได้เตรียมไว้คือ

- บันทึกลง [F2]
- จบการทำงาน [ESC]
- พลิกหน้าขึ้น [Pg Up]
- พลิกหน้าลง [Pg Dn]

- พลิกลงไปหน้าสุดท้าย [End]
- พลิกลงไปหน้าแรก [Home]
- กำหนดตำแหน่งแทรก [Shift-F7]
- สั่งแทรก [Ctrl-F7]
- สั่งเปลี่ยนสถานะการลบ [Shift-F9]
- สั่งลบระเบียนที่มีสถานะลบ [Ctrl-F9]
- สั่งแสดงระเบียนที่จำ [Shift-F5]
- สั่งจำระเบียน [Ctrl-F5]
- สั่งรับเงื่อนไขการค้นหา [Shift-F3]
- สั่งเริ่มค้น [Ctrl-F3]

เป็นควบคุมการสั่งงานกำหนดอยู่ข้างหลังคำสั่งนั้น

6. ถ้าไม่ใช่คำสั่งจบการทำงานก็จะไปทำงานตามการควบคุมต่าง ๆ ที่กำหนดให้ ซึ่งเป็นรหัสควบคุมการทำงานภายในของ File Editor นอกเหนือจากนี้แล้วจะไม่ทำงาน การทำงานต่าง ๆ เป็นดังนี้

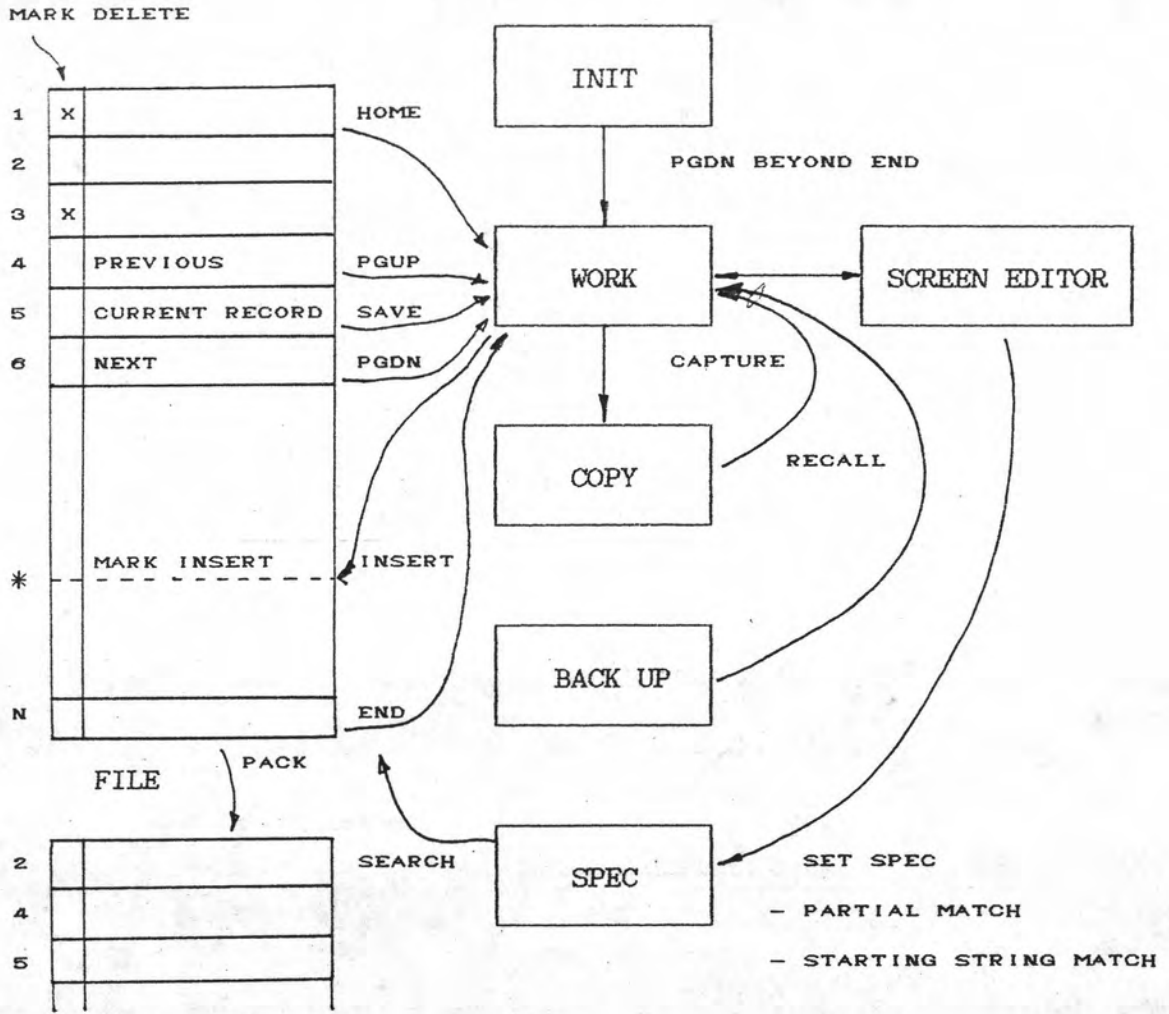
- บันทึก (Save) นำระเบียนที่ได้แก้ไขเรียบร้อยแล้วในหน่วยความจำทำงาน (WORK) บันทึกลงที่ระเบียน ที่ตัวชี้ตำแหน่งระเบียนขณะนั้น แล้วไปเรียกคำสั่งพลิกหน้าลง

- จบการทำงาน (Exit) คำสั่งให้จบจาก File Editor กลับสู่ MS-DOS prompt

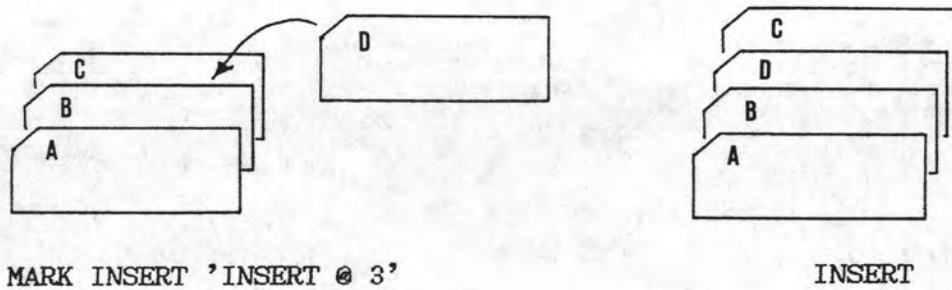
- พลิกหน้าขึ้น (Page Up) คำสั่งนี้ให้เลื่อนตัวชี้ตำแหน่งระเบียนในแฟ้มข้อมูลลงหนึ่งตำแหน่ง แต่ไม่ต่ำกว่าระเบียนแรกในแฟ้มข้อมูลนั้น แล้วอ่านระเบียนนั้นมาที่หน่วยความจำทำงาน (WORK)

- พลิกหน้าลง (Page Down) คำสั่งนี้ให้เลื่อนตัวชี้ตำแหน่งระเบียนในแฟ้มข้อมูลเพิ่มขึ้นหนึ่งตำแหน่ง แต่ไม่เกินระเบียนสุดท้ายของแฟ้มข้อมูลนั้น ก็จะอ่านระเบียนที่ถูกชี้มาที่หน่วยความจำทำงาน (WORK) ซึ่งเป็นการตรวจสอบหรือแก้ไขระเบียน แต่ถ้ายังสั่งพลิกหน้าลงอีกจนเกินระเบียนสุดท้าย ก็จะแสดงระเบียนว่างออกมา แล้วจะสั่งพลิกต่อไปไม่ได้อีกแล้ว

- พลิกไปหน้าสุดท้าย (End) เลื่อนตัวชี้ตำแหน่งระเบียบไปที่ระเบียบท้ายสุดของแฟ้มข้อมูลนั้น แล้วอ่านระเบียบนั้นมาที่หน่วยความจำทำงาน
- พลิกไปหน้าแรก (Home) เลื่อนตัวชี้ตำแหน่งระเบียบไปที่ระเบียบแรกสุดของแฟ้มข้อมูลนั้น แล้วอ่านระเบียบนั้นมาที่หน่วยความจำทำงาน
- กำหนดตำแหน่งแทรก (Mark Insert) จำค่าตัวชี้ตำแหน่งระเบียบในขณะนั้น เพื่อกำหนดตำแหน่งแทรก ถ้ามีคำสั่งให้กำหนดตำแหน่งแทรกอีกครั้ง จะเปลี่ยนตำแหน่งแทรกไปที่ตำแหน่งระเบียบใหม่ แต่ถ้าตำแหน่งระเบียบตรงกับที่ได้จำตำแหน่งไว้ก่อนจะยกเลิกการกำหนดตำแหน่งแทรกเดิม ในขณะที่มีการกำหนดตำแหน่งแทรกจะปรากฏข้อความ 'Insert @ Number' ที่ตำแหน่งแสดงข้อความพิเศษตัวที่ 3 การทำงานนี้คู่กับการสั่งแทรก
- สั่งแทรก (Insert) นำระเบียบที่ปรากฏอยู่บนหน้าจอภาพในขณะนั้นไปแทรกยังตำแหน่งระเบียบที่ได้กำหนดตำแหน่งแทรกไว้ล่วงหน้าแล้ว และเลื่อนระเบียบที่อยู่ข้างหลังตำแหน่งแทรกนั้นไปอีก 1 ตำแหน่ง
- สั่งเปลี่ยนสถานะการลบ (Mark Delete) ในแต่ละระเบียบจะมีข้อมูลย่อยชนิดตรรกะที่แสดงสถานะการลบของระเบียบนั้น การสั่งเปลี่ยนสถานะการลบนี้เป็นชนิดสลับกัน (Toggle) คือ ถ้าสถานะการลบเป็นจริงจะถูกเปลี่ยนเป็นเท็จ แต่ถ้าเป็นเท็จอยู่แล้วจะถูกเปลี่ยนเป็นจริง ถ้าสถานะการลบเป็นจริงจะปรากฏเครื่องหมาย 'X' ที่ตำแหน่งแสดงข้อความพิเศษตำแหน่งแรก
- สั่งลบระเบียบ (Pack) ทำการลบระเบียบที่มีสถานะการลบเป็นจริงทุกระเบียบออกจากแฟ้มข้อมูล ทำให้ขนาดของแฟ้มข้อมูลลดลง และหมายเลขระเบียบแต่ละระเบียบจะเปลี่ยนไป การสั่งแบบนี้จะไม่สามารถนำข้อมูลที่ถูกลบไปแล้วกลับคืนมาได้
- สั่งจำระเบียบ (Capture) นำระเบียบที่ปรากฏอยู่ที่หน้าจอภาพไปเก็บที่หน่วยความจำสำรองอันหนึ่งชื่อ Copy
- สั่งแสดงระเบียบที่จำ (RECALL) นำระเบียบที่เก็บที่หน่วยความจำสำรอง Copy มาแทนที่ระเบียบที่ปรากฏบนจอภาพ โดยจะเก็บระเบียบที่ปรากฏบนจอภาพนั้นไว้ก่อนที่หน่วยความจำสำรอง BACKUP ถ้าสั่งแสดงระเบียบที่จำอีกครั้ง ระเบียบที่ปรากฏบนจอภาพจะโดนแทนที่อีกด้วยระเบียบเดิมที่ได้เก็บไว้ก่อนล่วงหน้าแล้ว



รูปที่ 3.16 แสดงหน่วยความจำที่ใช้ และการทำงานของ File EDITOR



รูปที่ 3.17 แสดงการแทรกข้อมูล

- **สั่งรับเงื่อนไขการค้น (Set spec)** เมื่อสั่งรับเงื่อนไขการค้น จะปรากฏฟอร์มว่างขึ้นมา เพื่อให้ใส่ข้อความที่ต้องการค้นหาในช่องใส่ข้อความตามที่ได้กำหนดไว้ และมีข้อความปรากฏที่ตำแหน่งแสดงข้อความพิเศษที่ 4 ว่า 'To accept spec, Press Shift-F3' เพื่อเตือนให้กดแป้นพิมพ์ที่กำหนดซึ่งก็คือ แป้นพิมพ์สั่งรับเงื่อนไขการค้นนั่นเอง หลังจากที่ใส่ข้อความที่ต้องการค้นหาเรียบร้อยแล้ว แล้วนำระเบียบนั้นไปเก็บที่หน่วยความจำ SPEC

- **สั่งเริ่มค้น (Seach)** การค้นหานั้นจะทำการเริ่มค้นหาตั้งแต่ตำแหน่งระเบียบปัจจุบัน ไปจนจบแฟ้มข้อมูล หรือค้นพบ ส่วนที่จะค้นหาได้นั้นจะเป็นเฉพาะส่วนที่เป็นข้อความเท่านั้น จากหน่วยความจำ SPEC การเปรียบเทียบข้อความจะมีได้ 2 ชนิด คือ ชนิดเปรียบเทียบตรงกันบางส่วนที่ตำแหน่งใด ๆ และชนิดเปรียบเทียบจากตำแหน่งแรกของข้อความเท่านั้น การเปรียบเทียบทั้ง 2 ชนิดขึ้นกับรูปแบบของข้อความที่กำหนดไว้เมื่อสั่งรับเงื่อนไขการค้น การใส่ข้อความที่ต้องการค้นหานั้นสามารถใส่ได้หลายข้อความ ซึ่งจะเป็นการเปรียบเทียบข้อความทั้งหมด ถ้าผลที่ได้เป็นจริงทุกข้อความก็จะเลื่อนตัวชี้ตำแหน่งระเบียบไปที่ระเบียบนั้น และแสดงระเบียบนั้นบนจอภาพ ถ้าสั่งค้นอีกจะหมายถึงการค้นหาข้อความตามเงื่อนไขเดิมต่อไปอีก (หรือจะ ไปกำหนดเงื่อนไขใหม่แล้วมาสั่งค้นก็สามารถทำได้)

3.3.3 การใช้งาน File Editor

ก่อนที่จะใช้งาน File Editor ต้องทราบชื่อแฟ้มข้อมูลที่ต้องการจัดการ, รหัสโครงสร้างของแฟ้มข้อมูลนั้น, ชื่อฟอร์มที่สัมพันธ์กับแฟ้มข้อมูลที่ต้องการจัดการ และโปรแกรมที่จะทำงานต่อไป โดยฟอร์มนั้นต้องสร้างมาก่อนด้วยโปรแกรม Screen Generator ข้อมูลทั้งหมดนี้เราจะเรียกว่า พารามิเตอร์ โดยแบ่งออกเป็น 2 กลุ่ม

กลุ่มแรกชื่อ EDITFILE.DEF ประกอบด้วยชื่อแฟ้มข้อมูล และชื่อพารามิเตอร์กลุ่มที่ 2

กลุ่มที่ 2 ชื่ออะไรก็ได้ ประกอบด้วย ชื่อฟอร์ม, โครงสร้างของแฟ้มข้อมูล และชื่อโปรแกรมที่จะทำงานคือ

พารามิเตอร์ทั้ง 2 กลุ่มนี้สร้างขึ้นโดยใช้ Editor ใด ๆ ก็ได้ โดยเป็น Text File ข้อมูลละบรรทัด

เมื่อเราสั่งโปรแกรม File Editor ที่ชื่อ GFILE ที่ prompt ของ MS-DOS โปรแกรมจะทำการตรวจหาแฟ้ม EditFile.DEF ถ้ามีแฟ้มนี้อยู่ โปรแกรมก็จะอ่าน พารามิเตอร์ที่เหลือ แต่ถ้าไม่มีโปรแกรมจะสอบถามบนจอภาพ แล้วเข้าสู่การทำงานกับแฟ้มข้อมูล สำหรับวิธีการใช้งานภายใน File Editor อย่างละเอียดดูที่ภาคผนวก ก.

ตัวอย่าง การใช้งาน File Editor เช่นในการเก็บข้อมูลทะเบียน ประวัติคนเสียค่า ในแฟ้มข้อมูลชื่อ OLDSTUD.DAT ซึ่งประกอบด้วยข้อมูลย่อย

รหัส ชนิดข้อความอักษรตัวใหญ่ 7 คอลัมน์

ชื่อ ชนิดข้อความไทย 20 คอลัมน์

ที่ทำงาน ชนิดข้อความไทย 30 คอลัมน์ 2 บรรทัด

โทรศัพท์ ชนิดข้อความ 12 คอลัมน์

สิ่งแรกที่ต้องสร้างขึ้นคือ การสร้างฟอร์มขึ้นที่หน้าจอภาพเพื่อใช้ในการ แสดง ซึ่งสร้างโดยเรียกใช้ Screen Generator โดยวิธีใช้ในหัวข้อ 3.4.3 เพื่อสร้างหน้า จอภาพดังรูปที่ 3.1 ที่ได้แสดงไว้แล้ว สมมติเก็บลงในแฟ้มข้อมูลชื่อ OLDSTUD.SCN และ OLDSTUD.SEP

หลังจากนั้นกำหนดค่าเป็นพารามิเตอร์ โดยสร้างพารามิเตอร์ตัวแรกใน แฟ้มข้อมูลชื่อ EDITFILE.DEF เป็นชนิดข้อความด้วยโปรแกรมประมวลข้อความอะไรก็ได้ โดย ที่บรรทัดแรก เป็นชื่อแฟ้มข้อมูล OLDSTUD.DAT และบรรทัดที่ 2 เป็นชื่อพารามิเตอร์กลุ่มที่ 2 สมมติให้ชื่อ OLDSTUD.PRM

สร้างพารามิเตอร์กลุ่มที่ 2 เป็นชนิดข้อความเช่นกัน ในแฟ้มข้อมูลชื่อ OLDSTUD.PRM โดยที่บรรทัดแรก เป็นชื่อฟอร์ม OLDSTUD.SEP บรรทัดที่ 2 เป็นรหัสกำหนด โครงสร้างข้อมูลย่อย และช่องรับข้อมูล ในที่นี้เป็น 'U7T20T30T30S12' บรรทัดที่ 3 เป็นชื่อ ของโปรแกรมที่ต้องการให้ทำงานคือ ไปหลังจากจบการทำงานของ File Editor แล้ว โปรแกรมที่จะทำงานต่อไปนั้น ต้องเป็นโปรแกรมที่เกิดขึ้นจากเทอร์โมพลาสติกเท่านั้น ชื่อ โปรแกรมนี้ อาจไม่มีก็ได้ ถ้าไม่มีเมื่อจบการทำงานจาก File Editor จะกลับเข้าสู่ระบบ ปฏิบัติการต่อไป

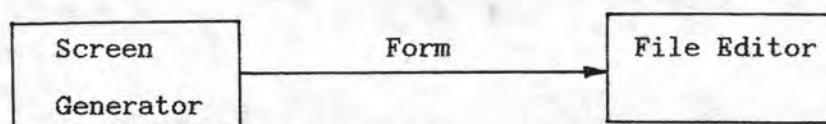
3.4 Screen Generator

ในหัวข้อนี้จะกล่าวถึง หน้าที, ขั้นตอนการทำงาน และการใช้งานของ Screen Generator ตามข้อกำหนดของโปรแกรมในหัวข้อ 3.2

3.4.1 หน้าที

จากหัวข้อ 3.2 ที่ได้กล่าวถึงข้อกำหนดของโปรแกรม และฟอร์ม และในหัวข้อ 3.3 ที่กล่าวถึงโปรแกรม File Editor สิ่งสำคัญก่อนที่จะใช้งาน File Editor ได้ต้องมีฟอร์มที่แทนระเบียบของแฟ้มข้อมูลเสียก่อน จึงได้แยกส่วนของโปรแกรมในการสร้างฟอร์มออกจากส่วนของ File Editor เพื่อให้มีความยืดหยุ่นในการใช้งาน ทั้งนี้ฟอร์มที่ได้ต้องเป็นไปตามข้อกำหนดของฟอร์มที่ได้กล่าวไว้แล้วในหัวข้อ 3.2

ดังนั้นหน้าที่ของ Screen Generator จึงมีหน้าที่หลักเพียงอย่างเดียวคือ สร้างฟอร์มขึ้นมา และมีส่วนประกอบอื่นๆ ที่ช่วยเหลือในการสร้างและแก้ไขฟอร์มนี้ให้ง่ายขึ้น การสร้างฟอร์มนั้นเป็นแบบ Full Screen Edit คือ สามารถวางข้อความต่างๆ ที่ตำแหน่งใดก็ได้ในขอบเขตที่กำหนด โดยมีแป้นควบคุมการเลื่อนเคอร์เซอร์เพื่อไปที่ใดๆ ภายในฟอร์มนั้น ฟอร์มที่สร้างขึ้นมีข้อความได้ทั้งไทยและอังกฤษ และมีสัญลักษณ์พิเศษที่ใช้กำหนดเป็นช่องใส่ข้อมูล และตำแหน่งข้อความพิเศษ หลังจากจัดวางตำแหน่งข้อความ และสัญลักษณ์ บนจอภาพได้ตามต้องการแล้ว โปรแกรม Screen Generator ก็จะทำการเก็บคุณสมบัติของฟอร์มนั้นลงในแฟ้มข้อมูล เพื่อให้ File Editor นำไปใช้งานต่อไป

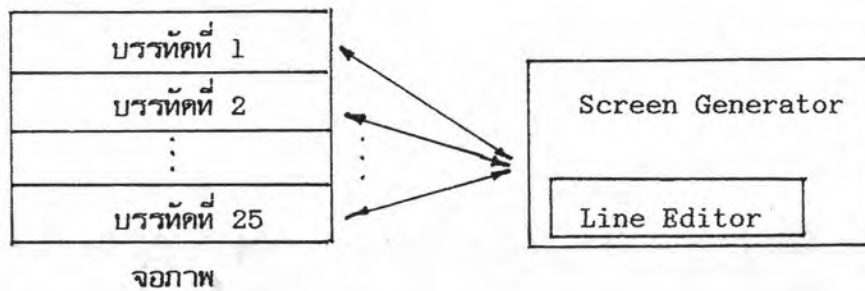


รูปที่ 3.18 แสดงความสัมพันธ์ระหว่าง Screen Generator และ File Editor

ก่อนจะกล่าวถึงขั้นตอนการทำงานจะกล่าวถึงองค์ประกอบของ Screen Generator ก่อน คือ หน่วยความจำที่ใช้ เนื่องจากเรากำหนดให้ฟอร์มมีขนาดคงที่คือ เท่ากับ หน้าจอภาพที่เห็น เพราะเราให้ฟอร์มหนึ่งแทนระเบียบหนึ่งๆ ในแฟ้มข้อมูล ซึ่งโดยปกติในระบบ

ฐานข้อมูลแบบสัมพันธ์ (Relational database) แต่ละระเบียบจะมีข้อมูลอยู่ไม่มากนัก จึงสามารถกำหนดให้ฟอร์มเท่ากับจอภาพได้

เราจะมองให้จอภาพเป็นข้อความหลายบรรทัดติดต่อกัน โดยมีความสามารถในการเลื่อนเคอร์เซอร์ไปที่ข้อความใดและตำแหน่งใดในข้อความนั้นได้ การเพิ่มหรือลบตัวอักษรของข้อความในบรรทัดนั้น จึงเป็นเพียงการเปลี่ยนแปลงข้อความในบรรทัดเท่านั้น ซึ่งคุณสมบัติการแก้ไขข้อความนี้จะอยู่ในส่วนเครื่องมือทางซอฟต์แวร์ Line Editor ที่จะกล่าวถึงในบทที่ 4 (หัวข้อ 4.2)



รูปที่ 3.19 แสดงความสัมพันธ์ระหว่าง Screen Generator และ Line Editor

นอกจากนี้ยังมีส่วนติดต่อกะหว่างหน่วยความจำที่เก็บข้อความต่าง ๆ กับเพิ่มข้อมูล เพื่อที่จะเก็บคุณสมบัติของฟอร์มที่สร้างขึ้น หรือใช้อ่านคุณสมบัติของฟอร์มนั้นกลับมาที่หน่วยความจำเพื่อแสดงบนจอภาพ

การที่ Screen Generator ใช้สร้างฟอร์มนั้น ไม่ได้ใช้เฉพาะกับ File Editor เท่านั้น ยังใช้เป็นส่วนร่วมในเครื่องมือทางซอฟต์แวร์ Screen Editor ที่จะกล่าวถึงในบทที่ 4 (หัวข้อ 4.3) โดยที่ File Editor จะจำกัดรูปแบบของฟอร์มมากกว่าของ Screen Editor

นอกจากนี้ ฟอร์มที่สร้างขึ้นยังใช้เป็นตัวกำหนดรูปแบบของรายงาน ซึ่งถูกสร้างโดยโปรแกรม Report Generator (แต่ในวิทยานิพนธ์ฉบับนี้ ไม่สามารถกล่าวถึงรายละเอียดของโปรแกรม Report Generator ได้)

ในหัวข้อต่อไปจะกล่าวถึงขั้นตอนการทำงานอย่างละเอียดของ Screen Generator

3.4.2 ขั้นตอนการทำงานของ Screen Generator

0. อ่านตารางรหัสของระบบเข้ามา
1. แสดงรายชื่อแฟ้มข้อมูลที่มีอยู่ขณะนั้น (มีส่วนขยายเป็น '.SCN') และเนื้อที่ว่างที่เหลือในดิสก์ (disk)
2. แสดงชนิดการทำงานให้เลือก 3 ข้อ
3. เลือกทำงานตามการทำงานที่เลือกไว้ เมื่อทำงานจบแล้วจะถามว่าต้องการทำงานต่อหรือไม่
4. กลับไปทำงานตั้งแต่ข้อ 1 ใหม่จนกว่าจะ ไม่ต้องการทำงานนี้อีก

คำอธิบายการทำงานของแต่ละขั้นตอนจะเป็นดังนี้

1. แสดงรายชื่อแฟ้มข้อมูลที่มีอยู่ขณะนั้น และเนื้อที่ว่างที่เหลือในดิสก์ การแสดงรายชื่อนี้จะมีลักษณะการทำงานเช่นเดียวกับคำสั่ง 'DIR/P' ของ MS-DOS [20]
2. แสดงชนิดการทำงานให้เลือก 3 ข้อ ดังรูปที่ 3.20 แล้วรอรับการเลือก โดยกดแป้นพิมพ์ตามหมายเลข 1,2 หรือ 3 แล้วกดแป้น [Enter]

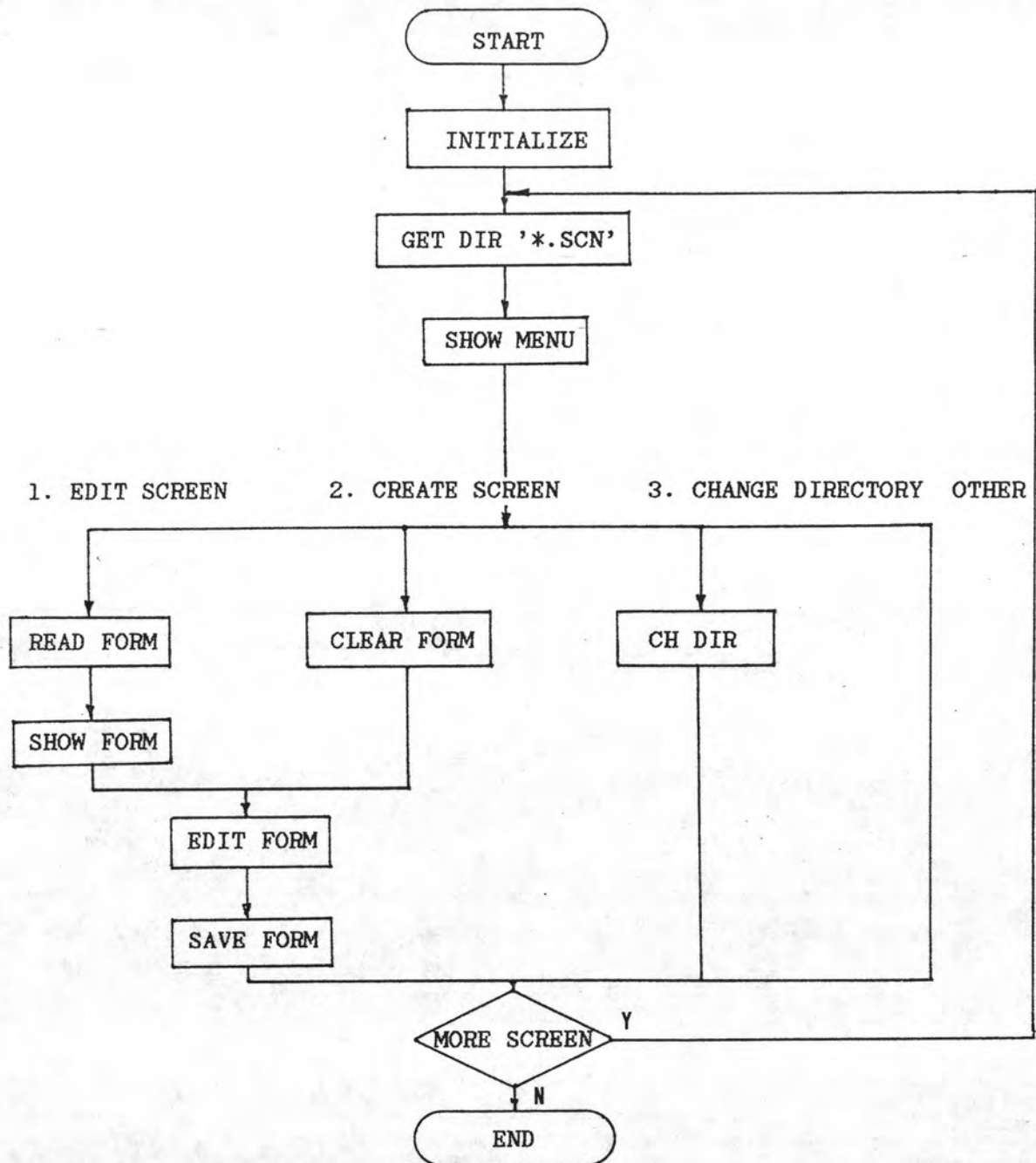
หมายเลขการทำงานเป็นดังนี้ คือ

1. สำหรับแก้ไขฟอร์มที่ได้สร้างไว้แล้ว
2. สร้างฟอร์มขึ้นมาใหม่จากหน้าว่าง
3. เปลี่ยนเส้นทาง (path) การจัดเก็บแฟ้มข้อมูล

<ol style="list-style-type: none"> 1. Edit Screen 2. Create Screen 3. Change Directory <p>Select (1-3) ?</p>

รูปที่ 3.20 แสดงหน้าจอภาพการทำงาน

3. ทำงานตามที่กำหนดคือ ถ้าเป็นการแก้ไขฟอร์มที่ได้สร้างไว้แล้ว จะถามชื่อฟอร์มนั้นเพื่อนำมาแสดงและแก้ไข ถ้าเป็นการสร้างฟอร์มขึ้นมาใหม่จากหน้าว่าง ก็จะไม่



รูปที่ 3.21 แผนภูมิการทำงานของ Screen Generator

ถามชื่อฟอร์ม แต่จะให้หน้าว่างมาเลย หลังจากทำการแก้ไขฟอร์มแล้ว ก็จะไม่ออกจากการแก้ไข ฟอร์มนั้นด้วยเป็นควบคุมที่กำหนดไว้ การทำงานต่าง ๆ ในขั้นนี้จะเป็นไปตามรูปที่ 3.21

รายละเอียดส่วนประกอบย่อยของขั้นตอนการทำงาน เป็นดังนี้

- Read Form เป็นการอ่านรูปแบบของฟอร์มมาจากแฟ้มข้อมูล โดยปกติจะหมายถึงแฟ้มข้อมูลที่มีส่วนขยายเป็น '.SCN' ที่รูปแบบการเก็บตามหัวข้อย่อยที่ 3.2.5 โดยอ่านมาไว้ที่หน่วยความจำ แบ่งเป็นบรรทัดเท่ากับขนาดจอภาพ

- Show Form ทำการแสดงข้อมูลในหน่วยความจำของแต่ละบรรทัดบนจอภาพ

- Clear Form ข้อมูลในหน่วยความจำจะว่าง และแสดงข้อมูลว่างบนจอภาพ

- Edit Form ส่วนนี้จะเรียกใช้ Line Editor ในการแก้ไขข้อความแต่ละบรรทัด โดยควบคุมการเลื่อนเคอร์เซอร์ขึ้นลง แทรกบรรทัด และลบบรรทัดเป็นควบคุมอื่นๆ นอกจากนี้เป็นการจบการทำงานการแก้ไขฟอร์ม

- Save Form ทำการแปลงข้อมูลในหน่วยความจำ เขียนลงในแฟ้มข้อมูล 2 รูปแบบ ตามโครงสร้างข้อมูลส่วนที่เกี่ยวกับฟอร์มตามข้อกำหนดในหัวข้อย่อย 3.2.5

- Ch Dir เป็นคำสั่งให้เปลี่ยน directory การทำงานขณะนั้น เพื่อให้สามารถแสดงแฟ้มข้อมูลของฟอร์มใน directory นั้นได้

4. สอบถามว่าจะทำงานอีกหรือไม่ โดยมีข้อความว่า 'More Screen (Y/N) ?' ถ้าตอบว่า 'Y' ก็จะไปขึ้นที่ 1 ใหม่ ถ้าไม่ใช้กลับไป MS-DOS

3.4.2 การใช้งาน

สำหรับรายละเอียดการใช้งานนี้ ได้รวมอยู่ที่ภาคผนวก ข.

3.5 สรุป

ในบทนี้ได้กล่าวถึงส่วนของโปรแกรม File Editor และ Screen Generator อย่างละเอียด โดยได้กล่าวถึงข้อกำหนดของฟอร์มและโครงสร้างข้อมูลที่ใช้ รวมทั้งโครงสร้างภายในแฟ้มข้อมูลที่ใช้ การกำหนดพารามิเตอร์ให้แก่ File Editor และวิธีใช้งานของ File Editor และ Screen Generator

ภายใน File Editor จะประกอบด้วยส่วนจัดการแฟ้มข้อมูล และส่วนจัดการฟอร์ม

โดยที่ส่วนจัดการเพิ่มข้อมูลจะใช้พารามิเตอร์ที่กำหนดชื่อเพิ่มข้อมูล และโครงสร้างเพิ่มข้อมูลนั้น
ไปใช้ในการอ่านระเบียบ และส่งให้ส่วนจัดการฟอร์มที่ใช้พารามิเตอร์ที่กำหนดชื่อฟอร์ม และ
โครงสร้างเพิ่มข้อมูลที่ตรงกับฟอร์มนั้น แสดงระเบียบนั้นบนจอภาพ รวมทั้งสามารถรับและแก้ไข
ตรวจสอบความถูกต้องของระเบียบได้ โดยที่ในบทนี้ได้กล่าวถึงส่วนจัดการเพิ่มข้อมูล สำหรับ
ส่วนจัดการฟอร์มจะเป็นหน้าที่ของเครื่องมือทางซอฟต์แวร์ Screen Editor ได้กล่าวถึงอย่าง
ละเอียดในหัวข้อ 4.3

สำหรับ Screen Generator ก็มีหน้าที่ในการสร้างและแก้ไขฟอร์มเท่านั้นเพื่อส่ง
ให้แก่ File Editor ไปใช้โดยที่ Screen Generator ประกอบด้วยส่วนแปลงข้อมูลในเพิ่ม
ข้อมูลกับจอภาพ และส่วนจัดการข้อความในบรรทัด ส่วนแปลงข้อมูลกับจอภาพเราได้กล่าวถึง
การจัดข้อมูลในเพิ่มข้อมูลของฟอร์มอย่างละเอียด โดยส่วนจัดการข้อความในแต่ละบรรทัดจะ
เป็นหน้าที่ของเครื่องมือทางซอฟต์แวร์ Line Editor ซึ่งจะกล่าวอย่างละเอียดในหัวข้อ 4.2