



เอ็ดิตเตอร์แบบเต็มจอภาพ "PE" สำหรับไมโครคอมพิวเตอร์
A Full Screen Text Editor "PE" For Microcomputer

จีรวุฒ มุณินทรนพมาศ
Jeerawoot Muninnoppamas

เลขหมู่	๐ ๐๙๕๖.๙๖.๗๒๙ ๗๖๔ ๒๕๓๙ ๑. ๒
Bib Key	๒๐๔๘๖๖
	๑. ๕. S.A. ๒๕๓๓

วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

Master of Science Thesis in Computer Science

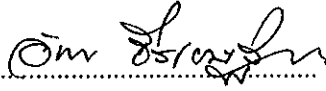
Prince of Songkla University

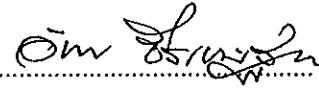
2539


ชื่อวิทยานิพนธ์ เอดิเตอร์แบบเต็มจอภาพ “PE” สำหรับไมโครคอมพิวเตอร์
ผู้เขียน นายจีรวุฒ มุรินทร์นพมาศ
สาขาวิชา วิทยาการคอมพิวเตอร์


คณะกรรมการที่ปรึกษา

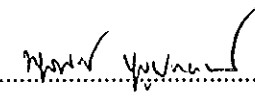
คณะกรรมการสอบ

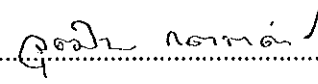
 ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อัชฌา ถีรเชษฐมงคล)

 ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อัชฌา ถีรเชษฐมงคล)

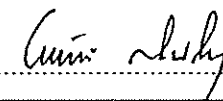
 กรรมการ
(อาจารย์นันทกานต์ สุเมธสิทธิกุล)

 กรรมการ
(อาจารย์นันทกานต์ สุเมธสิทธิกุล)

 กรรมการ
(รองศาสตราจารย์ ดร.พูลพงษ์ บุญพราหมณ์)

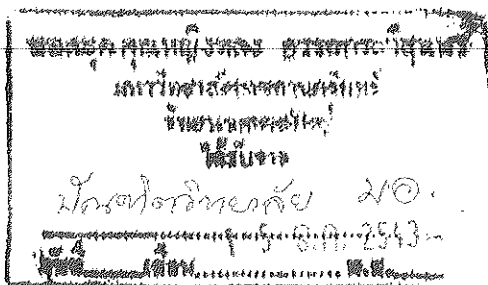
 กรรมการ
(อาจารย์ วุฒิพงศ์ เตชะดำรงสิน)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้แนบวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์



(ดร.ไพรัตน์ สงวนไพร)

คณบดีบัณฑิตวิทยาลัย



ชื่อวิทยานิพนธ์ เอดิเตอร์แบบเต็มจอภาพ “PE” สำหรับไมโครคอมพิวเตอร์
ผู้เขียน นายจรัญ มุนินทร์นพมาศ
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2538

บทคัดย่อ

เทกซ์เอดิเตอร์ หรือ เอดิเตอร์ เป็นโปรแกรมระบบที่เกิดขึ้นมาพร้อมกับการใช้งานคอมพิวเตอร์ผ่านทางเทอร์มินัลของระบบเครื่องคอมพิวเตอร์แบบเมนเฟรมและแบบมินิคอมพิวเตอร์เพื่อใช้ในการจัดทำโปรแกรมต้นฉบับด้วยภาษาแอสเซมบลีหรือภาษาคอมพิวเตอร์ระดับสูงทั่วไป นอกจากนั้น ในการพิมพ์เอกสารใดๆด้วยคอมพิวเตอร์ เอดิเตอร์ก็จะช่วยให้ผู้พิมพ์ไม่ต้องแก้ไขเอกสารที่พิมพ์ด้วยการพิมพ์เอกสารนั้นใหม่ทั้งหมดเหมือนการใช้เครื่องพิมพ์ดีด ผู้ใช้สามารถแทรกข้อความใหม่หรือบรรทัดใหม่ และสามารถลบข้อความหรือบรรทัดออกจากเอกสารตามที่ต้องการได้อย่างสะดวก

เอดิเตอร์ “PE” เป็นเอดิเตอร์แบบเต็มจอภาพสำหรับไมโครคอมพิวเตอร์ซึ่งสามารถใช้ได้ทั้งภาษาไทยและภาษาอังกฤษ การติดต่อกับผู้ใช้เป็นแบบรายการเลือกแบบดิ่งลง นอกจากนี้ จะมีความสามารถขั้นพื้นฐานของเอดิเตอร์แบบเต็มจอภาพทั่วไปซึ่งได้แก่ การเลื่อนตำแหน่งเคอร์เซอร์ การค้นหาและแทนที่ข้อความ การแทรกและแทนที่อักขระ/ข้อความ การลบตัวอักขระ/ข้อความ และการทำงานเกี่ยวกับบล็อกแล้ว ยังมีความสามารถในการจัดการเกี่ยวกับเพิ่มข้อมูลซึ่งได้แก่ การสร้างเพิ่มข้อมูลใหม่ การจัดเก็บข้อความลงเพิ่มข้อมูล การลบเพิ่มข้อมูลเก่า การจัดพิมพ์ข้อความภายในเพิ่มข้อมูล และการเปลี่ยนชื่อเพิ่มข้อมูลเก่าด้วย

Thesis Title A Full Screen Text Editor "PE" For Microcomputer
Author Mr.Jeerawoot Muninnoppamas
Major Program Computer Science
Academic Year 1995

- Abstract

A "text editor" or an "editor" is a system program created for a mainframe computer or minicomputer. This program is used to write any assembly language programs or other high level computer language programs. Moreover, to create any documents, an editor helps users to edit their documents more efficiently than a typewriter for which editing takes the form of retyping the complete document. By using an editor, user can append a new message, delete an undesired message and can move a message anywhere in the document easily but users themselves must reformat the document for the desired presentation.

The editor program in this project can use both Thai and English languages, and has a pulldown menu for an interface between the computer and the user. It not only has full screen basic editor abilities such as cursor movements, word finding and replacing, character insertion or replacing, character deletion and block managements, but also has file management abilities such as file creation, file saving, file deletion, file printing and file rename.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้ด้วยดีด้วยความช่วยเหลือ และสนับสนุนจากบุคคลหลายฝ่าย ซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูงมา ณ โอกาสนี้ คือ

ผู้ช่วยศาสตราจารย์ ดร.รัชดา วีระเชษฐมงคล อาจารย์ที่ปรึกษาที่ได้กรุณาให้คำปรึกษาความรู้ และข้อเสนอแนะทางวิชาการ พร้อมทั้งตรวจทาน และแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัยมาตลอด

คุณพ่อ คุณแม่ และ คุณชิตชัย มุนินทร์พมาศ ที่สนับสนุนทุนการศึกษาแก่ผู้วิจัย และเป็นกำลังใจด้วยดีตลอดมา

เจ้าหน้าที่ศูนย์คอมพิวเตอร์ และเจ้าหน้าที่ภาควิชาคณิตศาสตร์ทุกท่าน ที่ได้ให้ความช่วยเหลือ และอำนวยความสะดวกในการใช้เครื่องคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาช่วยตรวจ และแก้ไขวิทยานิพนธ์

จีรวุฑ มุนินทร์พมาศ

สารบัญ

	หน้า
บทคัดย่อ.....	(3)
Abstract.....	(4)
กิตติกรรมประกาศ.....	(5)
สารบัญ.....	(6)
รายการตาราง.....	(8)
รายการรูป.....	(9)
บทที่	
1. บทนำ.....	1
1.1 วัตถุประสงค์ของการทำวิจัย.....	2
1.2 ขอบเขตของการทำวิจัย.....	2
1.3 ขั้นตอน และระยะเวลาการดำเนินการวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	5
1.5 เครื่องมือในการทำการวิจัย.....	5
1.6 สถานที่ทำการวิจัย.....	5
2. โปรแกรมเอดิเตอร์แบบเต็มจอภาพ.....	6
2.1 การใช้งานเอดิเตอร์.....	6
2.1.1 เขียนและแก้ไขโปรแกรมภาษาต้นฉบับ.....	6
2.1.2 เป็นส่วนหนึ่งของโปรแกรมระบบสำเร็จรูป.....	6
2.1.3 จัดพิมพ์เอกสารอย่างง่าย.....	7
2.1.4 เป็นส่วนหนึ่งของโปรแกรมประมวลผลคำ.....	7
2.2 ประเภทของเอดิเตอร์.....	7
2.2.1 เอดิเตอร์แบบบรรทัด.....	7
2.2.2 เอดิเตอร์แบบเต็มจอภาพ.....	8
2.3 ความสามารถขั้นพื้นฐานของเอดิเตอร์แบบเต็มจอภาพ.....	12
2.3.1 การเลื่อนตำแหน่งเคอร์เซอร์.....	12
2.3.2 การค้นหาและแทนที่คำ/ข้อความ.....	12
2.3.3 การแทรก/แทนที่ คำ/ข้อความ.....	13

2.3.4	การลบตัวอักษร.....	13
2.3.5	การทำงานเกี่ยวกับกลุ่มของข้อความ.....	13
2.4	องค์ประกอบหลักของเอดิเตอร์.....	22
2.4.1	การจัดการเกี่ยวกับเนื้อหาภายในแฟ้มข้อมูล.....	22
2.4.2	การจัดการเกี่ยวกับแฟ้มข้อมูล.....	23
2.4.3	การจัดการด้านโปรแกรมสำหรับติดต่อกับผู้ใช้.....	24
2.4.4	การจัดการเกี่ยวกับการใช้งานภาษาไทย.....	24
2.5	หลักการพื้นฐานบางประการเกี่ยวกับการพัฒนาเอดิเตอร์	
	แบบเต็มจอภาพ.....	25
2.5.1	ตำแหน่งปัจจุบันของเคอร์เซอร์ และตัวชี้เลขที่ความจำหลัก.....	25
2.5.2	ขนาดของความจำหลักที่จอง.....	30
3.	การออกแบบและการพัฒนาโปรแกรมเอดิเตอร์ PE.....	31
3.1	ลักษณะการติดต่อใช้งาน.....	31
3.1.1	การใช้งานด้วยคำสั่งในรายการเลือก.....	31
3.1.2	การใช้งานด้วยคำสั่งที่แทนด้วยปุ่มแป้นพิมพ์ต่างๆ.....	32
3.2	รูปแบบและขนาดของหน้าจอ.....	32
3.3	สารบบแฟ้มข้อมูล.....	35
3.4	โครงสร้างข้อมูล.....	35
3.5	โครงสร้างระบบ.....	41
3.6	กระบวนการดำเนินงาน.....	53
	คำอธิบายโดยสังเขปสำหรับแต่ละกระบวนการของแผนภาพกระแสข้อมูล	
	ในรูปที่ 3.8 ถึง 3.22.....	69
4.	สรุป.....	74
4.1	ผลการวิจัย.....	74
4.2	ปัญหาและอุปสรรค.....	74
4.3	ข้อเสนอแนะ.....	75
	บรรณานุกรม.....	76
	ภาคผนวก.....	79
	ประวัติผู้เขียน.....	107

รายการตาราง

ตารางที่	หน้า
3.1 ชื่อรายการย่อยในแต่ละรายการหลักของรูปที่ 3.7 และชื่อโปรแกรมย่อยที่ถูกเรียกใช้ตามลำดับตัวอักษรของชื่อ.....	43
3.2 หน้าที่การทำงานของแต่ละโปรแกรมย่อย.....	49
ก.1 รวบรวมคำสั่งทั้งหมดสำหรับการใช้งานเอดิเตอร์ PE.....	104

รายการรูป

รูปที่	หน้า
2.1	ลักษณะของข้อความที่พิมพ์โดยใช้โปรแกรม EDLIN ของระบบปฏิบัติการ DOS..... 8
2.2	ลักษณะของข้อความที่พิมพ์โดยใช้เอดิเตอร์แบบเต็มจอภาพ..... 9
2.3	ตัวอย่างการแก้ไขข้อความในบรรทัดที่ 4 จาก "cls" เป็น "clear screen" และแสดงข้อความทั้งหมดหลังการแก้ไขโดยใช้เอดิเตอร์แบบบรรทัด 10
2.4	ผลลัพธ์ที่ได้หลังการแก้ไขข้อความในบรรทัดที่ 4 จาก "cls" เป็น "clear screen" บนจอภาพในรูปที่ 2.2 ด้วยเอดิเตอร์แบบเต็มจอภาพ..... 11
2.5	ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้ในการเลื่อนตำแหน่งเคอร์เซอร์ของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ..... 14
2.6	ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้ในการค้นหา คำ/ข้อความของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ..... 17
2.7	ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้ในการลบ/แทรก/แทนที่คำ/ข้อความของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ..... 18
2.8	ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้เกี่ยวกับบล็อกของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ..... 20
2.9	ความสัมพันธ์ระหว่างตำแหน่งปัจจุบันของเคอร์เซอร์บนจอภาพและตัวชี้บนความจำหลัก 26
2.10	ตำแหน่งปัจจุบันของเคอร์เซอร์และข้อความบนจอภาพ 28
2.11	ตำแหน่งปัจจุบันของเคอร์เซอร์และข้อความบนจอภาพหลังการกดปุ่มแป้นพิมพ์ PgDn 29
3.1	ส่วนประกอบของหน้าจอขณะใช้งานเอดิเตอร์ PE 34
3.2	สรวบบเพิ่มข้อมูลสำหรับเอดิเตอร์ PE..... 35
3.3	โครงสร้างข้อมูลก่อนและหลังการเพิ่มข้อความ "fat " 37
3.4	โครงสร้างข้อมูลก่อนและหลังการลบข้อความ "fat " 38
3.5	โครงสร้างข้อมูลก่อนและหลังการเปลี่ยนข้อความ "fat boy" เป็น

“thin boy”	39
3.6 โครงสร้างข้อมูลก่อนและหลังการค้นหาคำ “boy”	40
3.7 แผนภูมิรายการเลือกแบบดิ่งลง	42
3.8 กระบวนการดำเนินงานหลักของระบบ	54
3.9 กระบวนการ Configuration	55
3.10 กระบวนการ Main	56
3.11 กระบวนการ Display Status Line	57
3.12 กระบวนการ Menu	58
3.13 กระบวนการ Help	59
3.14 กระบวนการ Delete & Yank	60
3.15 กระบวนการ Screen	61
3.16 กระบวนการ Insert	62
3.17 กระบวนการ Move	63
3.18 กระบวนการ Position	64
3.19 กระบวนการ Find	65
3.20 กระบวนการ Block	66
3.21 กระบวนการ File Process	67
3.22 กระบวนการ Quit	68
ก.1 จอภาพเริ่มต้นสำหรับการทำงานในกรณีที่เกี่ยวข้องใช้งานเอดิเตอร์ PE ด้วยชื่อเพิ่มข้อมูลใหม่ชื่อ NEW_FILE	83
ก.2 จอภาพเริ่มต้นสำหรับการทำงานในกรณีที่เกี่ยวข้องใช้เอดิเตอร์ PE ด้วยชื่อเพิ่มข้อมูลเก่า	84
ก.3 จอภาพที่ได้เมื่อผู้ใช้กดปุ่มฟังก์ชันคีย์ F10 จากหน้าจอรูปที่ ก.1 หรือ ก.2	85
ก.4 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก File	89
ก.5 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Block	92
ก.6 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Search	94
ก.7 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Go to	96
ก.8 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Quit	98
ก.9 จอภาพภายหลังจากใช้คำสั่ง OS Shell	99
ก.10 จอภาพขณะใช้คำสั่ง Help (F1)	101

บทที่ 1

บทนำ

ปัจจุบัน จะพบว่านอกจากการเขียนโปรแกรมประยุกต์ (application program) และการเขียนโปรแกรมประยุกต์สำเร็จรูป (application program package) ยังมีการพัฒนาโปรแกรมระบบสำเร็จรูป (system program package) อำนวยความสะดวกในการใช้งานต่างๆออกมามากมาย อาทิเช่น PC Tools และ Norton Utility ซึ่งต่างก็เป็นโปรแกรมระบบสำเร็จรูปสำหรับจัดการกับหน่วยความจำสำรอง และจัดการกับเพิ่มข้อมูลในหน่วยความจำสำรอง เช่น ใช้ในการจัดเนื้อที่ของหน่วยความจำสำรองให้เป็นระเบียบ และช่วยนำเพิ่มข้อมูลที่ถูกลบไปกลับคืนมา เป็นต้น การที่จะให้ได้มาซึ่งโปรแกรมต่างๆเหล่านี้จะต้องผ่านกระบวนการหลายขั้นตอน เช่น เริ่มตั้งแต่เขียนโปรแกรมต้นฉบับ (source program) จากนั้นนำโปรแกรมต้นฉบับที่ได้ไปผ่านกระบวนการแปล โดยอาศัยตัวแปลภาษา (translator) เพื่อให้ได้โปรแกรมภาษาเครื่อง (object program) หลังจากนั้นจึงนำเอาโปรแกรมภาษาเครื่องที่ได้ไปผ่านกระบวนการประมวลผล เพื่อดำเนินงาน (execute) ให้ตามต้องการ

ถึงแม้ว่ากระบวนการที่จะให้ได้มาซึ่งการดำเนินงานของโปรแกรมหนึ่งๆจะแตกต่างกันบ้างในด้านของการเลือกใช้ชนิดของภาษาในการเขียนโปรแกรมต้นฉบับและการเลือกใช้ตัวแปลภาษาแต่สิ่งหนึ่งที่เหมือนกันคือในการเขียนโปรแกรมต้นฉบับ จะต้องอาศัยโปรแกรมที่เรียกว่า “เทกซ์ เอดิเตอร์” (text editor) หรือที่เรียกกันสั้นๆว่า “เอดิเตอร์” จะพบว่าหากปราศจากโปรแกรมนี้แล้วจะพัฒนาโปรแกรมต่างๆออกมาใช้ไม่ได้ ซึ่งส่งผลให้คอมพิวเตอร์หมดความหมาย เพราะไม่อาจพัฒนาโปรแกรมใดๆได้เลย ไม่ว่าจะเป็นโปรแกรมประยุกต์ โปรแกรมระบบ หรือ โปรแกรมสำเร็จรูปใดๆ

นอกจากนี้ จะพบว่าการทำงานหลายอย่างของเอดิเตอร์จะถูกแฝงอยู่ในการดำเนินงานของโปรแกรมระบบสำเร็จรูปจำนวนมาก เช่น โปรแกรมประมวลผลคำซึ่งรวมขีดความสามารถของโปรแกรมเอดิเตอร์และขีดความสามารถของการจัดรูปแบบการพิมพ์เอกสารอย่างอัตโนมัติเข้าด้วยกัน และโปรแกรมระบบจัดการฐานข้อมูลในส่วนของทำให้ภาษาฐานข้อมูล จะต้องรวมขีดความสามารถของการสร้าง ลบ และเปลี่ยนแปลงข้อความของเอดิเตอร์ไว้ในโปรแกรมด้วย เป็นต้น

จากความสำคัญของโปรแกรมเอดิเตอร์ดังกล่าวมาข้างต้น จากการที่โปรแกรมเอดิเตอร์เป็นโปรแกรมระบบขนาดเล็กที่สามารถถูกพัฒนาสำเร็จได้ด้วยคนเดียวในเวลาไม่นานนัก และ

จากการที่สามารถนำโปรแกรมย่อยหลายๆโปรแกรมในโปรแกรมเอดิเตอร์ไปใช้ในการร่วมพัฒนาโปรแกรมระบบสำเร็จรูปอื่นๆที่อาจถูกพัฒนาเป็นงานวิจัยต่อไปในภาควิชาวิทยาการคอมพิวเตอร์ เช่นโปรแกรมการตรวจสอบคำ (spelling checker) และโปรแกรมระบบจัดการฐานข้อมูล (Database Management System) เป็นต้น ซึ่งจะช่วยประหยัดทั้งเวลา และทรัพยากรที่ต้องใช้ในการพัฒนาเป็นอย่างมาก การพัฒนาโปรแกรมเอดิเตอร์เป็นงานวิจัยของงานวิทยานิพนธ์ในระดับบัณฑิตศึกษาของนักศึกษาสาขาวิชาวิทยาการคอมพิวเตอร์ จึงเป็นเรื่องน่าสนใจทดลองทำ

1.1 วัตถุประสงค์ของการทำวิจัย

1. เพื่อพัฒนาโปรแกรมเอดิเตอร์แบบเต็มจอภาพที่สามารถใช้งานได้ทั้งภาษาอังกฤษ และภาษาไทย และเป็นมิตรกับผู้ใช้
2. สามารถนำโปรแกรมย่อยหลายๆโปรแกรมในโปรแกรมเอดิเตอร์ไปใช้ในการร่วมพัฒนาโปรแกรมระบบสำเร็จรูปอื่นๆที่อาจถูกพัฒนาเป็นงานวิจัยต่อไปในภาควิชาวิทยาการคอมพิวเตอร์ เช่นโปรแกรมการตรวจสอบคำ และโปรแกรมระบบจัดการฐานข้อมูล

1.2 ขอบเขตของการทำวิจัย

ในการทำการวิจัยครั้งนี้เป็นการสร้างเอดิเตอร์แบบเต็มจอภาพ ซึ่งมีขีดความสามารถดังนี้

1. การจัดการเกี่ยวกับเพิ่มข้อมูล
2. การจัดการเกี่ยวกับเนื้อหาภายในเพิ่มข้อมูล
3. การจัดการเกี่ยวกับเคอร์เซอร์
4. การใช้งานกับภาษาไทย
5. การติดต่อใช้งานในรูปรายการเลือกแบบดึงลง (pulldown menu)
6. การพิมพ์ข้อมูลทั้งบนจอภาพและเครื่องพิมพ์

1.3 ขั้นตอน และระยะเวลาการดำเนินการวิจัย

1.3.1 ขั้นตอนการดำเนินงาน

1. ศึกษาภาษาซี
2. ศึกษาชนิด และคุณสมบัติของโปรแกรมเอดิเตอร์
3. ศึกษาและทำความเข้าใจเกี่ยวกับลักษณะและการทำงานของเอดิเตอร์แบบเต็มจอภาพที่มีใช้อยู่ทั่วไปในปัจจุบัน
4. วิเคราะห์และออกแบบเอดิเตอร์ที่จะพัฒนา
5. ออกแบบโครงสร้างการดำเนินงานของโปรแกรม และการแสดงผลบนจอภาพ

6. พัฒนา ทดสอบและแก้ไขโปรแกรม เพื่อให้ทำงานได้ถูกต้องสมบูรณ์ที่สุด
7. จัดทำเอกสารและจัดพิมพ์รายงาน

1.3.2 ระยะเวลาในการดำเนินงาน

ใช้เวลาทั้งสิ้นประมาณ 2 ปี 6 เดือน

เดือนที่ ขั้น ตอน การทำ งานวิจัย	พ.ย.-ธ.ค. 32	ม.ค.-เม.ย. 33	พ.ค.-มิ.ย. 33	ก.ค.-ก.ย. 33	ต.ค.-พ.ย. 33	ธ.ค.-ก.พ. 33 - 35	มี.ค.-เม.ย. 35
1	←→						
2		←→					
3			←→				
4				←→			
5					←→		
6						←→	
7							←→

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โปรแกรมเอดิเตอร์แบบเต็มจอภาพ ซึ่งสามารถนำไปพัฒนาต่อให้มีประสิทธิภาพมากขึ้นได้อีก
2. ได้โปรแกรมเอดิเตอร์ที่สามารถใช้งานได้ทั้งภาษาอังกฤษ และภาษาไทย ซึ่งจะอำนวยความสะดวกในการนำไปใช้เขียนโปรแกรมต้นฉบับที่ใช้ภาษาไทยได้ และสร้างแฟ้มข้อมูลต่างๆที่ประกอบด้วยข้อความทั้งภาษาไทยและภาษาอังกฤษ

1.5 เครื่องมือใช้ในการทำการวิจัย

1. เครื่องไมโครคอมพิวเตอร์ IBM PC หรือ IBM PC Compatible ที่มีฮาร์ดดิสก์
2. Turbo C เวอร์ชัน 2.0
3. ระบบปฏิบัติการ DOS เวอร์ชัน 3.3 ขึ้นไป
4. Thai driver ซึ่งมีรหัสภาษาไทย สมอ. เช่น VTHAI.COM ไทยทิพย์ และภาษาไทยของบริษัท International Research Corporation LTD. (IRC) เป็นต้น

1.6 สถานที่ทำการวิจัย

1. ศูนย์คอมพิวเตอร์มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่
2. ห้องปฏิบัติการไมโครคอมพิวเตอร์ ภาควิชาคณิตศาสตร์มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

บทที่ 2

โปรแกรมเอดิเตอร์แบบเต็มจอภาพ

เทกซ์เอดิเตอร์ หรือ เอดิเตอร์ เป็นโปรแกรมที่เกิดมาพร้อมกับการใช้งานคอมพิวเตอร์ผ่านทางเทอร์มินัล (terminal) ของระบบเครื่องคอมพิวเตอร์แบบเมนเฟรม (mainframe) และแบบมินิคอมพิวเตอร์ (mini-computer) ในการจัดทำโปรแกรมต้นฉบับ (source program) ด้วยภาษาแอสเซมบลี (assembly language) หรือภาษาคอมพิวเตอร์ระดับสูง (high level language) ทั่วไป นอกจากนี้ในการพิมพ์เอกสารใดๆด้วยคอมพิวเตอร์ เอดิเตอร์ก็จะช่วยให้ผู้พิมพ์ไม่ต้องแก้ไขเอกสารที่พิมพ์ด้วยการพิมพ์เอกสารนั้นใหม่ทั้งหมดเหมือนการใช้เครื่องพิมพ์ดีด ผู้ใช้สามารถแทรกข้อความใหม่หรือบรรทัดใหม่ สามารถลบข้อความ และขยับข้อความตามจุดต่างๆของเอกสารตามที่ต้องการได้อย่างสะดวก แต่อย่างไรก็ดีการพิมพ์ และแก้ไขเอกสารด้วยเอดิเตอร์ ผู้ใช้จะต้องจัดการรูปแบบความสวยงามของหน้าเอกสารที่พิมพ์ด้วยตนเอง ในเวลาต่อมาจึงเกิดโปรแกรมประมวลผลคำ (word processor) ซึ่งรวมขีดความสามารถของเอดิเตอร์ และขีดความสามารถในการจัดรูปแบบเอกสาร ข้อความ รูปแบบและขนาดของตัวอักษร ให้อยู่ในรูปแบบที่สวยงามเป็นระเบียบได้อย่างอัตโนมัติทันทีทันใดตามที่ต้องการ

2.1 การใช้งานเอดิเตอร์

ในปัจจุบัน การนำเอดิเตอร์ไปใช้งานมีอยู่ 4 ลักษณะใหญ่ๆดังนี้

2.1.1 เขียนและแก้ไขโปรแกรมภาษาต้นฉบับ

การใช้งานเอดิเตอร์ในลักษณะนี้ จะเป็นการนำเอดิเตอร์มาใช้เขียนโปรแกรมภาษาระดับสูงเช่น ภาษาซี ภาษาปาสคาล และโปรแกรมภาษาระดับต่ำ เช่น ภาษาแอสเซมบลี จากนั้นก็จะนำเอาโปรแกรมต้นฉบับที่ได้ไปทำการแปลโดยอาศัยตัวแปลภาษา ซึ่งถ้าหากโปรแกรมดังกล่าวมีความผิดพลาด ก็จะถูกนำกลับมาทำการแก้ไขด้วยเอดิเตอร์อีก จากนั้นก็จะนำกลับไปทำการแปลอีกครั้ง และถ้าหากโปรแกรมยังมีข้อผิดพลาดอีก ก็จะนำกลับมาทำการแก้ไขใหม่ เช่นนี้เรื่อยไปจนกว่าโปรแกรมจะทำงานได้ถูกต้อง

2.1.2 เป็นส่วนหนึ่งของโปรแกรมระบบสำเร็จรูป

เมื่อพิจารณาถึงโปรแกรมระบบสำเร็จรูปไม่ว่าจะเป็นทางด้านระบบจัดการฐานข้อมูล หรือ spread sheet จะพบว่าในการทำงานของโปรแกรดังกล่าวจะมีส่วนที่ให้ผู้จัดการกับข้อความอยู่ด้วยเสมอ เช่น เมื่อมีการใช้งานโปรแกรมระบบจัดการฐานข้อมูลโอราเคิล (ORACLE)

เป็นโปรแกรมระบบสำเร็จรูปทางด้านการจัดการฐานข้อมูล เพื่อทำการค้นหาข้อมูลที่เก็บอยู่ในฐานข้อมูลนั้น ผู้ใช้จำเป็นจะต้องสร้างคำถาม (query) ในรูปข้อความสอบถามขึ้นมาเสียก่อน จากนั้นจึงนำคำถามที่ได้ไปให้เครื่องทำการประมวลผลเพื่อให้ได้คำตอบตามต้องการ จากกระบวนการดำเนินงานเพื่อค้นหาข้อมูลดังกล่าว ถ้าหากพิจารณาให้ดี จะพบว่าในขั้นตอนของการสร้างคำถาม จะมีเอดิเตอร์เข้ามาเกี่ยวข้องอยู่ด้วยโดยที่ผู้ใช้เองอาจจะคิดไม่ถึง เนื่องจากผู้ใช้ส่วนใหญ่จะมุ่งความสนใจไปยังวากยสัมพันธ์ของภาษาฐานข้อมูล มากกว่าที่จะคิดไปว่ากำลังใช้งานเอดิเตอร์อยู่

2.1.3 จัดพิมพ์เอกสารอย่างง่าย

ในการจัดพิมพ์เอกสารที่ไม่จำเป็นจะต้องมีการจัดรูปแบบของการพิมพ์ให้สวยงามมากนัก เช่น ในการพิมพ์ข้อความสั้นๆ หรือการทำ mail merge ผู้ใช้ก็สามารถจัดพิมพ์โดยใช้เอดิเตอร์ได้และจัดรูปแบบของการพิมพ์ด้วยมือตามที่ต้องการ

2.1.4 เป็นส่วนหนึ่งของโปรแกรมประมวลผลคำ

ในการจัดพิมพ์เอกสารจำนวนมาก ซึ่งต้องการการจัดรูปแบบของเอกสารได้อย่างอัตโนมัติ โปรแกรมประมวลผลคำ จึงถูกพัฒนาเพื่ออำนวยความสะดวกในการจัดพิมพ์เอกสารโดยเฉพาะ โดยโปรแกรมประมวลผลคำทุกโปรแกรมอย่างน้อยจะต้องมีขีดความสามารถของเอดิเตอร์อยู่ด้วยเสมอ

2.2 ประเภทของเอดิเตอร์

จากการศึกษาโปรแกรมเอดิเตอร์ที่มีใช้บนเครื่องไมโครคอมพิวเตอร์ พบว่ามีเอดิเตอร์ที่นิยมใช้อยู่หลายตัว เช่น บนระบบปฏิบัติการ DOS มี EDLIN, EDIT และ SK เป็นต้น และบนระบบปฏิบัติการ UNIX มี vi, ed, ex และ pico เป็นต้น

โปรแกรมเอดิเตอร์ แบ่งโดยทั่วไปออกได้เป็น 2 ประเภท คือ เอดิเตอร์แบบบรรทัด (line editor) และ เอดิเตอร์แบบเต็มจอภาพ (full screen editor)

2.2.1 เอดิเตอร์แบบบรรทัด

เอดิเตอร์แบบบรรทัด เป็นเอดิเตอร์ซึ่งมีลักษณะการทำงานกับข้อความแบบบรรทัดที่บ่งบอก เพิ่มข้อมูลซึ่งใช้กับเอดิเตอร์ชนิดนี้จะประกอบด้วยข้อความซึ่งมีเลขที่บรรทัดของข้อความอยู่ด้วย คำสั่งในการทำงานกับข้อความจะมีการบ่งบอกเลขที่บรรทัดที่ต้องการด้วยเสมอ เอดิเตอร์แบบนี้ไม่ค่อยมีผู้นิยมใช้เนื่องจากใช้ยาก เพราะใช้คำสั่งควบคุมการทำงานได้ครั้งละบรรทัดเท่านั้น นอกจากนี้ ภายหลังจากการแก้ไขข้อความในแต่ละบรรทัด ผู้ใช้จะไม่สามารถเห็นได้ในทันทีว่าข้อความเดิมมีการแก้ไขไปเป็นเช่นไร นอกจากจะใช้คำสั่งแสดงข้อความนั้นๆออกมาดู จากเหตุผลดังกล่าว ผู้ใช้ส่วนใหญ่จะใช้เอดิเตอร์นี้เมื่อไม่มีเอดิเตอร์แบบเต็มจอภาพให้ใช้ รูปที่ 2.1 เป็นตัวอย่าง

ของข้อความของแฟ้มข้อมูลที่ถูกสร้างด้วยเอดิเตอร์แบบบรรทัดชื่อ EDLIN บนระบบปฏิบัติการ DOS

```

1:*set comspec=c:\command.com
2: prompt $p$g
3: rtkill
4: cls
5: dir/p
6: SET NU=C:\NU6
7: SET INCLUDE=D:\CLIPPER5\INCLUDE;%INCLUDE%

```

เลขที่บรรทัด

รูปที่ 2.1 ลักษณะของข้อความที่พิมพ์โดยใช้โปรแกรม EDLIN ของระบบปฏิบัติการ DOS

2.2.2 เอดิเตอร์แบบเต็มจอภาพ

เอดิเตอร์แบบเต็มจอภาพ เป็นเอดิเตอร์ซึ่งมีลักษณะการทำงานกับข้อความบนจอภาพได้ทุกบรรทัด โดยการใช้สัญลักษณ์พิเศษที่เรียกว่า เคอร์เซอร์ (cursor) ซึ่งสามารถถูกเลื่อนไปมาทางซ้าย และขวา ขึ้นและลงไปยังทุกตำแหน่งของข้อความบนจอภาพ เพื่อพิมพ์ข้อความ, ลบข้อความ และ/หรือ เปลี่ยนแปลงแก้ไขข้อความ ณ จุดที่เคอร์เซอร์อยู่ได้ตามต้องการ ดังนั้น ผู้ใช้จะสามารถจัดการกับข้อความ ณ ตำแหน่งใดๆบนหน้าจอได้อย่างสะดวก รูปที่ 2.2 แสดงตัวอย่างข้อความของแฟ้มข้อมูลที่ถูกสร้างด้วยเอดิเตอร์แบบเต็มจอภาพ โดยเครื่องหมาย “_” เป็นสัญลักษณ์แทนเคอร์เซอร์

```

Line 1      Col 27      Ins      C:\SAMPLE.TXT      Cols 27      Lines 7
set comspec=c:\command.com_
prompt $p$g      ↑      เคอร์เซอร์
rtkill
cls
dir/p
SET NU=C:\NU6
SET INCLUDE=D:\CLIPPER5\INCLUDE;%INCLUDE%

F1-Help  F2-Save  F3-Load  F4-Find  F5-Replace  F9-Thai/Eng  F10-Menu  Alt-X-Quit

```

รูปที่ 2.2 ลักษณะของข้อความที่พิมพ์โดยใช้เอดิเตอร์แบบเต็มจอภาพ

ความแตกต่างระหว่าง เอดิเตอร์แบบเต็มจอภาพ และ เอดิเตอร์แบบบรรทัด

ลักษณะความแตกต่างระหว่างเอดิเตอร์แบบเต็มจอภาพ และเอดิเตอร์แบบบรรทัดมีดังนี้

1. ลักษณะของเอดิเตอร์แบบเต็มจอภาพ จะเป็นแบบ WYSIWYG (อ่านว่า wiz-ee-wig) ซึ่งย่อมาจาก "What You See Is What You Get" ซึ่งหมายความว่า สิ่งที่ปรากฏอยู่บนหน้าจอภาพจะเหมือนกับสิ่งที่ถูกเก็บลงในแฟ้มข้อมูลหรือจะพิมพ์ออกทางเครื่องพิมพ์ ณ ขณะเวลานั้น แต่สำหรับเอดิเตอร์แบบบรรทัดนั้น สิ่งที่ปรากฏบนจอภาพ และสิ่งที่ถูกเก็บในแฟ้มข้อมูล หรือจะพิมพ์ออกทางเครื่องพิมพ์ อาจจะมีบางอย่างที่แตกต่างกัน รูปที่ 2.3 แสดงขั้นตอนการแก้ไขข้อความในบรรทัดที่ 4 จากข้อความ "cls" เป็นข้อความใหม่ "clear screen" พร้อมทั้งการใช้คำสั่งให้แสดงข้อความทั้งหมดในแฟ้มข้อมูลหลังการแก้ไขข้อความแล้วเมื่อใช้เอดิเตอร์แบบบรรทัด ในขณะที่รูปที่

2.4 แสดงหน้าจอที่เห็นหลังจากการแก้ไขข้อความ "cls" เป็น "clear screen" บนจอภาพในรูปแบบที่ 2.2 ด้วยเอดิเตอร์แบบเต็มจอภาพ

```

1:*set comspec=c:\command.com
2: prompt $p$g
3: rtkill
4: cls
5: dir/p
6: SET NU=C:
7: SET INCLUDE=D:;%INCLUDE%
*4
4:*cls
4:*clear screen
*|
1:*set comspec=c:\command.com
2: prompt $p$g
3: rtkill
4: clear screen
5: dir/p
6: SET NU=C:
7: SET INCLUDE=D:;%INCLUDE%
*
↑
เคอร์เซอร์

```

รูปที่ 2.3 ตัวอย่างการแก้ไขข้อความในบรรทัดที่ 4 จาก "cls" เป็น "clear screen" และแสดงข้อความทั้งหมดหลังการแก้ไขโดยใช้เอดิเตอร์แบบบรรทัด

ขั้นตอนของการแก้ไขข้อความจาก "cls" เป็น "clear screen" ในรูปที่ 2.3 โดยใช้เอดิเตอร์แบบบรรทัด จะเริ่มต้นจากการที่เครื่องคอมพิวเตอร์อ่านข้อความจากแฟ้มข้อมูลที่ต้องการแก้ไขไปเก็บไว้ในหน่วยความจำหลัก ณ ตำแหน่งที่เตรียมไว้ และเมื่อผู้ใช้งานต้องการดูข้อความที่เก็บอยู่ในแฟ้มข้อมูลนั้นก็สามารถทำได้โดยใช้คำสั่ง list (ด้วยการกดปุ่มอักขระ "l") ซึ่งข้อมูลที่เก็บอยู่

ในความจำหลักก็จะปรากฏบนจอภาพ และเมื่อผู้ใช้ต้องการแก้ไขข้อความจาก "cls" เป็น "clear screen" ในบรรทัดที่ 4 ก็ให้พิมพ์ตัวเลข 4 และกดปุ่มแป้นพิมพ์ "Enter" บนแป้นพิมพ์ เครื่องคอมพิวเตอร์ก็จะแสดงข้อความบนจอภาพดังนี้

```
4:*cls
```

```
4:*_
```

โดยข้อความ "4:*cls" จะเป็นการบอกให้ผู้ใช้ทราบว่าข้อความเก่าที่เก็บอยู่ในบรรทัดที่ 4 คือ "cls" และในบรรทัดต่อมาก็จะเป็นการบอกให้ผู้ใช้ทราบว่าข้อความที่ผู้ใช้จะพิมพ์ ณ จุดที่เคอร์เซอร์ "_" อยู่ (ในที่นี้คือ "clear screen") จะไปแทนที่ข้อความเดิม ("cls") ณ บรรทัดที่ 4 แต่ผู้ใช้ก็ยังไม่เห็นการเปลี่ยนแปลงที่เกิดขึ้นจนกว่าจะได้ใช้คำสั่ง list อีกครั้งหนึ่ง

```

Line 4      Col 13  Ins   C:\SAMPLE.TXT           Cols 27  Lines 7
set comspec=c:\command.com
prompt $p$g
rtkill
clear screen_
dir/p      ← เคอร์เซอร์
SET NU=C:
SET INCLUDE=D:;%INCLUDE%

```

```

F1-Help F2-Save F3-Load F4-Find F5-Replace F9-Thia/Eng F10-Menu Alt-X-Quit

```

รูปที่ 2.4 ผลลัพธ์ที่ได้หลังการแก้ไขข้อความในบรรทัดที่ 4 จาก "cls" เป็น "clear screen" บนจอภาพในรูปที่ 2.2 ด้วยเอดิเตอร์แบบเต็มจอภาพ

ขั้นตอนของการแก้ไขข้อความจาก “cls” เป็น “clear screen” ในรูปที่ 2.4 ด้วยการใช้อีดิเตอร์แบบเต็มจอภาพ จะเริ่มต้นจากการที่เครื่องคอมพิวเตอร์อ่านข้อความจากแฟ้มข้อมูลที่ต้องการแก้ไขไปเก็บไว้ในหน่วยความจำหลัก ณ ตำแหน่งที่เตรียมไว้พร้อมกับแสดงข้อความตั้งแต่เริ่มต้นบนจอภาพ และเมื่อผู้ใช้ต้องการแก้ไขข้อความจาก “cls” เป็น “clear screen” ก็เพียงแต่เลื่อนเคอร์เซอร์ไปยังตัวอักษร “c” ของข้อความ “cls” และกดปุ่มแป้นพิมพ์สำหรับลบข้อความดังกล่าว จากนั้นจึงพิมพ์ข้อความ “clear screen” แทน ณ ตำแหน่งเดิมนั้น ซึ่งก็จะไปปรากฏแทนที่ตำแหน่งเดิมของข้อความ “cls”

จากรูปขั้นตอนการแก้ไขข้อความดังกล่าวมาข้างต้น จะเห็นว่าการใช้อีดิเตอร์แบบเต็มจอภาพนั้น ผู้ใช้จะเห็นถึงความเปลี่ยนแปลงได้ทันที ในขณะที่การใช้อีดิเตอร์แบบบรรทัดจะไม่เห็นต้องใช้คำสั่งในการดูส่วนที่ถูกเปลี่ยนแปลงนั้นก่อน ดังนั้นอีดิเตอร์แบบบรรทัดจึงขาดคุณสมบัติของ WYSIWYG ซึ่งอีดิเตอร์แบบเต็มจอภาพมี

2. ในกรณีของอีดิเตอร์แบบเต็มจอภาพ ผู้ใช้สามารถเลื่อนเคอร์เซอร์ โดยใช้แป้นสัญลักษณ์ลูกศรบนปุ่มแป้นพิมพ์ (↑↓→←) ไปยังตำแหน่งใดๆบนหน้าจอตามต้องการ ซึ่งจะไม่สามารถทำได้เมื่อใช้อีดิเตอร์แบบบรรทัด

2.3 ความสามารถขั้นพื้นฐานของอีดิเตอร์แบบเต็มจอภาพ

ในการทำงานของอีดิเตอร์แบบเต็มจอภาพโดยทั่วไปจะอาศัยสัญลักษณ์พิเศษบนจอภาพที่เรียกว่า “เคอร์เซอร์” ซึ่งมักใช้เครื่องหมาย “_” (underscore) เป็นตัวบอกตำแหน่งปัจจุบันของเคอร์เซอร์บนจอภาพ สำหรับจัดการกับอักขระ/ข้อความ ที่ต้องการ ณ ตำแหน่งปัจจุบันของเคอร์เซอร์ และความสามารถหลักๆของอีดิเตอร์แบบเต็มจอภาพ ควรมีดังต่อไปนี้

2.3.1 การเลื่อนตำแหน่งเคอร์เซอร์

การเลื่อนตำแหน่งเคอร์เซอร์ จะเป็นการเคลื่อนย้ายเคอร์เซอร์ไปยังตำแหน่งใดๆบนจอภาพตามที่ต้องการโดยใช้ปุ่มแป้นพิมพ์ รูปที่ 2.5 แสดงตัวอย่างการใช้ปุ่มแป้นพิมพ์ในการเลื่อนเคอร์เซอร์ที่นิยมใช้กันของโปรแกรมอีดิเตอร์ และโปรแกรมประมวลผลคำต่างๆที่ใช้กันอยู่ในปัจจุบัน

2.3.2 การค้นหาและแทนที่คำ/ข้อความ

การค้นหาจะให้สำหรับค้นหาคำ/ข้อความที่มีอยู่แล้วโดยที่เคอร์เซอร์จะแสดงตำแหน่งอักขระแรกของคำ/ข้อความที่พบให้ผู้ใช้ทราบ ส่วนการแทนที่จะใช้สำหรับแทนที่คำ/ข้อความเดิมด้วยคำ/ข้อความใหม่ ซึ่งคำสั่งในการค้นหาและแทนที่ของโปรแกรมอีดิเตอร์ และโปรแกรม

ประมวลผลคำส่วนใหญ่ จะอาศัยปุ่มแป้นพิมพ์ในการค้นหาคำ/ข้อความ ดังตัวอย่างแสดงใน รูปที่ 2.6

2.3.3 การแทรก/แทนที่ คำ/ข้อความ

ในการแทรกคำ/ข้อความ จะเป็นการเลื่อนข้อความทั้งบรรทัดจากตำแหน่งปัจจุบันของเคอร์เซอร์ไปทางขวาเป็นจำนวนเท่ากับจำนวนตัวอักขระที่ประกอบขึ้นเป็นคำ/ข้อความที่จะแทรก จากนั้นจึงบรรจุคำ/ข้อความใหม่ลงยังตำแหน่งปัจจุบันของเคอร์เซอร์

ในการแทนที่คำ/ข้อความ จะเป็นการนำเอาคำ/ข้อความใหม่ไปแทนที่คำ/ข้อความซึ่งอยู่ในตำแหน่งปัจจุบันของเคอร์เซอร์ โดยไม่มีการเลื่อนข้อความเดิมทั้งบรรทัดจากตำแหน่งปัจจุบันของเคอร์เซอร์ไปทางขวา

โปรแกรมเอดิเตอร์ และโปรแกรมประมวลผลคำส่วนใหญ่ จะอาศัยปุ่มแป้นพิมพ์ กำหนดสถานะการแทรก/แทนที่ คำ/ข้อความ ดังตัวอย่างแสดงในรูปที่ 2.7

2.3.4 การลบตัวอักขระ

การลบตัวอักขระ จะเป็นการนำเอาตัวอักขระที่ไม่ต้องการออกจากความจำหลักและลบออกจากจอภาพ ซึ่งการลบตัวอักขระจะมีอยู่ 2 วิธีใหญ่ๆดังนี้

การลบตัวอักขระที่อยู่ ณ ตำแหน่งปัจจุบันของเคอร์เซอร์ จะเป็นการเลื่อนตัวอักขระแต่ละตัวที่อยู่ทางขวาของเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง ซึ่งเป็นผลให้ตัวอักขระเดิมที่อยู่ ณ ตำแหน่งปัจจุบันของเคอร์เซอร์ ถูกแทนที่ด้วยอักขระที่อยู่ถัดไปทางขวา

การลบตัวอักขระที่อยู่ทางซ้ายของตำแหน่งปัจจุบันของเคอร์เซอร์ จะเป็นการเลื่อนตัวอักขระแต่ละตัวจากตำแหน่งปัจจุบันของเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง เป็นผลให้ตัวอักขระเดิมที่อยู่ทางซ้ายของเคอร์เซอร์ถูกแทนที่ด้วยอักขระที่อยู่ถัดไปทางขวา (ที่เคอร์เซอร์อยู่) พร้อมกับเลื่อนตำแหน่งเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง รูปที่ 2.7 แสดงตัวอย่างการใช้ปุ่มแป้นพิมพ์ในการลบอักขระที่นิยมใช้กันของโปรแกรมเอดิเตอร์ และโปรแกรมประมวลผลคำต่างๆที่ใช้กันอยู่ในปัจจุบัน

2.3.5 การทำงานเกี่ยวกับกลุ่มของข้อความ

การทำงานเกี่ยวกับกลุ่มของข้อความซึ่งโดยทั่วไปมักเรียกเป็นหน่วยว่าบล็อก (block) จะเป็นการจัดการกับบล็อกในลักษณะต่างๆเช่น การคัดลอกข้อความหนึ่งบล็อก (copy) การเคลื่อนย้ายข้อความหนึ่งบล็อกจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง (cut and paste) การลบข้อความหนึ่งบล็อกทิ้ง และการพิมพ์ข้อความเพิ่มหนึ่งบล็อก เป็นต้น ซึ่งคำสั่งการทำงานเกี่ยวกับบล็อกของเอดิเตอร์ และโปรแกรมประมวลผลคำต่างๆโดยมากใช้ปุ่มแป้นพิมพ์ดังตัวอย่างแสดงใน รูปที่ 2.8

เลื่อนตำแหน่งเคอร์เซอร์							
ชื่อโปรแกรม	ไปทางซ้าย 1 ตำแหน่ง	ไปทางขวา 1 ตำแหน่ง	เลื่อนขึ้นบน 1 บรรทัด	เลื่อนลงล่าง 1 บรรทัด	ไปคำถัดไป	ไปคำที่แล้ว	ไปต้นเพิ่มข้อมูล
TC	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR, ^PgUp
SK	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR, ^PgUp
RW 2.1	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR
CW 1.5	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR, ^PgUp
LED	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR
ED	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR, ^Home
Kedit	←	→	↑	↓	^→	^←	^PgUp
DOS 5.0	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR, ^PgUp
PE	^S, ←	^D, →	^E, ↑	^X, ↓	^F, ^→	^A, ^←	^QR, ^PgUp

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.5 ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้ในการเลื่อนตำแหน่งเคอร์เซอร์ของเอดิเตอร์ และโปรแกรมประมวลผลคำต่างๆ

เลื่อนตำแหน่งเคอร์เซอร์						
ชื่อโปรแกรม	ไปท้ายเพิ่มข้อมูล	ไปต้นบรรทัด	ไปท้ายบรรทัด	ไปบรรทัดแรกของหน้าจอ	ไปบรรทัดสุดท้ายของหน้าจอ	ไปบรรทัดที่ผู้ใช้กำหนด
TC	^QC ,^PgDn	Home, ^QS	End, ^QD	^Home	^End, ^QX	
SK	^QC ,^PgDn	Home, ^QS	End, ^QD	^Home	^End, ^QX	
RW 2.1	^QC	^QS	^QD	^QE	^QX	
CW 1.5	^QC, ^PgDn	Home	End	^QE, ^Home	^End, ^QX	Alt-L
LED	^QC	^QS	^QD	^W	^Z	^QI
ED	^QC ,End	Home	End	^PgUp	^PgDn	^J
Kedit	^PgDn	F7	End	^Home		
DOS 5.0	^QC, ^PgDn	Home	End	^QE	^QX	
PE	^QC, ^PgDn	Home	End	Alt-T	Alt-B	Alt-G

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.5 (ต่อ)

เลื่อนตำแหน่งเคอร์เซอร์					
ชื่อโปรแกรม	เลื่อนจอขึ้น 1 บรรทัด	เลื่อนจอลง 1 บรรทัด	เลื่อนขึ้น 1 หน้าจอ	เลื่อนลง 1 หน้าจอ	ไปยังย่อหน้าถัดไป
TC	^W	^Z	^R, PgUp	^C, PgDn	Tab
SK	^W	^Z	^R, PgUp	^C, PgDn	Tab
RW 2.1	^W	^Z	^R, PgUp	^C, PgDn	Tab
CW 1.5	^W	^Z	^R, PgUp	^C, PgDn	Tab
LED			^R, PgUp	^C, PgDn	Tab
ED	^Z	^W	^R, PgUp	^C, PgDn	Tab
Kedit			^R, PgUp	^Home	Tab, F4
DOS 5.0	^W	^Z	^R, PgUp	^C, PgDn	Tab
PE	^W	^Z	^R, PgUp	^C, PgDn	Tab

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.5 (ต่อ)

ค้นหา/แทนที่			
ชื่อโปรแกรม	ค้นหาคำ	แทนที่	ค้นหาต่อไป
TC	^QF	^QA	^L
SK	^QF	^QA	^L
RW 2.1	^QF	^QA	^L
CW 1.5	^QF	^QA	^L
LED	^QF	^QA	^L
ED	^QF, F2	^QA	^L
Kedit			
DOS 5.0	^QF	^QA	^L, F3
PE	^QF	^QA	^L

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.6 ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้ในการค้นหาคำ/ข้อความ
ของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ

ลบ/แทรก						
ชื่อโปรแกรม	ลบตัวอักษร	ลบตัวอักษรทางซ้าย	ลบคำ	ลบบรรทัด	ลบไปจนหมดบรรทัด	แทรกบรรทัด
TC	^G, Del	^H, BS	^T	^Y	^QY	^N
SK	^G, Del	^H, BS	^T	^Y	^QY	^N, ^M
RW 2.1	^G, Del	^H, BS	^T	^Y	^QY	^M
CW 1.5	^G, Del	^H, BS	^T	^Y	^QY	^N
LED	^G, Del	^H, BS	^T	^Y	^QY	^M
ED	^G, Del	^H, BS	^T	^Y, Alt-D	^QY	^N, Alt-S
Kedit	Del	BS	Alt-W	Alt-D	^End	^P, ^F2
DOS 5.0	^G, Del	^H, BS	^T	^Y	^QY	^N
PE	^G, Del	^H, BS	^T	^Y	^QY	^N

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.7 ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้ในการลบ/แทรก/แทนที่ คำ/ข้อความของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ

ลบ/แทรก				
ชื่อโปรแกรม	พิมพ์แทรก/ ทับ	ลบจากต้น บรรทัดถึง เคอร์เซอร์	ลบจากตำแหน่ง ปัจจุบัน จนจบเพิ่ม	เอาข้อความ ที่ถูกลบไป กลับคืน
TC	^V, Ins			^QL
SK	^V, Ins			^QL
RW 2.1	^V	^Q+BS		^U
CW 1.5	^V, Ins			^QL
LED	^V, Ins			^U
ED	^V, Ins			^U
Kedit				Alt-R
DOS 5.0	^V, Ins			
PE	^V, Ins		Alt-A+Y	Alt-Y

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.7 (ต่อ)

บล็อก						
ชื่อโปรแกรม	ตำแหน่งเริ่มต้น	ตำแหน่งสุดท้าย	คัดลอก	ลบ	เคลื่อนย้าย	อ่าน
TC	^KB	^KK	^KC	^KY	^KV	^KR
SK	^KB, F7	^KK, F8	^KC	^KY	^KV	^KR
RW 2.1	^KB	^KK	^KC	^KY	^KV	^KR
CW 1.5	^KB	^KK	^KC	^KY	^KV	^KR
LED	^KB	^KK	^KC	^KY	^KV	^KR
ED	^KB, F1	^KK, F1	^KC	^KY	^KV	^KR
Kedit	Alt-Z	Alt-Z	Alt-K	Alt-G	Alt-M	
DOS 5.0	Shift+Arrow	Shift+Arrow	Alt-E+C	Alt-E+E	Alt-E+T+P	
PE	^KB	^KK	^KC	^KY	^KV	^KR

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.8 ตัวอย่างคำสั่งในรูปปุ่มแป้นพิมพ์ที่ใช้เกี่ยวกับบล็อกของเอดิเตอร์และโปรแกรมประมวลผลคำต่างๆ

บล็อก				
ชื่อโปรแกรม	เขียนบล็อกไป ยังเพิ่มข้อมูล	ซ่อน/แสดง	พิมพ์	ชนิดคอลัมน์
TC	^KW	^KH	^KP	
SK	^KW	^KH	^KP	
RW 2.1	^KW	^KH	^KP	^KN
CW 1.5	^KW	^KH	^KP	^KO
LED	^KW	^KH	^KW	^KN
ED	^KW	^KH	^KC	
Kedit	^KW	^KH	Alt-K	Alt-B
DOS 5.0	^KW	Shift+Arrow	Alt-K+P	
PE	^KW	^KH	^KP	^KO

หมายเหตุ “^” แทนปุ่ม “Ctrl” บนแป้นพิมพ์

รูปที่ 2.8 (ต่อ)

2.4 องค์ประกอบหลักของเอดิเตอร์

โดยทั่วไป โปรแกรมเอดิเตอร์ต้องมีความสามารถในการจัดการเกี่ยวกับเนื้อหาของข้อความภายในแฟ้มข้อมูลเป็นหลัก เช่น การสร้าง/เพิ่มข้อความใหม่ การลบข้อความเก่า และการปรับเปลี่ยนข้อความเก่าเป็นข้อความใหม่ตามที่ต้องการ แต่เอดิเตอร์ที่ดีควรอำนวยความสะดวกให้กับผู้ใช้สามารถใช้งานได้อย่างง่าย ๆ สบาย ๆ ทั้งการจัดการเกี่ยวกับข้อความภายในแฟ้มข้อมูล และการใช้คำสั่งในการสร้างแฟ้มข้อมูลใหม่ ลบแฟ้มข้อมูลเก่า และปรับเปลี่ยนแฟ้มข้อมูลเก่าสำหรับข้อความใหม่ด้วย ซึ่งปกติคำสั่งเหล่านี้จะเป็นส่วนหนึ่งของคำสั่งของระบบปฏิบัติการของเครื่องคอมพิวเตอร์นั่นเอง ดังนั้นเอดิเตอร์ใหม่ซึ่งถูกพัฒนาสำหรับใช้บนเครื่องไมโครคอมพิวเตอร์ในปัจจุบัน จึงรวมเอาขีดความสามารถส่วนนี้เข้าไปด้วย โดยเอดิเตอร์เหล่านี้จะถูกออกแบบให้ส่วนของโปรแกรมสำหรับติดต่อกับผู้ใช้ (user interface) อยู่ในรูปรายการเลือกแบบดิ่งลง เพื่อให้ผู้ใช้สามารถเรียกใช้งานต่างๆ ได้อย่างง่าย ๆ นอกจากนี้ เอดิเตอร์ซึ่งใช้งานในประเทศไทยก็ควรมีความสามารถในการจัดการเกี่ยวกับการใช้ภาษาไทยด้วย

โดยสรุป องค์ประกอบหลักๆ ของเอดิเตอร์สำหรับผู้ใช้ในประเทศไทยควรมีดังนี้

1. การจัดการเกี่ยวกับเนื้อหาภายในแฟ้มข้อมูล
2. การจัดการเกี่ยวกับแฟ้มข้อมูล
3. การจัดการด้านโปรแกรมสำหรับติดต่อกับผู้ใช้
4. การจัดการเกี่ยวกับการใช้งานภาษาไทย

2.4.1 การจัดการเกี่ยวกับเนื้อหาภายในแฟ้มข้อมูล

การจัดการเกี่ยวกับเนื้อหาภายในแฟ้มข้อมูลจะได้แก่ การเพิ่มข้อความใหม่ การลบข้อความเก่า และการแก้ไขข้อความเก่า โดยควรมีรายละเอียดดังนี้

การเพิ่มหรือแทรกข้อความใหม่ จะเป็นการนำเอาตัวอักขระทั้งหมดที่ประกอบขึ้นเป็นข้อความใหม่ไปเก็บไว้ในความจำหลัก ณ ตำแหน่งที่จองไว้ และนำข้อความดังกล่าวไปแสดงบนจอภาพ ณ ตำแหน่งที่เคอร์เซอร์ชี้ด้วย โดยที่ข้อความเดิมทางขวามือของเคอร์เซอร์ (ถ้ามี) จะถูกเลื่อนไปทางขวาเป็นจำนวนอักขระเท่ากับจำนวนอักขระในข้อความใหม่

การลบข้อความเก่า จะเป็นการนำเอาตัวอักขระทั้งหมดที่ประกอบขึ้นเป็นข้อความที่ต้องการลบออกจากความจำหลัก และลบข้อความดังกล่าวที่แสดงบนจอภาพ ณ ตำแหน่งที่เคอร์เซอร์ชี้ออก โดยข้อความที่อยู่ถัดไปทางขวาของข้อความที่ถูกลบออก (ถ้ามี) จะถูกเลื่อนมาทางซ้ายมือเป็นจำนวนอักขระเท่ากับจำนวนอักขระในข้อความเก่าที่ถูกลบออก

การแก้ไขข้อความเก่า มีอยู่ 2 วิธีใหญ่ๆ คือ วิธีที่หนึ่ง การลบข้อความเก่าทิ้งด้วยการเลือกข้อความที่ต้องการลบและกดปุ่มแป้นพิมพ์ Delete หรือ ใช้คำสั่งลบข้อความครั้งละ 1 คำ

จากนั้นจึงพิมพ์ข้อความใหม่ลงไปแทนที่ และสำหรับวิธีที่สอง จะเป็นการพิมพ์ข้อความใหม่ทับข้อความเก่า ซึ่งในการใช้วิธีพิมพ์แบบนี้ ผู้ใช้จะต้องกดปุ่มแป้นพิมพ์เพื่อเปลี่ยนสถานะการพิมพ์ข้อความจาก “พิมพ์แทรก” (insert) เป็น “พิมพ์ทับ” (replace) จากนั้นจึงพิมพ์ข้อความใหม่ลงไป ณ ตำแหน่งของข้อความเก่าที่ต้องการเปลี่ยน

2.4.2 การจัดการเกี่ยวกับเพิ่มข้อมูล

การจัดการเกี่ยวกับเพิ่มข้อมูล จะได้แก่ การสร้างเพิ่มข้อมูลใหม่ การจัดเก็บข้อความลงเพิ่มข้อมูล การลบเพิ่มข้อมูลเก่า และการจัดพิมพ์ข้อความภายในเพิ่มข้อมูลออกบนจอภาพและออกทางเครื่องพิมพ์

ในการจัดสร้างเพิ่มข้อมูลใหม่ จะกระทำก็ต่อเมื่อผู้ใช้กำหนดชื่อเพิ่มข้อมูลซึ่งถูกต้องตามหลักการตั้งชื่อเพิ่มข้อมูลของระบบปฏิบัติการ โดยเอ็ดิเตอร์จะจัดการเก็บชื่อเพิ่มข้อมูลที่สร้างใหม่ไว้ในสารบบเพิ่มข้อมูล (file directory) ของระบบปฏิบัติการพร้อมของเนื้อที่ในความจำหลักไว้สำหรับเก็บข้อความที่ผู้ใช้ป้อนเข้าทางแป้นพิมพ์ ในกรณีที่ผู้ใช้ไม่ได้กำหนดชื่อเพิ่มข้อมูล เอ็ดิเตอร์ก็จะถือว่าผู้ใช้ต้องการสร้างเพิ่มข้อมูลใหม่ตามชื่อปริยาย (default) ที่เอ็ดิเตอร์กำหนดเอาไว้

การจัดเก็บข้อความ จะเป็นการนำเอาข้อความที่อยู่ในความจำหลักไปเก็บไว้ในหน่วยความจำสำรองซึ่งได้แก่ อุปกรณ์ฟลอปปีดิสก์ หรืออุปกรณ์ฮาร์ดดิสก์ ภายใต้ชื่อเพิ่มข้อมูลที่ผู้ใช้กำหนด ในกรณีที่ชื่อเพิ่มข้อมูลที่ผู้ใช้กำหนดตรงกับชื่อเพิ่มข้อมูลที่อยู่แล้ว ข้อความที่มีอยู่แล้วในเพิ่มข้อมูลเดิมจะถูกแทนที่ด้วยข้อความใหม่

การจัดพิมพ์ข้อความออกทางเครื่องพิมพ์ จะเป็นการนำเอาข้อความที่เก็บอยู่ในหน่วยความจำหลักออกมาพิมพ์ทางเครื่องพิมพ์ ซึ่งแบ่งออกได้เป็น 2 ลักษณะดังนี้

1. พิมพ์ข้อความทั้งหมด

เป็นการพิมพ์ข้อความทั้งหมดออกทางเครื่องพิมพ์ ซึ่งเป็นการนำข้อความทั้งหมดที่เก็บอยู่ในหน่วยความจำ ส่งออกทางเครื่องพิมพ์

2. พิมพ์ข้อความเพียงบางส่วน

เป็นการพิมพ์ข้อความเพียงบางส่วนออกทางเครื่องพิมพ์ โดยผู้ใช้ต้องกำหนดส่วนของข้อความที่ต้องการพิมพ์ให้เป็นหน่วยที่เรียกว่าบล็อก (block) เสียก่อน จากนั้นจึงใช้คำสั่งของบล็อกสำหรับการส่งข้อความออกทางเครื่องพิมพ์

การลบเพิ่มข้อมูลเก่า จะเป็นการคืนเนื้อที่ที่ใช้จัดเก็บข้อมูลในหน่วยความจำสำรองให้กับระบบปฏิบัติการ ซึ่งจะทำให้สามารถนำเอาเนื้อที่ดังกล่าวกลับมาใช้ได้อีก โดยปกติแล้วในการลบเพิ่มข้อมูลเก่านี้ ระบบปฏิบัติการจะนำอักขระพิเศษบางอย่าง 1 ตัวอักขระไปแทนที่อักขระ

ตัวแรกของชื่อแฟ้มข้อมูลที่เก็บอยู่ใน File Allocation Table (FAT) เพื่อเป็นการบอกให้ระบบปฏิบัติการทราบว่าจะสามารถนำเอาเนื้อที่ที่เก็บข้อมูลเก่านี้มาใช้ได้ ซึ่งวิธีการลบข้อมูลดังกล่าวนี้ไม่ได้เป็นการลบข้อมูลที่เก็บอยู่ในหน่วยความจำสำรองจริงๆ ณ ขณะเวลาที่ใช้คำสั่งลบ

2.4.3 การจัดการด้านโปรแกรมสำหรับติดต่อกับผู้ใช้

ดังที่ทราบกันดีแล้วว่าในปัจจุบันนี้ คอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันของคนเรามากแต่ผู้ที่ใช้คอมพิวเตอร์ส่วนใหญ่มักจะประสบปัญหาเกี่ยวกับการใช้งานคอมพิวเตอร์เสมอ ทั้งนี้เนื่องมาจากการขาดความรู้ความเข้าใจลักษณะการใช้งานเครื่องคอมพิวเตอร์ ทำให้ประสบความยุ่งยากในการใช้งาน รวมทั้งขาดแรงจูงใจในการใช้คอมพิวเตอร์ ดังนั้นในการพัฒนาโปรแกรมระบบสำเร็จรูปใดๆจึงควรมีการพัฒนาโปรแกรมสำหรับติดต่อกับผู้ใช้ (user interface) ขึ้นมาเพื่อช่วยอำนวยความสะดวกในการทำงานของผู้ใช้ให้ใช้งานได้อย่างง่ายดาย และเป็นการดึงดูดใจผู้ใช้ให้หันมาใช้คอมพิวเตอร์กันมากขึ้น

โปรแกรมสำหรับติดต่อกับผู้ใช้ คือจุดที่เกิดการโต้ตอบกันระหว่างผู้ใช้และอุปกรณ์ที่ใช้ ซึ่งส่วนของโปรแกรมสำหรับติดต่อกับผู้ใช้ของการใช้งานคอมพิวเตอร์จะประกอบด้วยส่วนสำคัญ 3 ส่วนคือ ฮาร์ดแวร์ โปรแกรมที่พัฒนา และ เอกสารอ้างอิง โดยที่ฮาร์ดแวร์จะเป็นอุปกรณ์ที่ผู้ใช้ใช้ในการโต้ตอบกับเครื่องคอมพิวเตอร์ ซึ่งอุปกรณ์ดังกล่าวได้แก่ แป้นพิมพ์ เมาส์ และจอภาพ เป็นต้น ส่วนโปรแกรมที่พัฒนาจะใช้สำหรับตอบสนองความต้องการความสะดวกในการใช้งานของผู้ใช้ สำหรับเอกสารอ้างอิง เป็นส่วนสุดท้ายที่ใช้ในการอธิบาย และสนับสนุนการใช้และการบำรุงรักษาโปรแกรม

ในส่วนของการพัฒนาโปรแกรมสำหรับผู้ใช้ทั่วไป นอกจากจะพิจารณาถึงวัตถุประสงค์การทำงานแล้ว ผู้ผลิตควรคำนึงถึงส่วนของโปรแกรมสำหรับติดต่อกับผู้ใช้ ระหว่างผู้ใช้และโปรแกรม ด้วยการจัดสร้างรูปแบบการใช้งานในลักษณะมีการโต้ตอบระหว่างผู้ใช้และโปรแกรม (dialog style) มาใช้กับโปรแกรมที่พัฒนา โดยที่รูปแบบของการโต้ตอบกันระหว่างผู้ใช้กับโปรแกรมมีอยู่ด้วยกันหลายรูปแบบ เช่นการใช้ รายการเลือก (menu) วิธีการถามตอบ และการใช้ปุ่มฟังก์ชันคีย์ (function keys) เป็นต้น

2.4.4 การจัดการเกี่ยวกับการใช้งานภาษาไทย

โดยทั่วไปการนำภาษาไทยมาใช้กับคอมพิวเตอร์เมื่อพิจารณาจากลักษณะการแสดงผลจะมีอยู่ 2 ลักษณะคือ graphic mode และ text mode

การแสดงผลภาษาไทยในลักษณะ graphic mode จะสามารถแบ่งเป็น 2 ประเภทใหญ่ๆคือ แบบระยะอักขระคงที่ (monospaced fonts) และ แบบระยะอักขระเป็นสัดส่วน

(proportion spaced fonts) ตัวอย่างโปรแกรมสำเร็จรูปที่แสดงผลภาษาไทยในแบบระยะอักขระคงที่ ได้แก่ ไทยทิพย์ (VTHAI) และ CU-Writer เป็นต้น ตัวอย่างโปรแกรมสำเร็จรูปที่แสดงผลภาษาไทยในแบบระยะอักขระเป็นสัดส่วน ได้แก่ ไมโครซอฟต์วินโดว 3.0 และ Thai Win เป็นต้น

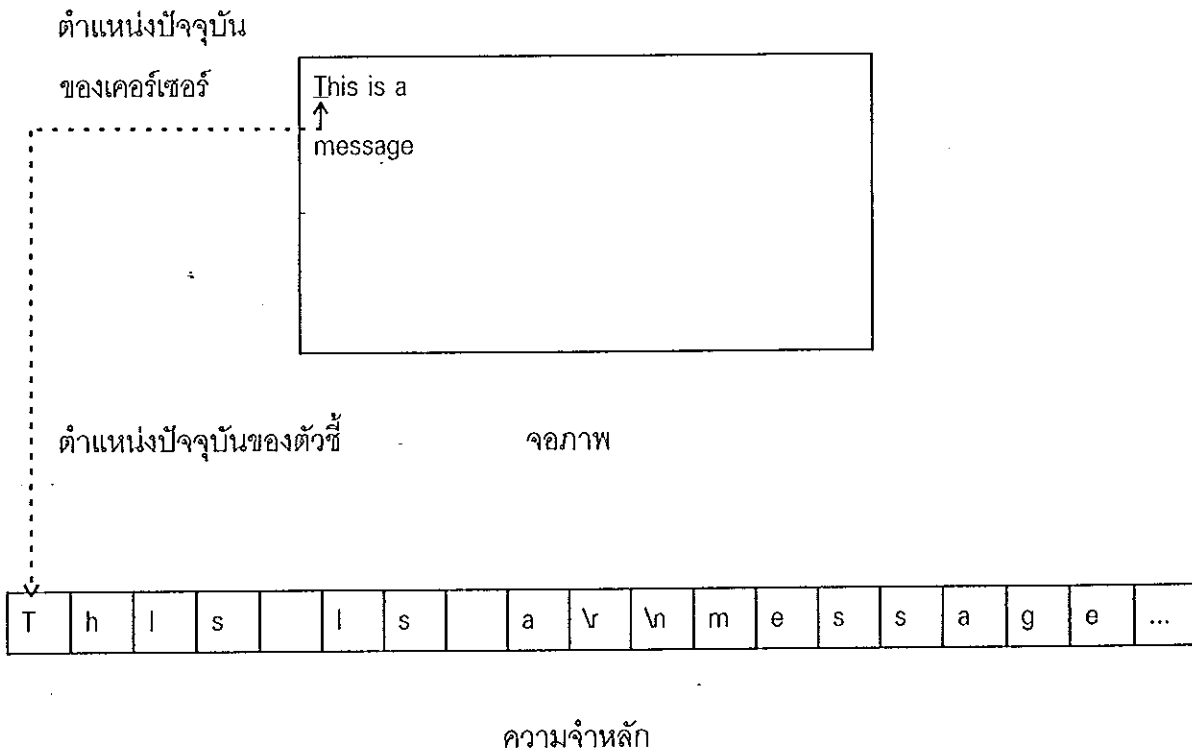
การแสดงผลภาษาไทยในลักษณะ text mode จะมีอยู่หลายวิธี แต่วิธีที่นิยมใช้กันเป็นส่วนใหญ่ก็คือการอาศัย Thai drivers ซึ่งเป็นโปรแกรมการดำเนินงานกับภาษาไทยที่ใช้งานร่วมกับการ์ดภาษาไทยซึ่งเป็นอุปกรณ์ฮาร์ดแวร์ โปรแกรมเหล่านี้จัดว่าเป็นระบบการดำเนินงานกับภาษาไทย ซึ่งจะต้องได้รับการสร้างขึ้นตามข้อกำหนดมาตรฐานของ IBM-PC BIOS และ MS-DOS และยิ่งไปกว่านั้นจะต้องทำงานร่วมกับโปรแกรมต่างประเทศได้ด้วย ในปัจจุบันระบบการดำเนินงานกับภาษาไทยที่ใช้กันอยู่จะถูกพัฒนาให้แสดงผลภาษาไทยได้ครั้งละ 25 บรรทัด

2.5 หลักการพื้นฐานบางประการเกี่ยวกับการพัฒนาเอดิเตอร์แบบเต็มจอภาพ

ปัญหาที่สำคัญที่สุดประการหนึ่งของการพัฒนาโปรแกรมเอดิเตอร์แบบเต็มจอภาพ คือการจัดการความสัมพันธ์ระหว่างข้อความที่เก็บอยู่ในความจำหลักและข้อความที่ปรากฏอยู่บนหน้าจอภาพ ณ ขณะเวลาหนึ่งๆ ซึ่งแตกต่างกันทางด้านกายภาพให้สอดคล้องกัน และเรื่องดังกล่าวเป็นสิ่งที่จำเป็นในการพัฒนาโปรแกรมเอดิเตอร์แบบเต็มจอภาพไม่ว่าจะใช้ขั้นตอนวิธีแบบใดก็ตาม

2.5.1 ตำแหน่งปัจจุบันของเคอร์เซอร์ และตัวชี้เลขที่ความจำหลัก

ตำแหน่งปัจจุบันของเคอร์เซอร์ก็คือจุดที่การทำงานขั้นต่อไปจะเกิดขึ้นบนจอภาพ ซึ่งเมื่อมองในลักษณะของโครงสร้างข้อมูลในความจำหลักที่ใช้ ตำแหน่งปัจจุบันก็คือตัวชี้ (pointer) ซึ่งชี้ไปยังเลขที่ความจำหลัก ณ ตำแหน่งของอักขระเดียวกับที่เคอร์เซอร์ขึ้นบนจอภาพ และเมื่อมองในลักษณะของจอภาพ ตำแหน่งปัจจุบัน ก็คือค่าคู่ลำดับของแถวและสดมภ์ของตำแหน่งของจอภาพที่เคอร์เซอร์อยู่ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 ความสัมพันธ์ระหว่างตำแหน่งปัจจุบันของเคอร์เซอร์บนจอภาพและตัวชี้บนความจำหลัก

จากรูปที่ 2.9 แสดงให้เห็นถึงข้อเท็จจริง 2 ประการ ที่เกิดขึ้นในการจัดการความสัมพันธ์ระหว่างตำแหน่งปัจจุบันของเคอร์เซอร์บนจอภาพและตำแหน่งปัจจุบันของตัวชี้เลขที่ความจำหลัก ดังนี้

ประการแรก จะพบว่าข้อความที่เก็บในความจำหลักจะมีลักษณะเชิงเส้น ส่วนข้อความที่ปรากฏบนจอภาพจะมีลักษณะเป็นตาราง 2 มิติ

ประการที่สอง ความจำหลักใหญ่พอที่จะเก็บข้อความทั้งหมดได้ ทำให้สามารถอ้างถึงตำแหน่งอักขระใดๆของข้อความบนความจำหลักได้ด้วยเลขที่ความจำหลักแบบต่อเนื่องตลอดได้ ส่วนจอภาพจะแสดงข้อความจากความจำหลักครั้งหนึ่งๆได้เพียงบางส่วนเท่านั้นซึ่งหมายความว่าความสัมพันธ์ระหว่างตำแหน่งแถวและสดมภ์ซึ่งถูกแทนด้วยคู่ลำดับ (X,Y) ใดๆบนหน้าจอ และตำแหน่งเลขที่ความจำหลัก ไม่ได้เป็นชนิดหนึ่งต่อหนึ่งแต่เป็นชนิดหนึ่งต่อหลาย กล่าวคือตำแหน่ง (X,Y) ใดๆบนจอภาพ ไม่ได้จำเพาะว่าจะต้องเป็นตำแหน่งใดตำแหน่งหนึ่งโดยเฉพาะบนความจำหลัก เมื่อมีการกดปุ่มแป้นพิมพ์ PgUp หรือ PgDn จะทำให้ข้อความบนจอภาพเปลี่ยนแปลงไปครั้งละ 25 บรรทัดของข้อความบนความจำหลัก (เดินหน้าไป 25 บรรทัดสำหรับ PgDn และถอยหลังไป 25 บรรทัดสำหรับ PgUp) โดยที่ตำแหน่งของเคอร์เซอร์ยังคงเดิม ซึ่งจะเห็นได้ว่าตำแหน่งของเคอร์เซอร์ในขณะนั้นไม่ใช่ตำแหน่งเดิมของตัวชี้บนความจำหลักก่อนการกดปุ่มแป้นพิมพ์ PgDn/PgUp รูปที่ 2.11 แสดงข้อความบนหน้าจอที่เปลี่ยนไปเมื่อมีการกดปุ่มแป้นพิมพ์ PgDn กับหน้าจอของรูปที่ 2.10

Line 2	Col 11	Ins	C:\RWAL1.TXT	Cols 20	Lines 46
--------	--------	-----	--------------	---------	----------

```

#include "proto1.h"
#include "externs.h"
    ↑
    |
    |_____ เคอร์เซอร์
/* Move current location left. */
void left(void)
{
if(curloc <= buf && scrnx <= 0)
- {
gotoxy(0, 18);
printf("***** %d ", *curloc);
return;

/* find end of line */
while(*curloc != '\n')
{
curloc++;
current_x++;
scrnx++;
screen_x++;

if(scrnx > LINE_LEN) {
F1-Help F2-Save F3-Load F4-Find F5-Replace F9-Thia/Eng F10-Menu Alt-X-Quit

```

รูปที่ 2.10 ตำแหน่งปัจจุบันของเคอร์เซอร์ และข้อความบนจอภาพ

```

Line 25   Col 11   Ins   C:\RWL1.TXT           Cols 130   Lines 46

    scrnx = screen_x = 0;

    display_dinamic_scrn(0,screen_ptr);
    }   ↑   เคอร์เซอร์
    col_diff++;

    }

}

else {

    curloc--;

    if(scrnx != current_x) {

        scrnx = 0;

        if(!WORD_LEFT)

            display_dinamic_scrn(0,screen_ptr);

    }

}

}

else

    curloc--;

}/* end left() */

F1-Help  F2-Save  F3-Load  F4-Find  F5-Replace  F9-Thia/Eng  F10-Menu  Alt-X-Quit

```

รูปที่ 2.11 ตำแหน่งปัจจุบันของเคอร์เซอร์และข้อความบนจอภาพหลังการกดปุ่ม
แป้นพิมพ์ PgDn

2.5.2 ขนาดของความจำหลักที่จอง

สิ่งที่ควรคำนึงอย่างหนึ่งสำหรับการทำงานของเอดิเตอร์ ก็คือการทำงานทุกอย่างกับข้อความ โดยเฉพาะการแทรก/แทนที่ และการลบอักขระ จะต้องอยู่ในขอบเขตของความจำหลักที่จองไว้และในกรณีของการอ่านข้อความจากแฟ้มข้อมูลไปเก็บไว้ในความจำหลักที่จองไว้ ถ้าหากแฟ้มข้อมูลมีขนาดใหญ่กว่าขนาดของความจำหลักที่จองไว้ เอดิเตอร์ก็จะไม่สามารถอ่านข้อความทั้งหมดจากแฟ้มข้อมูลไปเก็บยังความจำหลักที่จองไว้ได้ โดยจะแสดงข้อความบอกให้ผู้ใช้งานทราบ และภายหลังจากผู้ใช้กดปุ่มแป้นพิมพ์ตามที่เอดิเตอร์กำหนดเพื่อแสดงว่ารับทราบปัญหาที่เกิดขึ้นแล้ว เอดิเตอร์ก็จะยกเลิกคำสั่งอ่านข้อมูลจากแฟ้มข้อมูลดังกล่าว และกลับไปยังสถานะที่พร้อมรองรับคำสั่งใหม่

บทที่ 3

การออกแบบและการพัฒนาโปรแกรมเอดิเตอร์ PE

PE (PSU editor) เป็นเอดิเตอร์แบบเต็มจอภาพที่ถูกพัฒนาขึ้นมาด้วยภาษาซีเพื่อสำหรับใช้งานกับเครื่องไมโครคอมพิวเตอร์ทั้งชนิด PC และ AT ภายใต้ระบบปฏิบัติการ DOS และให้สามารถใช้งานได้ทั้งภาษาไทยและภาษาอังกฤษ โดยโปรแกรมไดรเวอร์ภาษาไทย (Thai driver program) ที่ใช้ร่วมกับเอดิเตอร์ PE ควรเป็น VTHAI.COM ไทยทิพย์ หรือ ภาษาไทยของบริษัท International Research Corporation LTD. (IRC) เพื่อให้ผู้ใช้จะได้ใช้งานได้ง่าย

จากผลการศึกษาเอดิเตอร์แบบเต็มจอภาพหลายชนิดที่มีการใช้งานภายใต้ระบบปฏิบัติการ DOS พบว่าเอดิเตอร์แบบเต็มจอภาพส่วนใหญ่จะมีความสามารถหลักๆเช่นเดียวกันซึ่งได้แก่ การจัดการเกี่ยวกับการเลื่อนตำแหน่งเคอร์เซอร์ การค้นหาและแทนที่คำ/ข้อความ การแทรก/แทนที่คำ/ข้อความ การลบตัวอักษร การทำงานกับกลุ่มของข้อความ และการจัดการเกี่ยวกับเนื้อหาในแฟ้มข้อมูล แต่สิ่งที่แตกต่างกันอย่างเห็นได้ชัดคือเอดิเตอร์เหล่านั้นยังขาดความสามารถหลายประการเกี่ยวกับการจัดการกับแฟ้มข้อมูลซึ่งได้แก่ การคัดลอกแฟ้มข้อมูลที่อยู่ในหน่วยความจำสำรอง การลบแฟ้มข้อมูลที่อยู่ในหน่วยความจำสำรอง และการเปลี่ยนชื่อแฟ้มข้อมูลที่อยู่ในหน่วยความจำสำรอง นอกจากนี้ในกรณีของการจัดเก็บข้อความไว้ในแฟ้มข้อมูลที่มีอยู่แล้ว เอดิเตอร์แบบเต็มจอภาพที่มีการใช้งานอยู่ในปัจจุบันนี้จะกำหนดทางเลือกในการจัดเก็บข้อความเอาไว้เพียง 2 อย่างซึ่งได้แก่การจัดเก็บหรือไม่จัดเก็บข้อความใหม่แทนที่ข้อความเดิมในแฟ้มข้อมูล ผู้ใช้จะไม่สามารถนำเอาข้อความจากแฟ้มข้อมูลหนึ่งไปจัดเก็บเพิ่มเติมต่อท้ายข้อความในแฟ้มข้อมูลอื่นได้ ดังนั้นผู้วิจัยจึงได้เพิ่มเติมความสามารถเหล่านี้ไว้ในโปรแกรมเอดิเตอร์ PE

ในการพัฒนาโปรแกรมระบบสำเร็จรูปสำหรับให้ผู้ใช้ทั่วไปจำนวนมากใช้ สิ่งหนึ่งที่ผู้พัฒนาโปรแกรมต้องคำนึงถึงอย่างมาก คือวิธีการติดต่อใช้งานระหว่างผู้ใช้และโปรแกรม (user interface) ผู้พัฒนาควรออกแบบโปรแกรมให้มีลักษณะการใช้งานที่เป็นมิตรกับผู้ใช้ (user friendly) กล่าวคือ ผู้ใช้ควรสามารถใช้งานโปรแกรมเองได้ มีความรู้สึกสะดวกสบายในการใช้ ไม่ต้องถามหรือพึ่งพาคณอื่นมากนัก ดังนั้น ในการออกแบบเพื่อพัฒนาโปรแกรมเอดิเตอร์ PE ผู้พัฒนาได้คำนึงถึงจุดนี้เป็นอย่างมาก

3.1 ลักษณะการติดต่อใช้งาน

ในการพัฒนาโปรแกรมเอดิเตอร์ PE ผู้พัฒนาได้ออกแบบวิธีการติดต่อใช้งานของผู้ใช้ใน 2 ลักษณะ ซึ่งเป็นที่นิยมกันทั่วไป คือ การใช้งานด้วยคำสั่งในรายการเลือกแบบดิ่งลง และการใช้งานด้วยคำสั่งที่แทนด้วยปุ่มแป้นพิมพ์ต่างๆ

3.1.1 การใช้งานด้วยคำสั่งในรายการเลือก

ในการใช้งานด้วยคำสั่งในรายการเลือกนั้นจะอาศัยการจัดรูปแบบรายการเลือกในลักษณะที่เรียกว่า รายการเลือกแบบดิ่งลง โดยที่รายการเลือกจะประกอบด้วยตัวเลือกหลายๆ ซึ่งในแต่ละตัว เลือกหลักจะประกอบด้วยกลุ่มคำสั่งที่ทำงานลักษณะเดียวกัน เช่น คำสั่งเลื่อนเคอร์เซอร์ (ไปทางซ้าย 1 อักขระ ไปทางขวา 1 อักขระ ลงล่าง 1 บรรทัด และขึ้นบน 1 บรรทัด) อย่างไรก็ตามหน้าที่การทำงานในส่วนของรายการเลือกนี้ จะไม่ครอบคลุมการทำงานทั้งหมดของโปรแกรม จะครอบคลุมเฉพาะการทำงานที่สำคัญเท่านั้น และในส่วนของการทำงานที่เหลือนี้ จะใช้คำสั่งที่แทนด้วยปุ่มแป้นพิมพ์ต่างๆ

3.1.2 การใช้งานด้วยคำสั่งที่แทนด้วยปุ่มแป้นพิมพ์ต่างๆ

ในการใช้งานด้วยคำสั่งที่แทนด้วยปุ่มแป้นพิมพ์ต่างๆ จะเป็นการใช้งานโดยการกดปุ่มแป้นพิมพ์บนแป้นพิมพ์ให้ตรงกับคำสั่งการทำงานของโปรแกรม การทำงานในส่วนนี้ ผู้ใช้จำเป็นต้องทราบคำสั่ง และปุ่มแป้นพิมพ์ที่สมนัยกันในการใช้งานโปรแกรม ในกรณีที่ผู้ใช้ต้องการดูคำสั่ง และปุ่มแป้นพิมพ์ที่สมนัยกัน ก็สามารถขอความช่วยเหลือ (help) ของโปรแกรม โดยรายละเอียดคำสั่งและปุ่มแป้นพิมพ์ที่สมนัยกัน ดูได้จากภาคผนวก ก

3.2 รูปแบบและขนาดของหน้าจอ

สำหรับหน้าจอของการทำงานของโปรแกรมเอดิเตอร์ PE ถูกออกแบบให้อยู่ในรูปการทำงานทุกอย่างอยู่ในกรอบหน้าต่างเดียว โดยขนาดของหน้าต่าง สามารถบรรจุข้อความได้ทั้งสิ้น 23 บรรทัด และแต่ละบรรทัดบรรจุอักขระได้ไม่จำกัดจำนวน หน้าต่างที่ถูกออกแบบมี 2 รูปแบบ

รูปแบบที่ 1 เป็นหน้าจอที่ได้หลังจากการเรียกใช้โปรแกรม PE ซึ่งส่วนประกอบของหน้าจอ

จะแบ่งออกเป็น 3 ส่วน ดังแสดงในรูปที่ 3.1 (ก) โดยมีรายละเอียดดังนี้

ส่วนที่ 1 เป็นข้อมูลที่บอกให้ผู้ใช้ทราบถึงตำแหน่งเลขที่บรรทัด และเลขที่สดมภ์ของเคอร์เซอร์ สถานะของการพิมพ์ ชื่อแฟ้มข้อมูลที่กำลังใช้ จำนวนของสดมภ์ทั้งหมดที่มีอยู่ในแถวที่เคอร์เซอร์อยู่ และจำนวนทั้งหมดของแถวที่มีอยู่ในแฟ้มข้อมูล

ส่วนที่ 2 เป็นเนื้อที่สำหรับให้ผู้ใช้จัดพิมพ์ข้อความ

ส่วนที่ 3 เป็นส่วนที่ใช้แสดงคำสั่งการใช้งานโปรแกรมที่สำคัญๆซึ่งเป็นการอำนวยความสะดวกให้กับผู้ใช้

รูปแบบที่ 2 เป็นหน้าจอที่ได้หลังจากการเรียกใช้โปรแกรม PE และผู้ใช้กดปุ่มฟังก์ชันคีย์ F10 ซึ่งส่วนประกอบของหน้าจอ จะแบ่งออกเป็น 4 ส่วน ดังแสดงในรูปที่ 3.1 (ข) โดยมีรายละเอียดดังนี้

ส่วนที่ 1 เป็นข้อมูลที่บอกให้ผู้ใช้ทราบถึง ชื่อของโปรแกรมที่กำลังใช้งาน (PE) ฉบับที่ของโปรแกรม (version) ปีที่จัดสร้างโปรแกรมได้สำเร็จ และ สถานที่จัดสร้างโปรแกรม

ส่วนที่ 2 เป็นรายการหลักของการใช้งานเอดิเตอร์ PE ซึ่งมีอยู่ทั้งหมด 5 รายการ โดยแต่ละรายการหลักจะประกอบด้วยรายการย่อยจำนวนหนึ่ง รายละเอียดดูได้จากภาคผนวก ก

ส่วนที่ 3 เป็นส่วนที่ใช้แสดงข้อความของแฟ้มข้อมูลที่ผู้ใช้กำลังจัดการ

ส่วนที่ 4 เป็นส่วนที่ใช้สำหรับบอกให้ผู้ใช้ทราบถึงวิธีการเลิกใช้งานเอดิเตอร์ PE ในรูปแบบที่ 2 เพื่อกลับไปสู่การทำงานในรูปแบบที่ 1

```

ส่วนที่ 1 |
|-----|-----|-----|-----|-----|-----|
| ECHO OFF |
| set conspec=c:\command.com |
| : \DOS\SHARE.EXE |
| set prompt $p$g |
| verify ON |
| PATH=C:\WINDOWS;C:\DOS;C:\TOOLS;C:\ZIP;C:\SCAN;C:\NTP\BIN;C:\NU7; |
| set NU=C:\NU6 |
| set CW=CW152 |
| SHIELD /LOCK/SWAP/CV |
|
ส่วนที่ 2 |
|
ส่วนที่ 3 |
|-----|-----|-----|-----|-----|-----|
| ESC=QUIT |

```

(ก)

```

ส่วนที่ 1 |
ส่วนที่ 2 |-----|-----|-----|-----|-----|-----|
| F1=Copy to Clipboard | F2=Search | F3=Go to | F4=Quit |
|
ส่วนที่ 3 |
|-----|-----|-----|-----|-----|-----|
| : \DOS\SHARE.EXE |
| et prompt $p$g |
| erify ON |
| ATH=C:\WINDOWS;C:\DOS;C:\TOOLS;C:\ZIP;C:\SCAN;C:\NTP\BIN;C:\NU7; |
| et NU=C:\NU6 |
| et CW=CW152 |
| SHIELD /LOCK/SWAP/CV |
|
ส่วนที่ 4 |
|-----|-----|-----|-----|-----|-----|
| ESC=QUIT |

```

(ข)

รูปที่ 3.1 ส่วนประกอบของหน้าจอขณะใช้งานเอดิเตอร์ PE

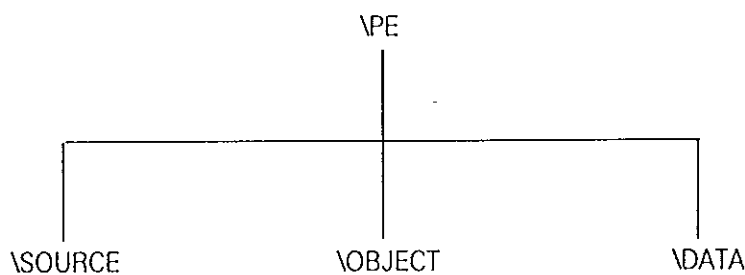
3.3 สารบบเพิ่มข้อมูล

สารบบเพิ่มข้อมูล สำหรับเก็บเพิ่มข้อมูลของงานวิจัยนี้เป็นแผนผังต้นไม้ (tree diagram) ดังแสดงในรูปที่ 3.2 มีรายละเอียดดังนี้

สารบบ “\PE\SOURCE” เป็นที่จัดเก็บเพิ่มข้อมูลโปรแกรมต้นฉบับซึ่งเขียนด้วยภาษาซีของ
 เอดิเตอร์ PE ที่ได้พัฒนาขึ้นทั้งหมด

สารบบ “\PE\OBJECT” เป็นที่จัดเก็บเพิ่มข้อมูลโปรแกรมภาษาเครื่องของเอดิเตอร์ PE

สารบบ “\PE\DATA” เป็นที่จัดเก็บเพิ่มข้อมูลต่างๆที่ได้จากการสร้างด้วยเอดิเตอร์ PE
 ของผู้ใช้



รูปที่ 3.2 สารบบเพิ่มข้อมูลสำหรับเอดิเตอร์ PE

3.4 โครงสร้างข้อมูล

โครงสร้างข้อมูล สำหรับเก็บข้อความที่ผู้ใช้ป้อนเข้ามาทางแป้นพิมพ์ หรือที่ถูกอ่านมาจาก
 เพิ่มข้อมูลในความจำสำรอง มีความสำคัญต่อการดำเนินงานของโปรแกรมเอดิเตอร์ PE อย่างยิ่ง
 ผู้วิจัยได้เลือกใช้โครงสร้างข้อมูลแบบลิสต์ลำดับ (sequential list) โดยการดำเนินงานกับโครงสร้าง-
 ข้อมูลมีอยู่ 4 ชนิด คือ

- การเพิ่มข้อความใหม่
- การลบข้อความเก่าออก
- การเปลี่ยนแปลงแก้ไขข้อความเก่าด้วยข้อความใหม่
- การค้นหาค่า/ข้อความเก่า

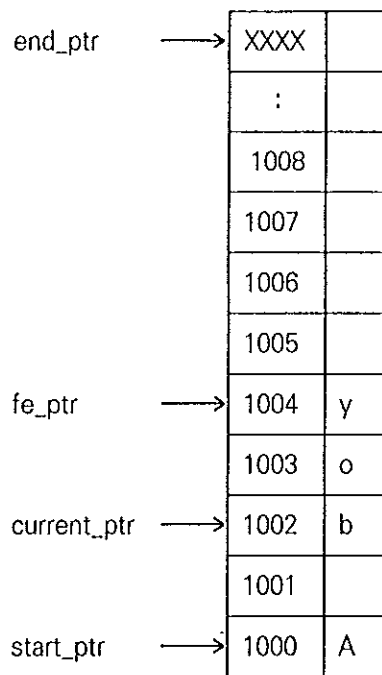
โครงสร้างข้อมูลแบบลิสต์ลำดับที่ใช้ จะใช้เนื้อที่ความจำหลักจำนวน 50 KB ต่อเนื่องกัน โดย
 มีตัวชี้ (pointer) หลัก 4 ตัว ที่ใช้ในการดำเนินงาน

- start_ptr เป็นตัวชี้บอกเลขที่ความจำหลักแรกบนความจำหลักที่จองไว้
- current_ptr เป็นตัวชี้บอกเลขที่ความจำหลักของอักขระในความจำหลักที่จองไว้ที่สมนัย
 กับตำแหน่งปัจจุบันของเคอร์เซอร์บนจอภาพ

- end_ptr เป็นตัวชี้บอกเลขที่ความจำหลักตัวสุดท้ายบนความจำหลักที่จองไว้
- fe_ptr เป็นตัวชี้บอกเลขที่ความจำหลักของอักขระตัวสุดท้ายของข้อความในความจำหลักที่จองไว้

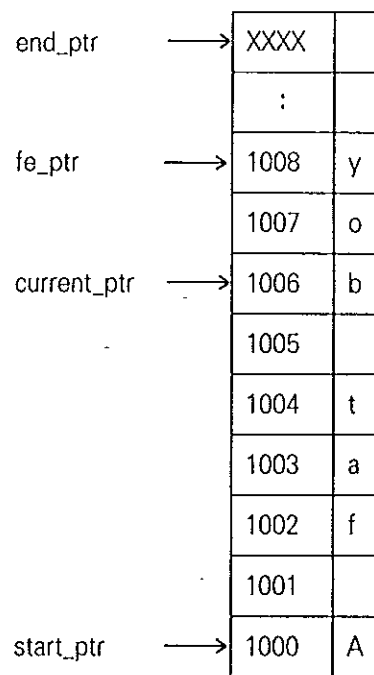
ขณะเริ่มต้นการทำงานของแฟ้มข้อมูลใหม่ ตัวชี้ start_ptr current_ptr และ fe_ptr จะชี้ไปยังจุดเริ่มต้นของความจำหลักที่จองไว้ ในขณะที่ end_ptr จะชี้ไปยังจุดสิ้นสุดของความจำหลักดังกล่าว

การเพิ่มข้อความใหม่ เป็นการเพิ่มหรือแทรกข้อความใหม่ที่ละอักขระไปเก็บในความจำหลักที่จองไว้ ณ ตำแหน่งที่ชี้โดย current_ptr โดยอักขระเดิมที่อยู่ ณ ตำแหน่งที่ current_ptr จะถูกเคลื่อนย้ายไปเก็บยังความจำหลักที่อยู่ถัดไป (ณ ตำแหน่ง current_ptr+1) จากนั้นอักขระตัวใหม่จะถูกเก็บในความจำหลัก ณ ตำแหน่งที่ชี้โดย current_ptr และภายหลังจากที่อักขระถูกเก็บในความจำหลักแล้ว ค่าของ current_ptr และ fe_ptr จะถูกเพิ่มขึ้นครั้งละ 1 เสมอ ในการเพิ่มข้อความใหม่นี้ ผู้ใช้สามารถเพิ่มได้เรื่อยๆ ตราบที่ fe_ptr ยังมีค่าน้อยกว่า end_ptr เมื่อใดก็ตามที่ผู้ใช้เพิ่มข้อความในความจำหลักซึ่งเป็นผลให้ fe_ptr มีค่าเท่ากับ end_ptr เอดีเดอริกก็จะแสดงข้อความบอกให้ผู้ใช้ทราบว่าไม่สามารถเพิ่มข้อความได้อีก รูปที่ 3.3(ก) และ 3.3(ข) เป็นตัวอย่างของโครงสร้างข้อมูลก่อนและภายหลังเพิ่มข้อความ "fat " ณ ตำแหน่งที่ถูกชี้โดย current_ptr ตามลำดับ



(ก)

ก่อนการเพิ่มข้อความ "fat "



(ข)

ภายหลังการเพิ่มข้อความ "fat "

รูปที่ 3.3 โครงสร้างข้อมูลก่อนและหลังการเพิ่มข้อความ "fat "

การลบข้อความเก่า เป็นการลบข้อความทีละอักขระโดยอาศัย `current_ptr` เป็นตัวกำหนดจุดเริ่มต้นของข้อความที่ต้องการลบ และนำข้อความทั้งหมดในความจำหลักที่อยู่ถัดจากข้อความที่ถูกลบมาแทนที่ข้อความเก่าในความจำหลักที่ถูกชี้โดย `current_ptr` ด้วยเหตุนี้จึงไม่มีการเปลี่ยนแปลงค่าของ `current_ptr` รูปที่ 3.4(ก) และ 3.4(ข) เป็นตัวอย่างของโครงสร้างข้อมูลก่อนและภายหลังจากลบข้อความ "fat" ณ ตำแหน่งที่ถูกชี้โดย `current_ptr` ตามลำดับ

<code>end_ptr</code>	→	XXXX	
		:	
<code>fe_ptr</code>	→	1008	y
		1007	o
		1006	b
		1005	
		1004	t
		1003	a
<code>current_ptr</code>	→	1002	f
		1001	
<code>start_ptr</code>	→	1000	A

(ก)

ก่อนลบข้อความ "fat "

<code>end_ptr</code>	→	XXXX	
		:	
		1008	
		1007	
		1006	
		1005	
<code>fe_ptr</code>	→	1004	y
		1003	o
<code>current_ptr</code>	→	1002	b
		1001	
<code>start_ptr</code>	→	1000	A

(ข)

ภายหลังจากลบข้อความ "fat "

รูปที่ 3.4 โครงสร้างข้อมูลก่อนและหลังการลบข้อความ "fat "

การเปลี่ยนแปลงแก้ไขข้อความเก่า เป็นวิธีการลบข้อความเก่าออกจากโครงสร้างข้อมูล แล้วเพิ่มข้อความใหม่เข้าไปในโครงสร้างข้อมูล หรือการพิมพ์ทับข้อความเดิมด้วยข้อความใหม่ ซึ่งการพิมพ์ทับข้อความเดิมด้วยข้อความใหม่นี้จะมีการจัดการกับข้อมูลในโครงสร้างข้อมูลคล้ายคลึงกับการเพิ่มข้อความใหม่ จะแตกต่างกันเฉพาะตรงที่ไม่มีการเคลื่อนย้ายอักขระในความจำหลัก ณ ตำแหน่งที่ชี้โดย `current_ptr` ไปยังตำแหน่งถัดไป ก่อนที่จะมีการเพิ่มอักขระตัวใหม่ รูปที่ 3.5(ก) และ 3.5(ข) เป็นตัวอย่างของข้อมูลในความจำหลักก่อนและภายหลังการเปลี่ยนแปลงข้อความ “fat boy” เป็น “thin boy” ณ ตำแหน่งที่ถูกชี้โดย `current_ptr` ตามลำดับ

<code>end_ptr</code>	→	XXXX	
		:	
		1010	
		1009	
<code>fe_ptr</code>	→	1008	y
		1007	o
		1006	b
		1005	
		1004	t
		1003	a
<code>current_ptr</code>	→	1002	f
		1001	
<code>start_ptr</code>	→	1000	A

(ก)

ก่อนเปลี่ยนข้อความ “fat boy”
เป็น “thin boy”

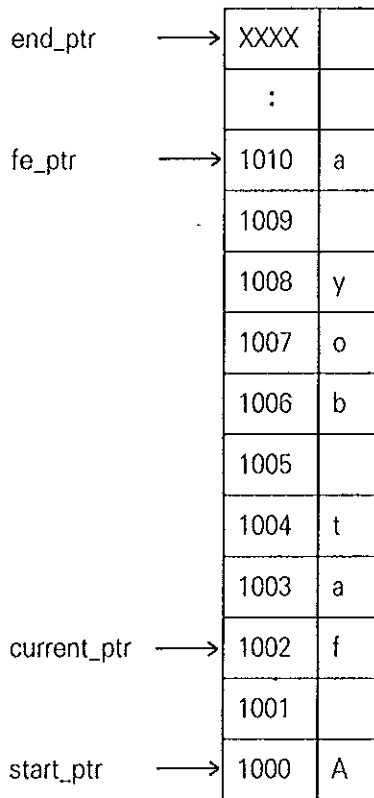
<code>end_ptr</code>	→	XXXX	
		:	
<code>fe_ptr, current_ptr</code>	→	1010	
		1009	y
		1008	o
		1007	b
		1006	
		1005	n
		1004	i
		1003	h
		1002	t
		1001	
<code>start_ptr</code>	→	1000	A

(ข)

ภายหลังเปลี่ยนข้อความ “fat boy”
เป็น “thin boy”

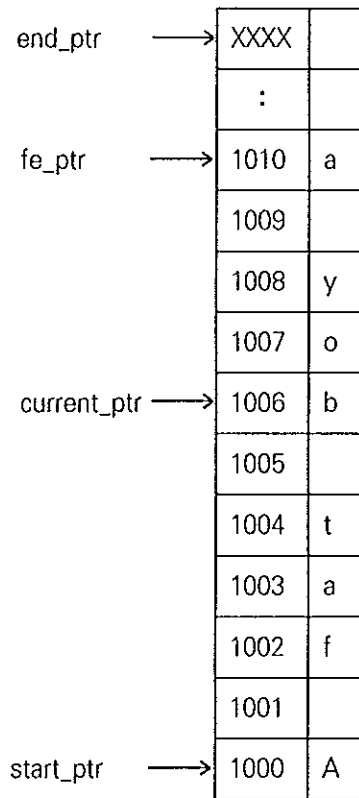
รูปที่ 3.5 โครงสร้างข้อมูลก่อนและหลังการเปลี่ยนข้อความ “fat boy” เป็น “thin boy”

การค้นหาคำ/ข้อความเก่า เป็นการค้นหาคำหรือข้อความที่อยู่ในความจำหลักที่จองไว้ ในกรณีที่สามารถค้นพบคำ/ข้อความที่ผู้ใช้ระบุ `current_ptr` จะถูกเลื่อนไปยังเลขที่ความจำหลักที่เก็บอักขระตัวแรกของคำ/ข้อความดังกล่าว รูปที่ 3.6(ก) และ 3.6(ข) เป็นตัวอย่างของโครงสร้างข้อมูลก่อนและภายหลังการค้นหาคำ "boy" ณ ตำแหน่งที่ถูกชี้โดย `current_ptr` ตามลำดับ ในกรณีที่ไม่สามารถค้นพบคำ/ข้อความที่ผู้ใช้ระบุ `current_ptr` ก็จะมียังคงชี้ ณ เลขที่ความจำหลักเดิมก่อนที่จะทำการค้นหาคำ/ข้อความ



(ก)

ก่อนการค้นหาคำ "boy"



(ข)

ภายหลังการค้นหาคำ "boy"

รูปที่ 3.6 โครงสร้างข้อมูลก่อนและหลังการค้นหาคำ "boy"

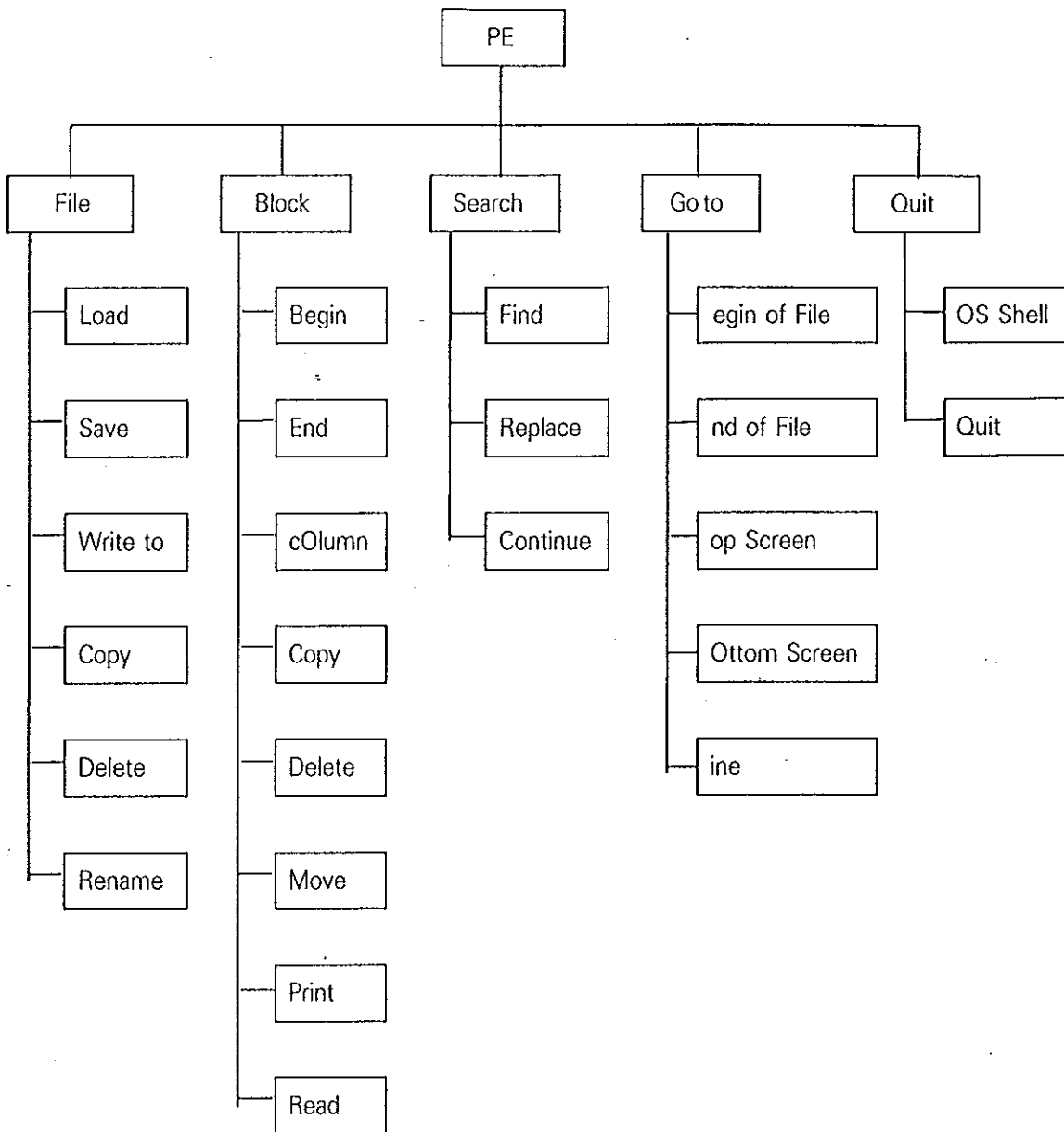
สำหรับการเริ่มต้นทำงานกับข้อความในแฟ้มข้อมูลเดิมที่มีอยู่แล้ว ตัวชี้ `start_ptr` และ `current_ptr` ก็จะชี้ไปยังจุดเริ่มต้นของข้อความจำหลักที่จองไว้ ในขณะที่ `fe_ptr` จะชี้ไปยังเลขที่ความจำหลักซึ่งเก็บอักขระตัวสุดท้ายของแฟ้มข้อมูลที่กำลังใช้งาน และ `end_ptr` จะชี้ไปยังจุดสิ้นสุดของข้อความจำหลักดังกล่าว เช่นเดียวกันกับการเริ่มต้นทำงานกับแฟ้มข้อมูลใหม่ การทำงานกับแฟ้มข้อมูลเดิมที่มีอยู่แล้ว ไม่ว่าจะเป็นการเพิ่มข้อความใหม่ ลบข้อความเก่า หรือเปลี่ยนแปลงข้อความเก่าด้วยข้อความใหม่ จะเป็นหน้าที่ของผู้ใช้ที่จะเป็นผู้เลื่อน `current_ptr` (โดยผ่านทางเคอร์เซอร์) ไปยังตำแหน่งของอักขระของข้อความที่ต้องการทำงานด้วย

3.5 โครงสร้างระบบ

เอดิเตอร์ PE ถูกออกแบบการใช้งานให้อยู่ในรูปที่คำสั่งต่างๆถูกเลือกจากรายการเลือกแบบดิ่งลง โดยมีรายการหลัก (อยู่ในแถวแนวนอน) และรายการย่อยในแต่ละรายการหลัก (อยู่ในแถวแนวดิ่ง) ดังแสดงในรูปที่ 3.7 (ดูรูปที่ 3.1(ข) ประกอบ)

สำหรับการดำเนินงานในส่วนต่างๆของเอดิเตอร์ PE ถูกออกแบบให้อยู่ในรูปโปรแกรมย่อยหรือที่เรียกว่าโมดูล (module) ซึ่งมีการเชื่อมโยงถึงกันและกัน โครงสร้างระบบหลักในการดำเนินงานจะคล้ายคลึงกับโครงสร้างรายการเลือกแบบดิ่งลงในรูปที่ 3.7 โดยการดำเนินงานของแต่ละรายการย่อยในรายการหลักจะต้องเรียกใช้โปรแกรมย่อยสำหรับดำเนินงานต่างๆตามวัตถุประสงค์ของการดำเนินงานของรายการย่อยนั้นๆ ตารางที่ 3.1 แสดงชื่อรายการย่อยในแต่ละรายการหลักของรูปที่ 3.7 และชื่อโปรแกรมย่อยที่ถูกเรียกใช้ตามลำดับตัวอักษรของชื่อ แต่ละโปรแกรมย่อยในตารางที่ 3.1 ที่ถูกเรียกใช้จะมีหน้าที่ในการดำเนินงานเฉพาะอย่าง ดังรายละเอียดแสดงในตารางที่

3.2



รูปที่ 3.7 แผนภูมิรายการเลือกแบบดิ่งลง

ตารางที่ 3.1 ชื่อรายการย่อยในแต่ละรายการหลักของรูปที่ 3.7 และชื่อโปรแกรมย่อย
ที่ถูกเรียกใช้ตามลำดับตัวอักษรของชื่อ

รายการย่อยในรายการหลัก File	โปรแกรมย่อยที่เรียกใช้
Load	clear_screen
	cursoff
	curson
	display_scrn
	edit_gets
	initial_values
	goto_xy
	load
	save
	set_drive_path
	window_construct
	window_close
	write_char
Save	clear_screen
	cursoff
	curson
	display_scrn
	edit_gets
	goto_xy
	load
save	
write_char	
Write to	edit_gets
	print_process
	save

ตารางที่ 3.1 (ต่อ)

	set_drive_path
	window_construct
	window_close
Copy	copy_file
	cursoff
	curson
	edit_gets
	goto_xy
	window_construct
	window_close
	write_char
Delete	cursoff
	curson
	delete_file
	edit_gets
	window_construct
	window_close
Rename	cursoff
	curson
	edit_gets
	rename_file
	window_construct
	window_close
รายการย่อยในรายการหลัก Block	โปรแกรมย่อยที่เรียกใช้
Begin	write_char
End	display_dinamic_scrn
	window_construct
	window_close
cOolumn	column_block

ตารางที่ 3.1 (ต่อ)

	display_dinamic_scrn
	edit_gets
	window_construct
	window_close
Copy	copy_block
	display_dinamic_scrn
	downchanged_screen_ptr
	find_current_screen
	space_increasing
Delete	dele_block
	display_dinamic_screen
	downchanged_screen_ptr
	down_screen_ptr
Move	copy_block
	dele_block
	move_block
	space_increasing
Print	edit_gets
	init_prn
	print_process
	print
	prtrdy
	window_construct
	window_close
Read	edit_gets
	read_file
	window_construct
	window_close

ตารางที่ 3.1 (ต่อ)

รายการย่อยในรายการหลัก Search	โปรแกรมย่อยที่เรียกใช้
Find	clear_screen
	cursoff
	curson
	display_scrn
	display_dinamic_scrn
	edit_gets
	find_current_screen
	get_name
	goto_xy
	screen_xy
	search
	window_construct
	window_close
	word_uplow_strncmp
Replace	clear_screen
	clr_from_x_to_eol
	display_dinamic_scrn
	display_scrn
	edit_gets
	get_name
	goto_xy
	printline
	replace
	replace_process
	screen_xy
	search
	window_construct
	window_close

ตารางที่ 3.1 (ต่อ)

Continue	continue_search
	replace
	search
รายการย่อยในรายการหลัก Go to	โปรแกรมย่อยที่เรียกใช้
Begin of File	clear_screen
	display_scrn
	goto_xy
	home
End of File	clear_screen
	display_dinamic_scrn
	endline
	gotoend
	screen_xy
Top Screen	goto_xy
	topscreen
	upline
bOttom Screen	bottomscreen
	goto_xy
Line	clear_screen
	cursoff
	curson
	display_scrn
	edit_gets
	line_number
	screen_xy
	window_construct
	window_close
รายการย่อยในรายการหลัก Quit	โปรแกรมย่อยที่เรียกใช้
OS Shell	clear_screen

ตารางที่ 3.1 (ต่อ)

	os_shell
	write_char
Quit	clear_screen
	edit_gets
	goto_xy
	save
	window_construct
	window_close

ในการพัฒนาโปรแกรมสำหรับงานวิจัยครั้งนี้ได้แบ่งการทำงานทั้งหมดออกเป็นโปรแกรมย่อยๆ ซึ่งแต่ละโปรแกรมจะทำงานเฉพาะอย่างและเมื่อนำโปรแกรมย่อยๆเหล่านี้มาเชื่อมโยงกันหลักที่ทำงานได้ตรงตามวัตถุประสงค์

ตารางที่ 3.2 หน้าทีการทำงานของแต่ละโปรแกรมย่อย

ชื่อโปรแกรมย่อย	หน้าที่การทำงาน
add_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
beginline	เลื่อนเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นของบรรทัด
bottomscreen	เลื่อนเคอร์เซอร์ไปยังบรรทัดสุดท้ายของจอภาพ
change_eliminate	เปลี่ยนตัวอักษรจาก capital letter เป็นตัวอักษรปกติ
clrline	ทำให้เป็นบรรทัดว่าง
clearlines	ทำให้บรรทัดปัจจุบันถึงบรรทัดสุดท้ายของจอเป็นบรรทัดว่าง
clear_screen	ทำให้หน้าจอว่าง
clr_from_x_to_eol	ทำให้ตำแหน่งปัจจุบันของเคอร์เซอร์ถึงตำแหน่งสุดท้ายของบรรทัดเป็นที่ว่าง
column_block	สร้างบล็อกชนิดคอลัมน์
copy_block	ทำการคัดลอกข้อความในบล็อกไปยังตำแหน่งใดๆในเพิ่มข้อมูล หรือไปยังเพิ่มข้อมูลอื่น
copy_file	ทำการสำเนาเพิ่มข้อมูล
continue_search	ทำการค้นหาข้อความเดิมต่อไป
current_kill_to_eol	ลบตัวอักษรจากตำแหน่งปัจจุบันของเคอร์เซอร์ถึงสุดบรรทัด
cursoff	ซ่อนเคอร์เซอร์
curson	แสดงเคอร์เซอร์
downchanged_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
delete_char	ลบครั้งละ 1 อักขระ
delete_file	ทำการลบเพิ่มข้อมูล
dele_block	ทำการลบข้อความในบล็อก

ตารางที่ 3.2 (ต่อ)

ชื่อโปรแกรมย่อย	หน้าที่การทำงาน
decrease_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
display_scrn	แสดงข้อความบนจอภาพ
down_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
downline	เลื่อนเคอร์เซอร์ลง 1 บรรทัด
downscreen	เลื่อนจอภาพลง 1 บรรทัด
display_dinamic_scrn	แสดงข้อความบนจอภาพ
drive_path	ทำการค้นหาเส้นทางและไดรว์ ปัจจุบัน
edit_clr_eol	ทำให้ตำแหน่งปัจจุบันถึงตำแหน่งสุดท้ายของบรรทัดเป็นที่ว่าง
edit	เป็นโปรแกรมหลักซึ่งจะไปเรียกใช้โปรแกรมย่อยอื่นๆ
edit_gets	รับข้อความจากแป้นพิมพ์
empty_position	ใช้ในการตรวจสอบตำแหน่งบนจอภาพว่ามีตัวอักษรหรือไม่
endline	เลื่อนเคอร์เซอร์ไปยังท้ายสุดของบรรทัด
find_current_screen	ค้นหาหน้าจอปัจจุบัน
find_screen_ptr	กำหนดตำแหน่งเริ่มต้นของจอภาพปัจจุบัน
find_lend	ค้นหาจำนวนคอลัมน์ทั้งหมดในแต่ละบรรทัด
search_file	ค้นหาเพิ่มข้อมูล
get_name	ใช้สำหรับข้อความที่ต้องการค้นหาหรือแทนที่
gotoend	ไปยังตำแหน่งท้ายสุดของเพิ่มข้อมูล
goto_xy	ไปยังตำแหน่งใดๆบนหน้าจอ
header_line	แสดงข้อความเกี่ยวกับบรรทัด คอลัมน์ และชื่อเพิ่มข้อมูล
help	แสดงข้อความช่วยเหลือที่บรรทัดสุดท้ายของจอภาพ
home	เลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นของเพิ่มข้อมูล
initial_values	กำหนดค่าเริ่มต้นให้กับตัวแปร

ตารางที่ 3.2 (ต่อ)

ชื่อโปรแกรมย่อย	หน้าที่การทำงาน
increase_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
insert_line	ทำการแทรกบรรทัดว่างต่อบรรทัดปัจจุบัน
kill_line	ทำการลบบรรทัดที่เคอร์เซอร์ปรากฏอยู่
left	เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง
line_number	ทำการเคลื่อนย้ายเคอร์เซอร์ไปยังบรรทัดที่กำหนด
load	ทำการอ่านเพิ่มข้อมูลจากหน่วยความจำสำรองไปเก็บไว้ในหน่วยความจำ
manage_option	ทำหน้าที่จัดการเงื่อนไขของการค้นหาหรือแทนที่
move_block	ทำการเคลื่อนย้ายข้อความในบล็อกไปยังตำแหน่งอื่น
newdown_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
os_shell	ออกจากโปรแกรมชั่วคราวเพื่อไปยังระบบปฏิบัติการ
pagedown	ไปยังหน้าจอถัดไป 1 หน้าจอ
pageup	ไปยังหน้าจอก่อนหน้า 1 หน้าจอ
print_process	จัดการเกี่ยวกับการเลือกลักษณะการพิมพ์
init_prn	ทำการ reset เครื่องพิมพ์
prtrdy	ตรวจสอบความเรียบร้อยของเครื่องพิมพ์
print	ทำการส่งกลุ่มของข้อความหรือเพิ่มข้อมูลไปยังเครื่องพิมพ์
printline	แสดงข้อความเฉพาะบรรทัดที่กำหนดบนจอภาพ
range	จัดลำดับของเงื่อนไขของการค้นหาหรือแทนที่
read_file	ทำการอ่านข้อมูลจากเพิ่มข้อมูลอื่นมายังเพิ่มข้อมูลที่กำลังใช้งาน
rename_file	ทำการเปลี่ยนชื่อเพิ่มข้อมูล

ตารางที่ 3.2 (ต่อ)

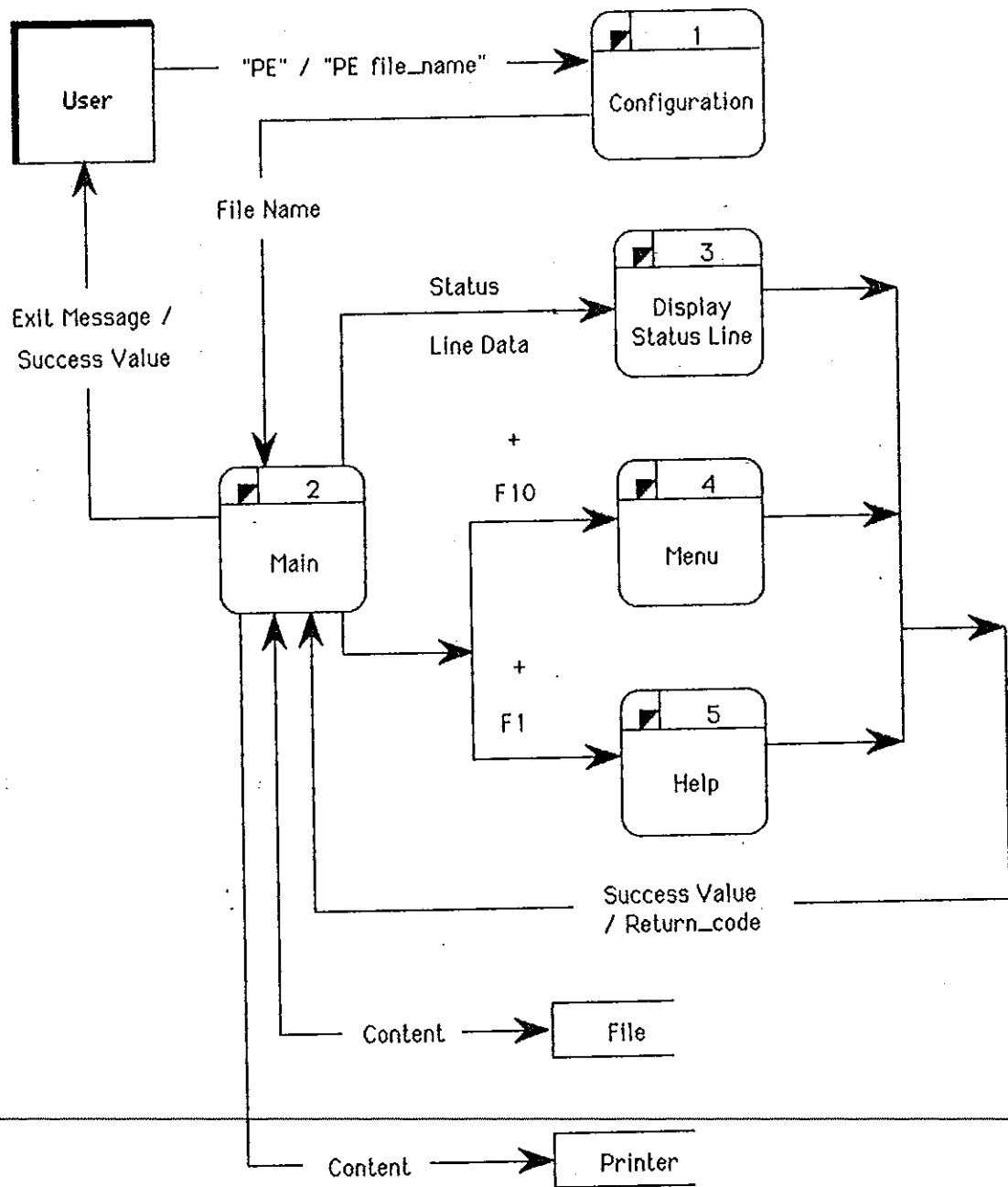
ชื่อโปรแกรมย่อย	หน้าที่การทำงาน
replace_process	ทำการแทนที่ข้อความเดิมด้วยข้อความที่ต้องการ
replace	จัดการเกี่ยวกับการแทนที่ข้อความ
return_process	จัดการจอภาพเมื่อตัวอักษรที่พิมพ์เป็น carriage return
right	เลื่อนเคอร์เซอร์ไปทางขวา 1 อักขระ
save	ทำการจัดเก็บเพิ่มข้อมูล
screen_xy	ค้นหาตำแหน่งเริ่มต้นของข้อความที่จะนำมาแสดงบนจอภาพ
scrolldn	เลื่อนจอภาพลง 1 จอภาพ
scrollup	เลื่อนจอภาพขึ้น 1 จอภาพ
set_dta	จัดเตรียม disk transfer area
search	ทำการค้นหาข้อความที่ต้องการ
show_help	จัดสร้าง help window
sort_option	ทำการจัดเรียงเงื่อนไขของการค้นหาหรือแทนที่
space_increasing	เป็นการเพิ่มตัวอักขระว่าง (blank character) จากท้ายสุดของบรรทัดถึงตำแหน่งที่เคอร์เซอร์อยู่
topscreen	เลื่อนเคอร์เซอร์ไปยังบรรทัดบนสุดของจอภาพ
transfer_char	ทำการส่งตัวอักขระที่เก็บอยู่ใน buffer ไปยัง buf
upline	เลื่อนเคอร์เซอร์ขึ้น 1 บรรทัด
upscreen	เลื่อนจอภาพขึ้น 1 บรรทัด
up_screen_ptr	กำหนดตำแหน่งเริ่มต้นของแต่ละหน้าจอ
word_uplow_strcmp	ทำการเปรียบเทียบข้อความ 2 ข้อความ
word_delete	ลบคำที่อยู่ทางขวา 1 คำ
word_left	เลื่อนเคอร์เซอร์ไปทางซ้าย 1 คำ
wordright	เลื่อนเคอร์เซอร์ไปทางขวา 1 คำ
write_char	แสดงตัวอักขระบนหน้าจอ
yank	ทำการเอาบรรทัดที่ถูกลบไปกลับคืน

ตารางที่ 3.2 (ต่อ)

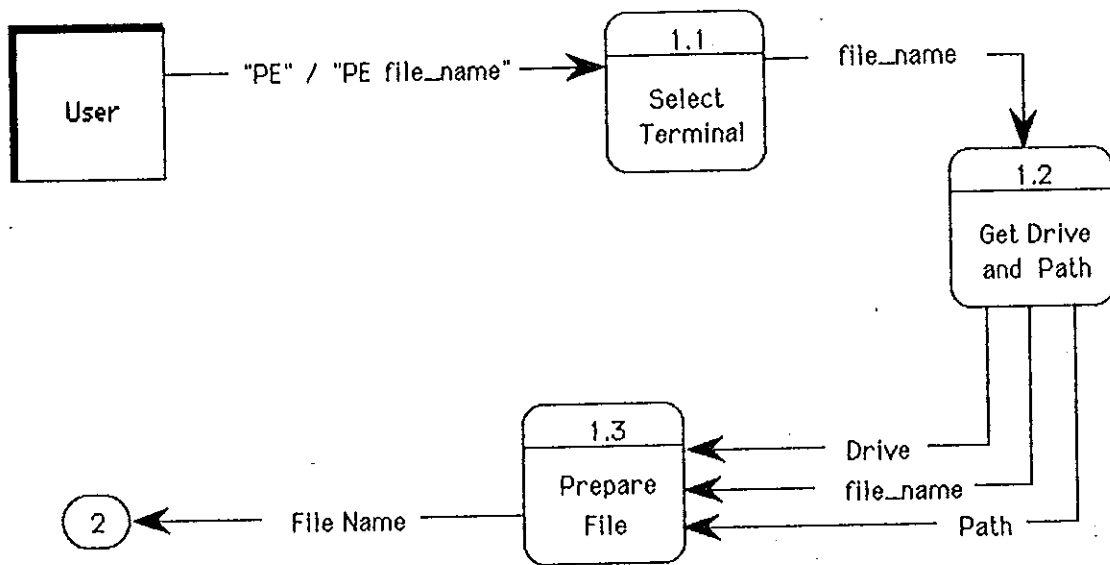
ชื่อโปรแกรมย่อย	หน้าที่การทำงาน
window_construct	ทำการสร้างหน้าต่าง
window_box	ทำการตีกรอบรอบหน้าต่าง
active_window_message	แสดงข้อความในหน้าต่าง
window_close	ทำการปิดหน้าต่างที่กำลังใช้งาน
window_open	ทำการเปิดหน้าต่าง
pulldown_menu	จัดสร้างและแสดง pulldown menu

3.6 กระบวนการดำเนินงาน

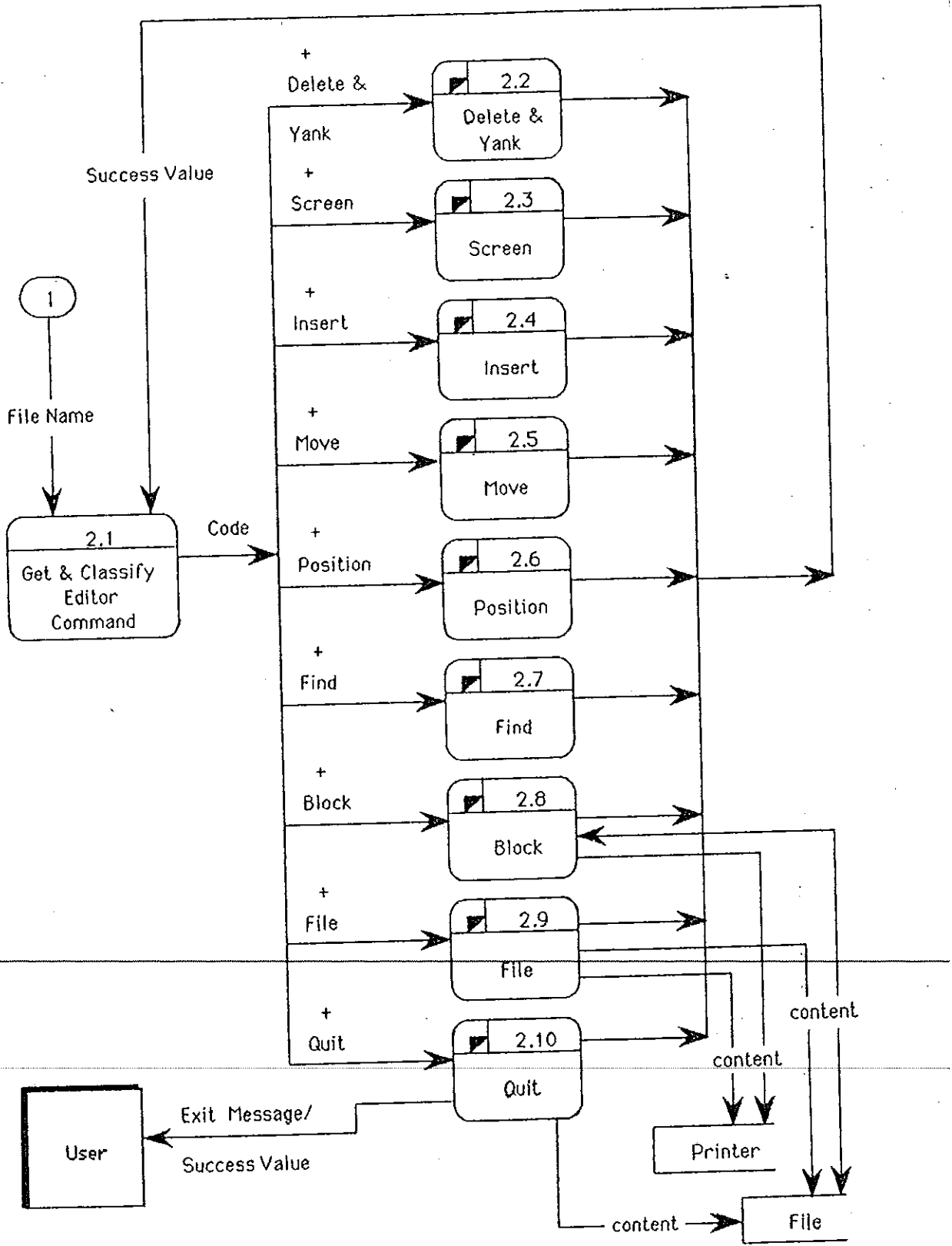
ผู้วิจัยได้เลือกแผนภาพกระแสข้อมูล (dataflow diagram) ในการแสดงกระบวนการดำเนินงานของเอดิเตอร์ PE ทั้งระบบ มีรายละเอียดดังแสดงในรูปที่ 3.8 ถึงรูปที่ 3.22



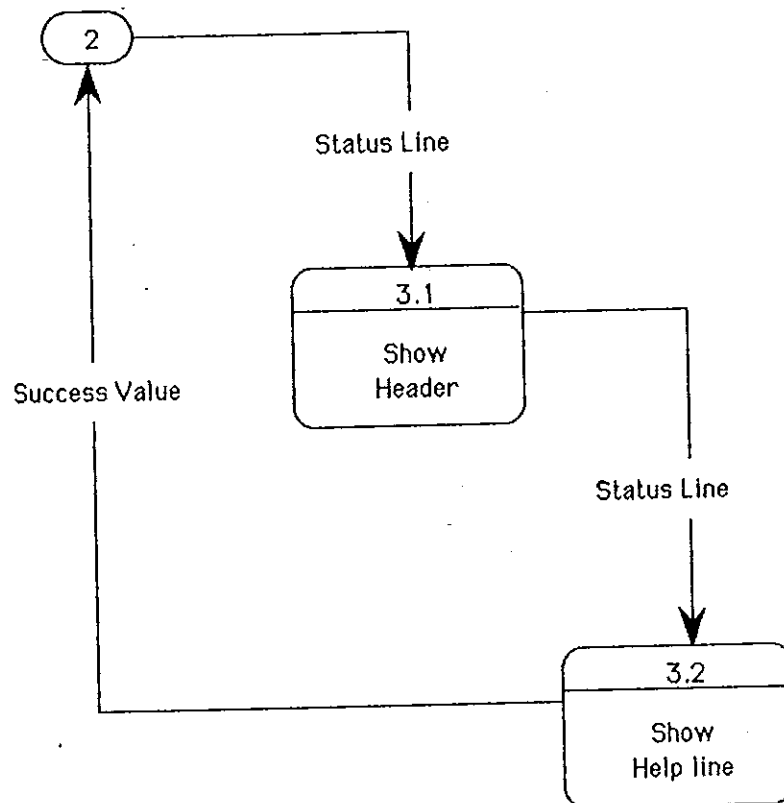
รูปที่ 3.8 กระบวนการดำเนินงานหลักของระบบ



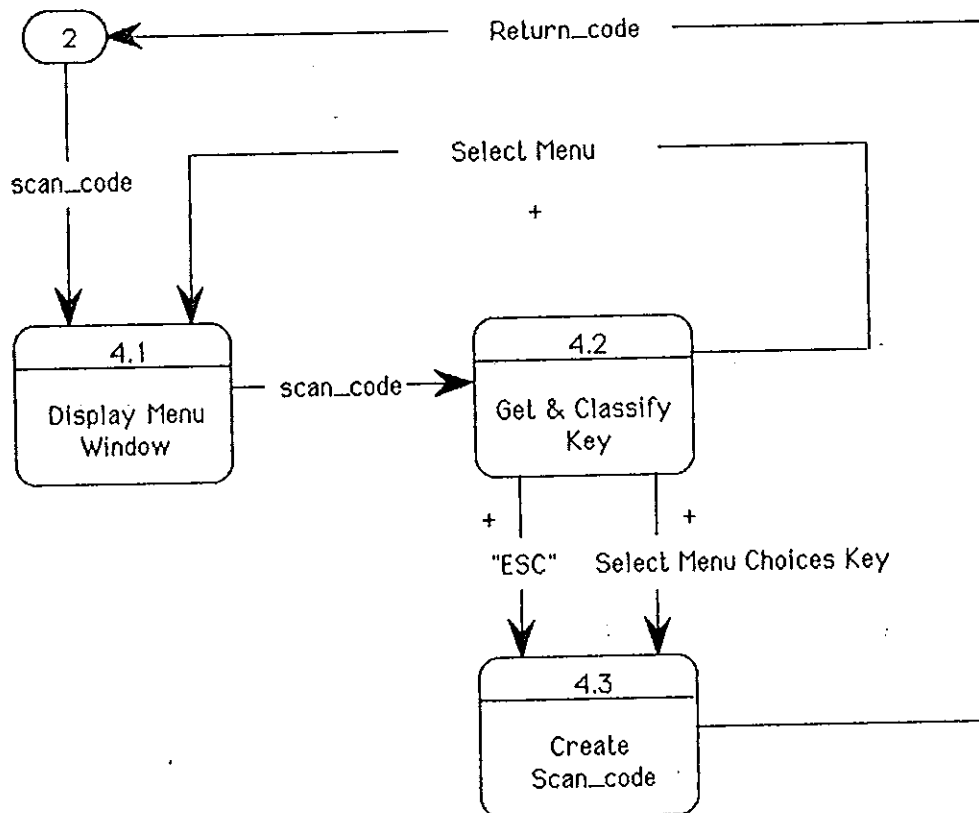
รูปที่ 3.9 กระบวนการ Configuration



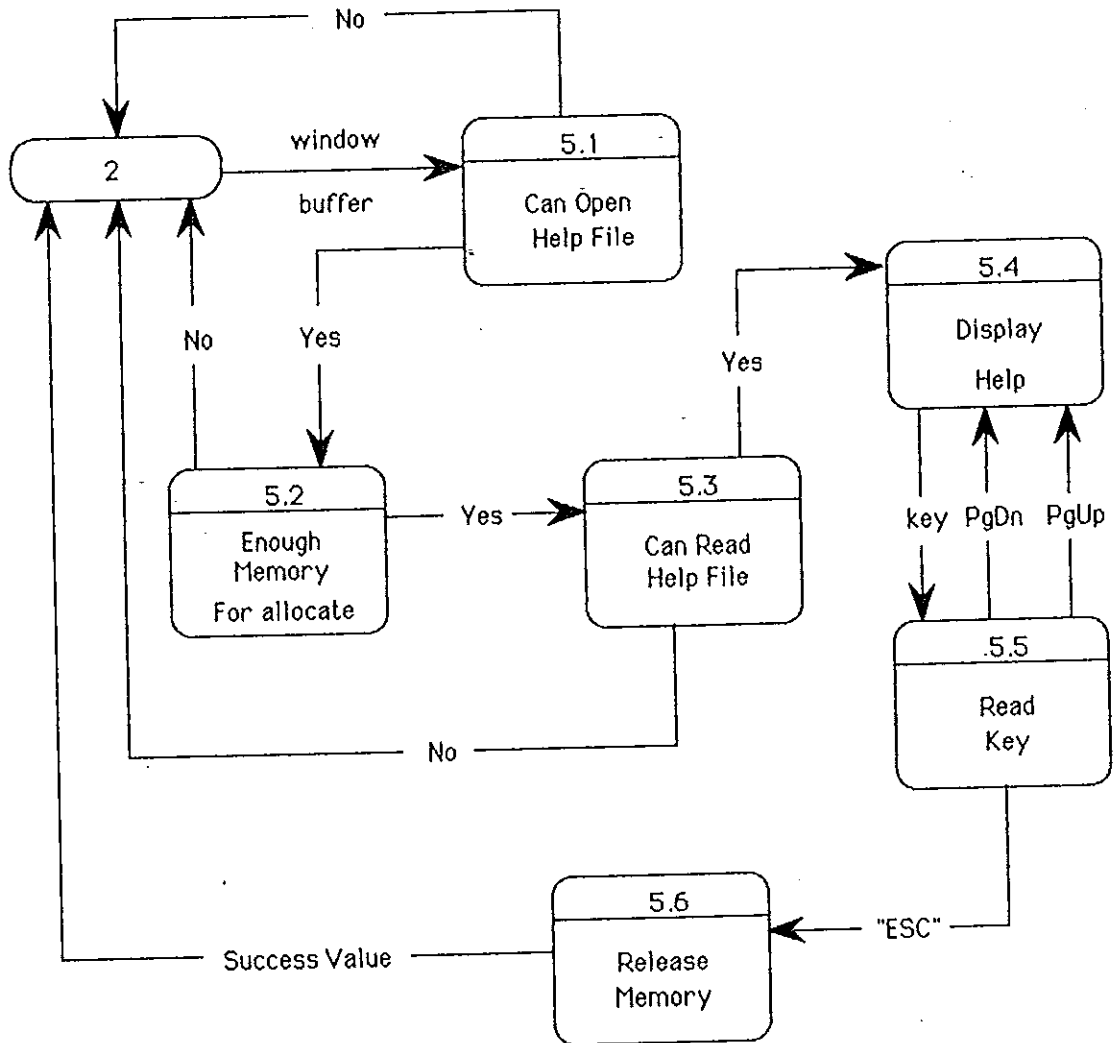
รูปที่ 3.10 กระบวนการ Main



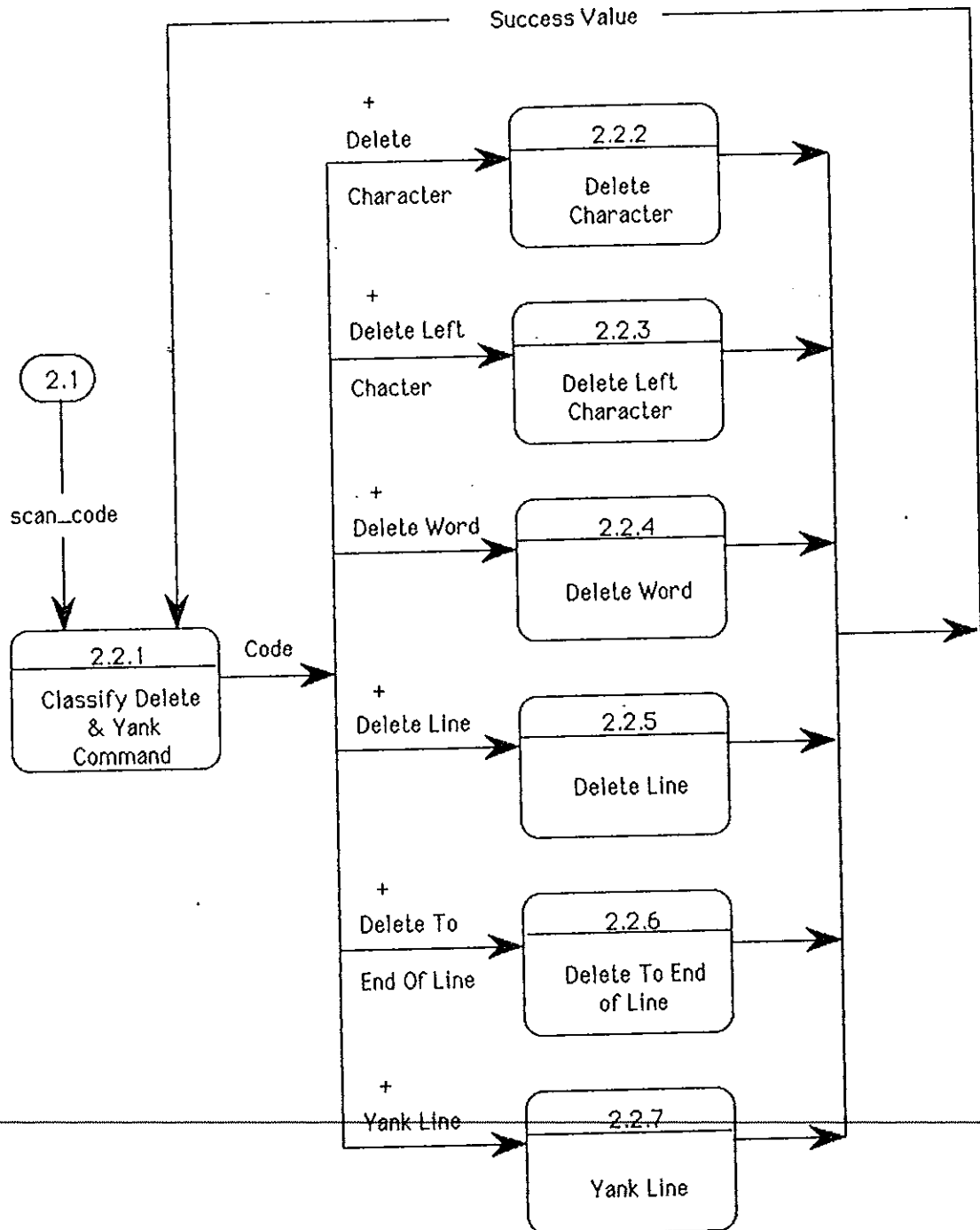
รูปที่ 3.11 กระบวนการ Display Status Line



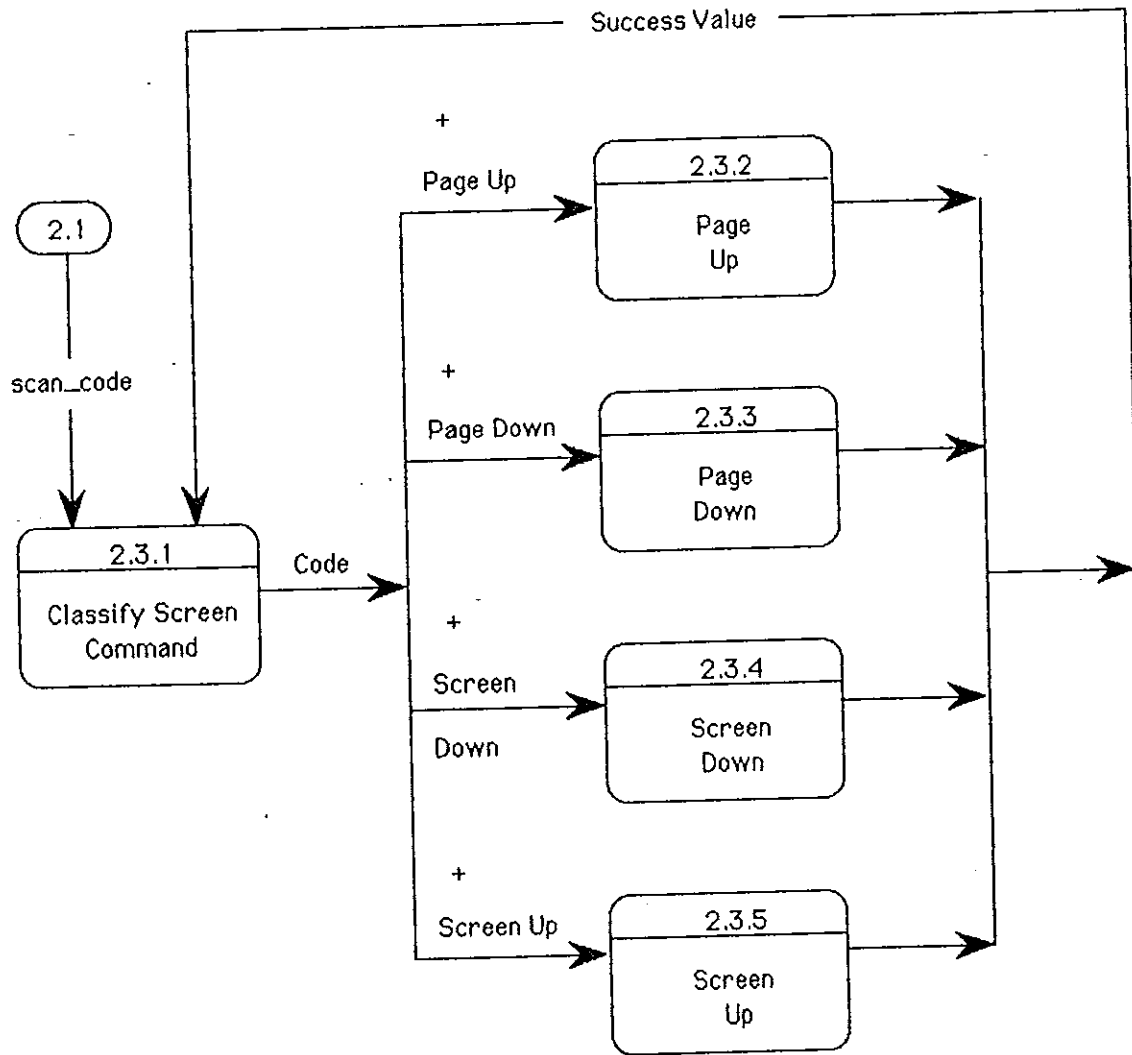
รูปที่ 3.12 กระบวนการ Menu



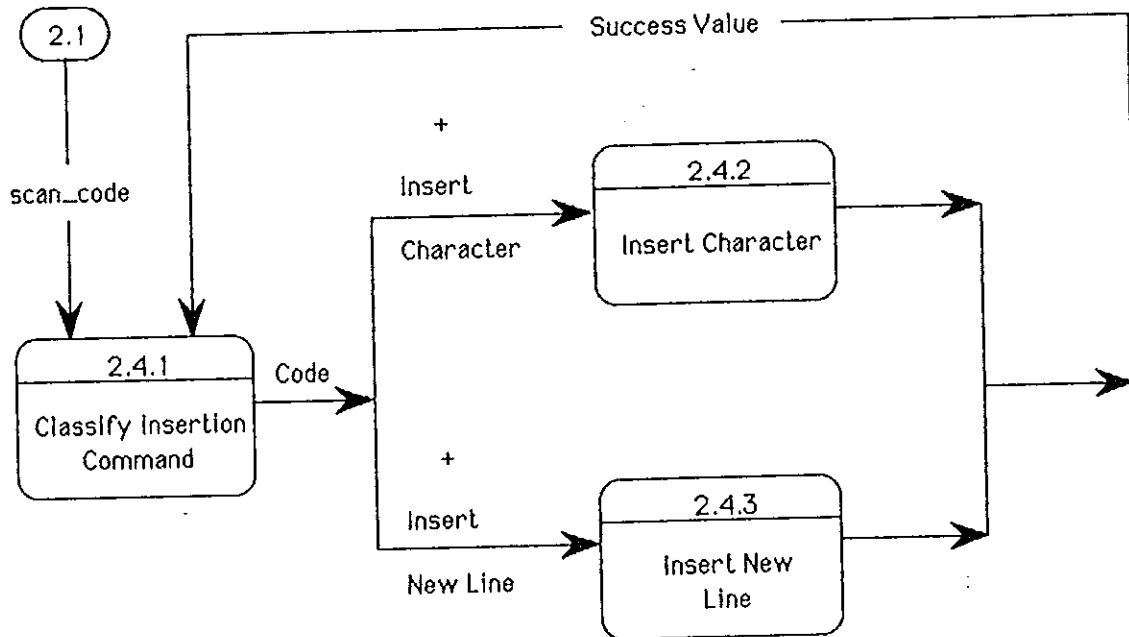
รูปที่ 3.13 กระบวนการ Help



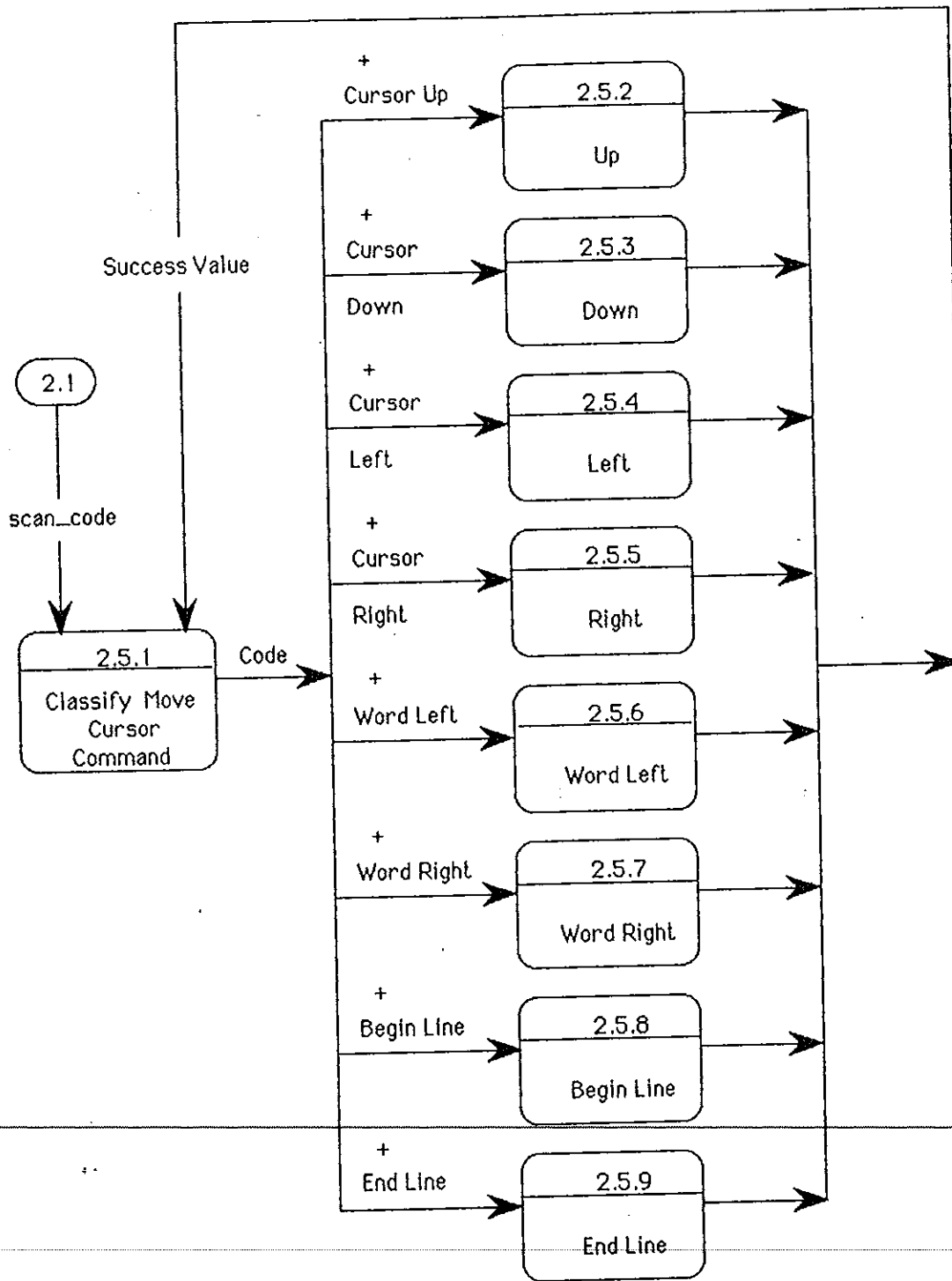
รูปที่ 3.14 กระบวนการ Delete & Yank



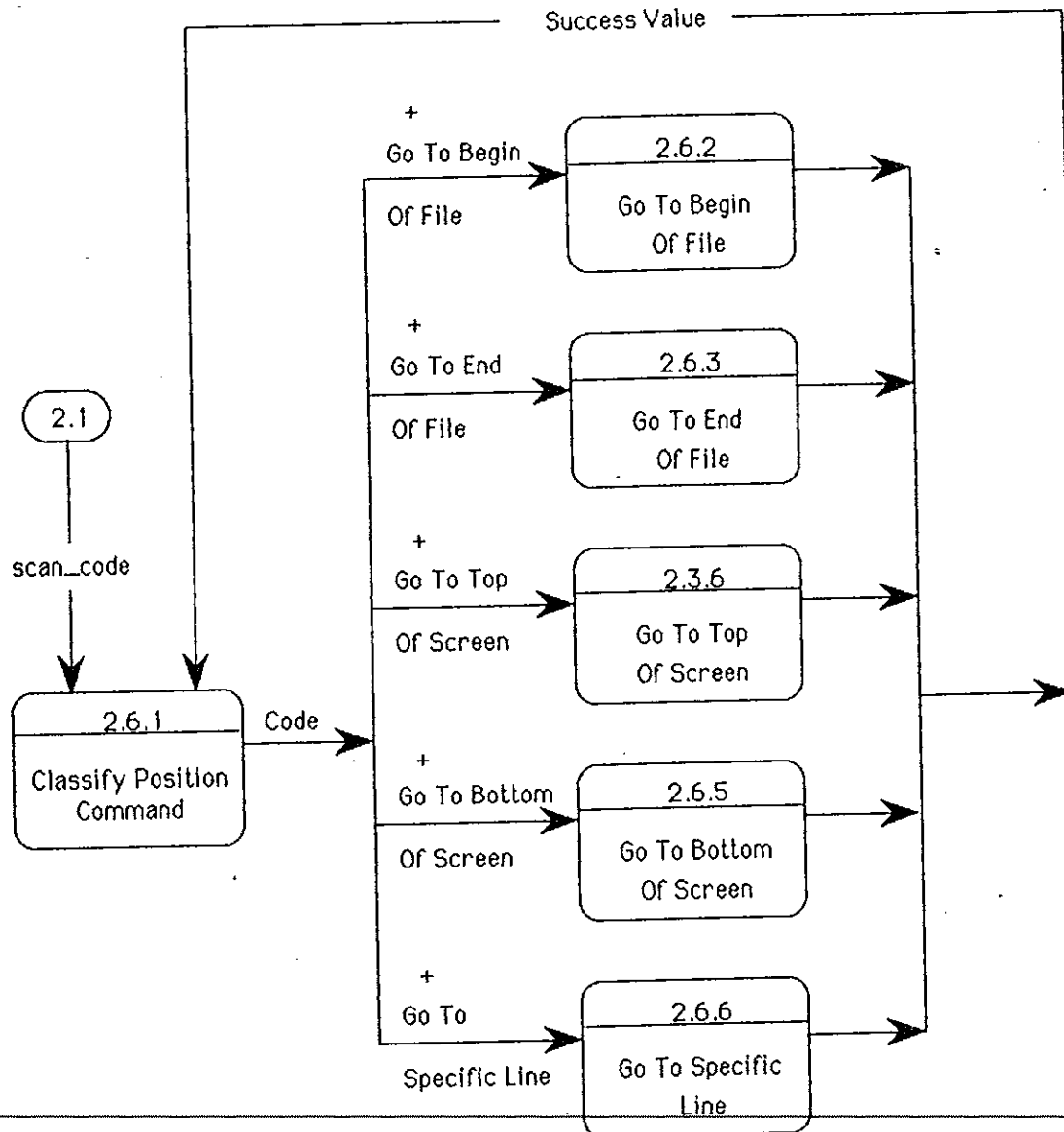
รูปที่ 3.15 กระบวนการ Screen



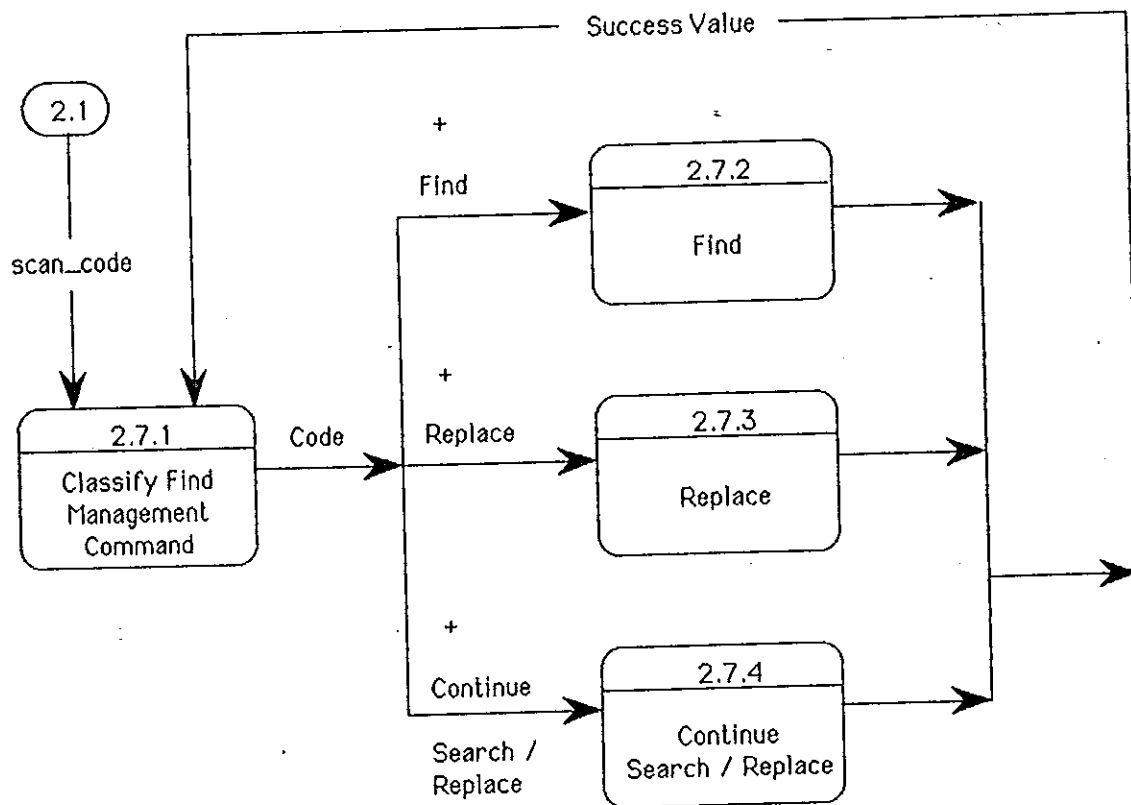
รูปที่ 3.16 กระบวนการ Insert



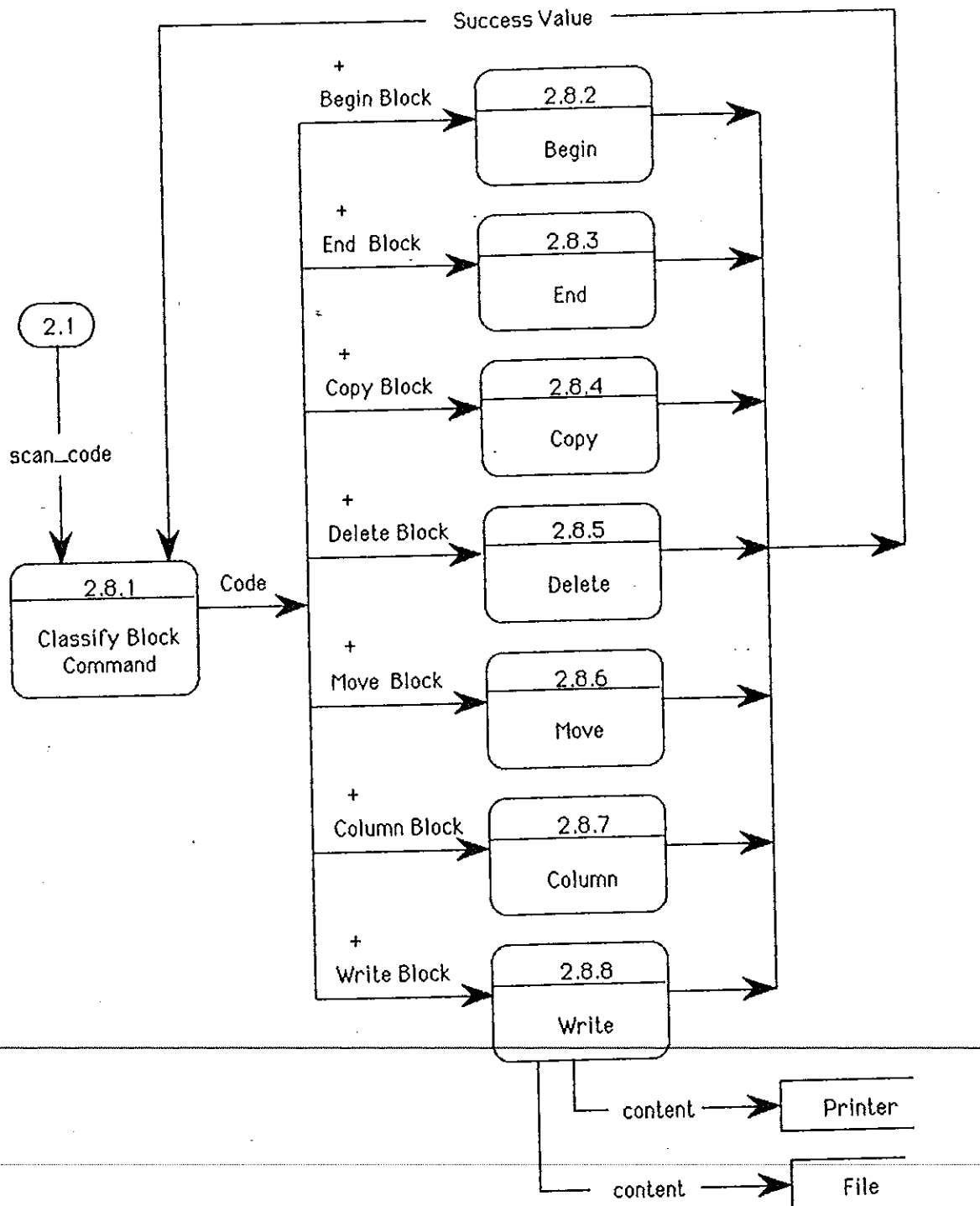
รูปที่ 3.17 กระบวนการ Move



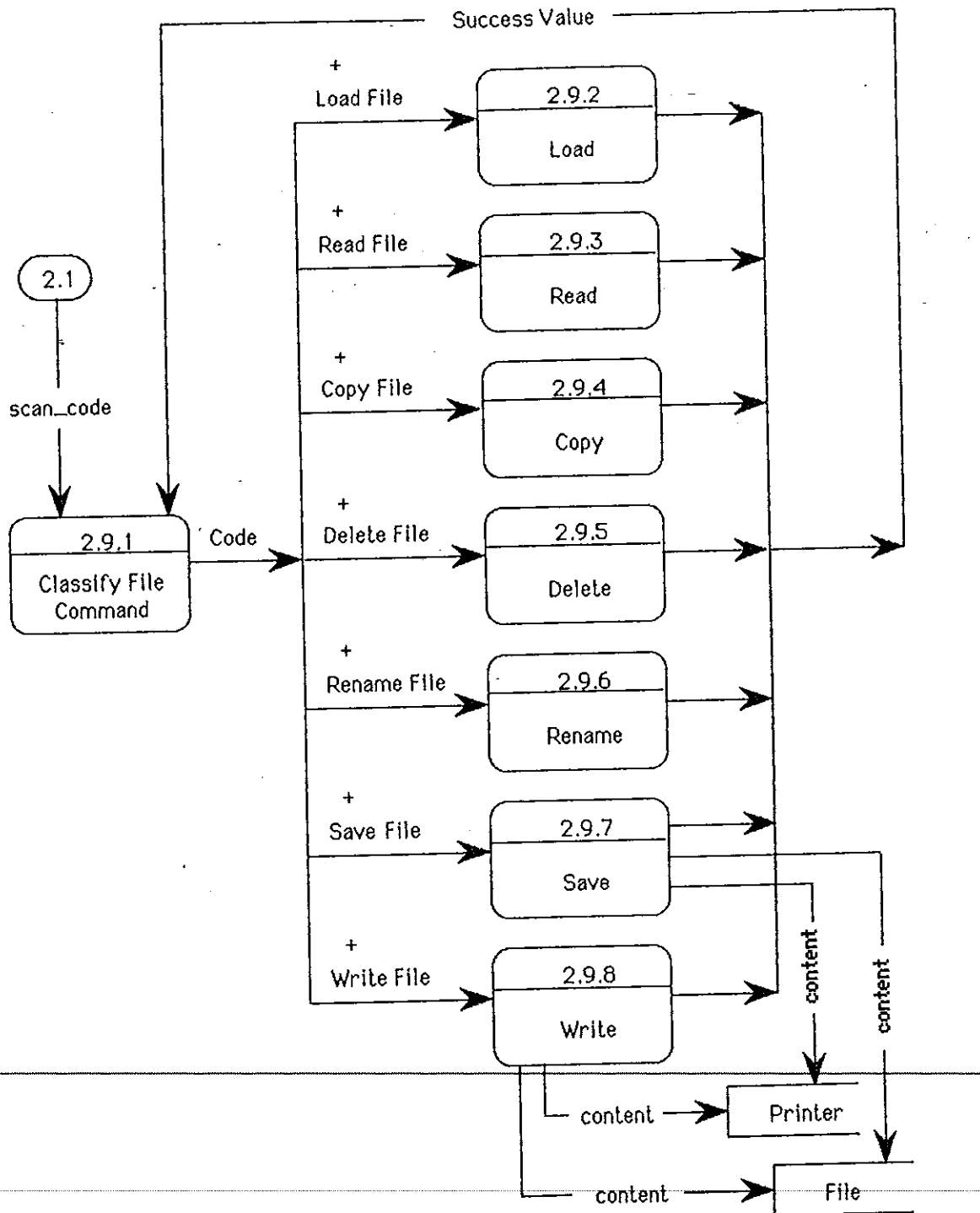
รูปที่ 3.18 กระบวนการ Position



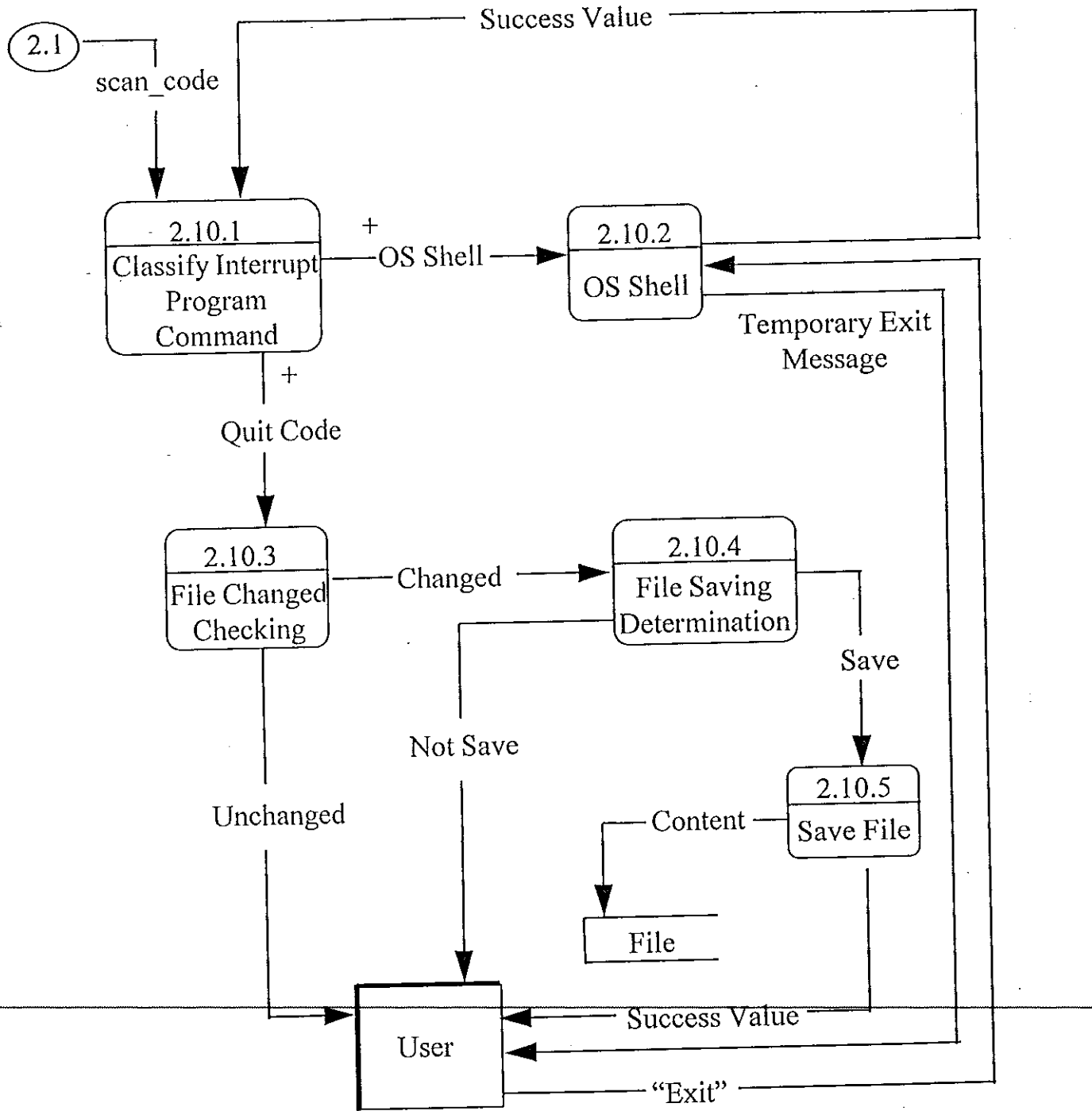
รูปที่ 3.19 กระบวนการ Find



รูปที่ 3.20 กระบวนการ Block



รูปที่ 3.21 กระบวนการ File Process



รูปที่ 3.22 กระบวนการ Quit

คำอธิบายโดยสังเขปสำหรับแต่ละกระบวนการของแผนภาพกระแสข้อมูลในรูปที่ 3.8 ถึง 3.22 มีดังนี้

รูปที่ 3.8

- กระบวนการที่ 1 เป็นการจัดเตรียมเพิ่มข้อมูล และจัดการเกี่ยวกับจอภาพให้พร้อมที่จะทำงาน
- กระบวนการที่ 2 เป็นการดำเนินงานต่างๆที่เกี่ยวกับการเรียกใช้คำสั่งในการใช้งานโปรแกรม
- กระบวนการที่ 3 เป็นการแสดงข้อมูลเฉพาะเพื่อการดำเนินงาน และแสดงคำสั่งในการใช้งานโปรแกรม
- กระบวนการที่ 4 เป็นการดำเนินงานเกี่ยวกับการจัดสร้างรายการเลือกแบบดิ่งลง และการเรียกใช้คำสั่งของโปรแกรมโดยผ่านทางรายการเลือก
- กระบวนการที่ 5 เป็นการดำเนินงานในส่วนของการแสดงความช่วยเหลือในการใช้งานโปรแกรม

รูปที่ 3.9

- กระบวนการที่ 1.1 เป็นการเลือกชนิดจอภาพให้เหมาะสมกับการทำงานของโปรแกรมอย่างอัตโนมัติ
- กระบวนการที่ 1.2 เป็นการตรวจสอบไดรฟ์ (drive) และเส้นทาง (path) ที่ผู้ใช้งานกำลังใช้งานโปรแกรม
- กระบวนการที่ 1.3 เป็นการจัดเตรียมเพิ่มข้อมูลที่จะนำไปใช้งาน

รูปที่ 3.10

- กระบวนการที่ 2.1 เป็นการรับและจำแนกคำสั่งการทำงานของโปรแกรม
- กระบวนการที่ 2.2 เป็นการดำเนินงานเกี่ยวกับการลบข้อความและนำข้อความที่ถูกลบทิ้งกลับคืนมาอีกครั้ง
- กระบวนการที่ 2.3 เป็นการดำเนินงานเกี่ยวกับการแสดงผลทางจอภาพ
- กระบวนการที่ 2.4 เป็นการดำเนินงานเกี่ยวกับการรับตัวอักษรทางแป้นพิมพ์ และการเพิ่ม/แทรกแถวว่าง
- กระบวนการที่ 2.5 เป็นการดำเนินงานเกี่ยวกับการเลื่อนตำแหน่งของเคอร์เซอร์ไปยังตำแหน่งต่างๆภายในแถว หรือ ทำการเลื่อนเคอร์เซอร์ไปยังแถวที่อยู่ก่อนหรือหลังจากแถวปัจจุบันครั้งละ 1 แถว
- กระบวนการที่ 2.6 เป็นการดำเนินงานเกี่ยวกับการเลื่อนตำแหน่งของเคอร์เซอร์ไปยังข้อความแถวบนสุดหรือข้อความแถวล่างสุดของจอภาพ หรือ ทำการเลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นหรือจุดสิ้นสุดของเพิ่มข้อมูล รวมทั้งเลื่อนเคอร์เซอร์ไปยังแถวใดๆที่ผู้ใช้ต้องการ

กระบวนการที่ 2.7 เป็นการดำเนินงานเกี่ยวกับการค้นหา หรือแทนที่ข้อความที่อยู่ภายในแฟ้ม

ข้อมูล

กระบวนการที่ 2.8 เป็นการดำเนินงานต่างๆเกี่ยวกับข้อความที่มีจำนวนมากเป็นชุด

กระบวนการที่ 2.9 เป็นการดำเนินงานต่างๆเกี่ยวกับแฟ้มข้อมูล

กระบวนการที่ 2.10 เป็นการดำเนินงานเกี่ยวกับการออกจากโปรแกรมชั่วคราว หรือสิ้นสุดการใช้
งาน

รูปที่ 3.11

กระบวนการที่ 3.1 เป็นการแสดงข้อมูลเฉพาะเพื่อการดำเนินงาน ซึ่งได้แก่ชื่อแฟ้มข้อมูลที่กำลังทำงานด้วย หมายเลขของแถวและสดมภ์ ณ ตำแหน่งปัจจุบันที่เคอร์เซอร์อยู่ รวมทั้งจำนวนทั้งหมดของแถวในแฟ้มข้อมูลและจำนวนทั้งหมดของสดมภ์ในแต่ละแถว

กระบวนการที่ 3.2 เป็นการแสดงคำสั่งในการใช้งานโปรแกรมที่สำคัญและใช้กันมาก

รูปที่ 3.12

กระบวนการที่ 4.1 เป็นการแสดงรายการเลือกแบบดิ่งลง

กระบวนการที่ 4.2 เป็นการรับและจำแนกคำสั่งการใช้งานในรายการเลือกแบบดิ่งลง

กระบวนการที่ 4.3 เป็นการจัดสร้างรหัสคำสั่งการใช้งานโปรแกรม

รูปที่ 3.13

กระบวนการที่ 5.1 เป็นการตรวจสอบว่า สามารถเปิดแฟ้มข้อมูลที่เก็บข้อความแสดงความช่วยเหลือได้หรือไม่

กระบวนการที่ 5.2 เป็นการตรวจสอบว่ามีหน่วยความจำเพียงพอให้สามารถบรรจุข้อความแสดงความช่วยเหลือหรือไม่

กระบวนการที่ 5.3 เป็นการตรวจสอบว่าสามารถอ่านข้อความจากแฟ้มข้อมูลที่เก็บข้อความแสดงความช่วยเหลือหรือไม่

กระบวนการที่ 5.4 เป็นการแสดงข้อความแสดงความช่วยเหลือบนจอภาพ

กระบวนการที่ 5.5 เป็นการอ่านอักขระจากแป้นพิมพ์

กระบวนการที่ 5.6 เป็นการคืนหน่วยความจำที่จองไว้ ให้กับหน่วยความจำหลัก

รูปที่ 3.14

- กระบวนการที่ 2.2.1 เป็นการจำแนกคำสั่งเกี่ยวกับการลบข้อความและการนำข้อความที่ถูกลบทิ้งกลับคืนมา
- กระบวนการที่ 2.2.2 เป็นการดำเนินงานเกี่ยวกับการลบข้อความครึ่งละ 1 อักขระ
- กระบวนการที่ 2.2.3 เป็นการดำเนินงานเกี่ยวกับการลบข้อความที่อยู่ทางซ้ายของเคอร์เซอร์ครึ่งละ 1 อักขระ
- กระบวนการที่ 2.2.4 เป็นการดำเนินงานเกี่ยวกับการลบข้อความครึ่งละ 1 คำ
- กระบวนการที่ 2.2.5 เป็นการดำเนินงานเกี่ยวกับการลบข้อความครึ่งละ 1 แถว
- กระบวนการที่ 2.2.6 เป็นการดำเนินงานเกี่ยวกับการลบข้อความจากตำแหน่งที่เคอร์เซอร์อยู่ไปจนกระทั่งสิ้นสุดบรรทัด
- กระบวนการที่ 2.2.7 เป็นการดำเนินงานเกี่ยวกับการนำเอาข้อความที่ถูกลบทิ้งไปกลับคืนมา

รูปที่ 3.15

- กระบวนการที่ 2.3.1 เป็นการจำแนกคำสั่งเกี่ยวกับการแสดงผลทางจอภาพ
- กระบวนการที่ 2.3.2 เป็นการดำเนินงานเกี่ยวกับการแสดงข้อความของหน้าจอที่แล้ว
- กระบวนการที่ 2.3.3 เป็นการดำเนินงานเกี่ยวกับการแสดงข้อความของหน้าจอถัดไป
- กระบวนการที่ 2.3.4 เป็นการดำเนินงานเกี่ยวกับการเลื่อนจอภาพลงครึ่งละ 1 แถว
- กระบวนการที่ 2.3.5 เป็นการดำเนินงานเกี่ยวกับการเลื่อนจอภาพขึ้นครึ่งละ 1 แถว

รูปที่ 3.16

- กระบวนการที่ 2.4.1 เป็นการจำแนกคำสั่งเกี่ยวกับการรับตัวอักขระทางแป้นพิมพ์ และการเพิ่มแทรกแถวว่าง
- กระบวนการที่ 2.4.2 เป็นการดำเนินงานเกี่ยวกับการรับตัวอักขระทางแป้นพิมพ์ครึ่งละ 1 อักขระ
- กระบวนการที่ 2.4.3 เป็นการดำเนินงานเกี่ยวกับการเพิ่มแทรกแถวว่างระหว่างแถวที่มีข้อความ

รูปที่ 3.17

- กระบวนการที่ 2.5.1 เป็นการจำแนกคำสั่งเกี่ยวกับการเลื่อนตำแหน่งของเคอร์เซอร์ไปยังตำแหน่งต่างๆภายในแถว หรือ ทำการเลื่อนเคอร์เซอร์ไปยังแถวที่อยู่ก่อน หรือ หลังจากแถวปัจจุบันครึ่งละ 1 แถว
- กระบวนการที่ 2.5.2 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังแถวที่อยู่ก่อนหน้าแถวปัจจุบัน

กระบวนการที่ 2.5.3 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังแถวที่อยู่ถัดจากแถวปัจจุบัน

กระบวนการที่ 2.5.4 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปทางซ้าย 1 อักขระ

กระบวนการที่ 2.5.5 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปทางขวา 1 อักขระ

กระบวนการที่ 2.5.6 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปทางซ้าย 1 คำ

กระบวนการที่ 2.5.7 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปทางขวา 1 คำ

กระบวนการที่ 2.5.8 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นของแถว

กระบวนการที่ 2.5.9 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปทางจุดสิ้นสุดของแถว

รูปที่ 3.18

กระบวนการที่ 2.6.1 เป็นการจำแนกคำสั่งเกี่ยวกับการเลื่อนตำแหน่งของเคอร์เซอร์ไปยังข้อความแถวบนสุดหรือข้อความแถวล่างสุดของจอภาพ หรือ ทำการเลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นหรือจุดสิ้นสุดของแฟ้มข้อมูล รวมทั้งเลื่อนเคอร์เซอร์ไปยังแถวใดๆที่ผู้ใช้ต้องการ

กระบวนการที่ 2.6.2 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นของแฟ้มข้อมูล

กระบวนการที่ 2.6.3 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังจุดสิ้นสุดของแฟ้มข้อมูล

กระบวนการที่ 2.6.4 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังข้อความแถวบนสุดของจอภาพ

กระบวนการที่ 2.6.5 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังข้อความแถวล่างสุดของจอภาพ

กระบวนการที่ 2.6.6 เป็นการดำเนินงานเกี่ยวกับการเลื่อนเคอร์เซอร์ไปยังแถวใดๆที่ผู้ใช้ต้องการ

รูปที่ 3.19

กระบวนการที่ 2.7.1 เป็นการจำแนกคำสั่งเกี่ยวกับการค้นหาหรือแทนที่ข้อความที่อยู่ภายในแฟ้มข้อมูล

กระบวนการที่ 2.7.2 เป็นการดำเนินงานเกี่ยวกับการค้นหาข้อความที่ผู้ใช้ต้องการ

กระบวนการที่ 2.7.3 เป็นการดำเนินงานเกี่ยวกับการแทนที่ข้อความเดิมด้วยข้อความใหม่

กระบวนการที่ 2.7.4 เป็นการดำเนินงานเกี่ยวกับการค้นหา หรือ แทนที่ข้อความเดิมอีกครั้ง

รูปที่ 3.20

- กระบวนการที่ 2.8.1 เป็นการจำแนกคำสั่งการดำเนินงานต่างๆเกี่ยวกับข้อความที่มีจำนวนมาก
- กระบวนการที่ 2.8.2 เป็นการกำหนดจุดเริ่มต้นของบล็อก
- กระบวนการที่ 2.8.3 เป็นการกำหนดจุดสิ้นสุดของบล็อก
- กระบวนการที่ 2.8.4 เป็นการดำเนินงานเกี่ยวกับการคัดลอกบล็อก ไปไว้ยังตำแหน่งอื่นภายใน
เพิ่มข้อมูล
- กระบวนการที่ 2.8.5 เป็นการดำเนินงานเกี่ยวกับการลบบล็อก
- กระบวนการที่ 2.8.6 เป็นการดำเนินงานเกี่ยวกับการเคลื่อนย้ายบล็อกไปไว้ยังตำแหน่งอื่นภายใน
เพิ่มข้อมูล
- กระบวนการที่ 2.8.7 เป็นการดำเนินงานเกี่ยวกับการจัดสร้างบล็อกชนิดสดมภ์
- กระบวนการที่ 2.8.8 เป็นการดำเนินงานเกี่ยวกับการคัดลอกบล็อกไปไว้ยังเพิ่มข้อมูลอื่นหรือไป
ยังเครื่องพิมพ์

รูปที่ 3.21

- กระบวนการที่ 2.9.1 เป็นการจำแนกคำสั่งการดำเนินงานต่างๆเกี่ยวกับเพิ่มข้อมูล
- กระบวนการที่ 2.9.2 เป็นการนำเอาเพิ่มข้อมูลใหม่ไปไว้ในหน่วยความจำหลักที่จองเอาไว้
- กระบวนการที่ 2.9.3 เป็นการอ่านข้อมูลจากเพิ่มข้อมูลหนึ่งมาแทรกยังตำแหน่งที่เคอร์เซอร์อยู่
- กระบวนการที่ 2.9.4 เป็นการคัดลอกข้อมูลจากเพิ่มข้อมูลหนึ่ง ไปเก็บไว้ยังเพิ่มข้อมูลใหม่
- กระบวนการที่ 2.9.5 เป็นการดำเนินงานเกี่ยวกับการลบเพิ่มข้อมูล
- กระบวนการที่ 2.9.6 เป็นการดำเนินงานเกี่ยวกับเปลี่ยนชื่อเพิ่มข้อมูล
- กระบวนการที่ 2.9.7 เป็นการจัดเก็บข้อความที่ผู้ใช้จัดพิมพ์
- กระบวนการที่ 2.9.8 เป็นการดำเนินงานเกี่ยวกับการคัดลอกข้อมูลทั้งหมดของเพิ่มข้อมูล ไปไว้
ยังเพิ่มข้อมูลอื่น หรือไปยังเครื่องพิมพ์

รูปที่ 3.22

- กระบวนการที่ 2.10.1 เป็นการจำแนกคำสั่งเกี่ยวกับการออกจากโปรแกรมชั่วคราว หรือสิ้นสุดการใช้งาน
- กระบวนการที่ 2.10.2 เป็นการดำเนินงานเกี่ยวกับการออกจากโปรแกรมชั่วคราว
- กระบวนการที่ 2.10.3 เป็นการตรวจสอบว่ามีการแก้ไขเพิ่มข้อมูลที่กำลังใช้งานอยู่หรือไม่
- กระบวนการที่ 2.10.4 เป็นการสอบถามผู้ใช้ว่าต้องการจัดเก็บเพิ่มข้อมูลที่ได้รับการแก้ไขหรือไม่
- กระบวนการที่ 2.10.5 เป็นการจัดเก็บข้อมูลไปไว้ยังเพิ่มข้อมูล

บทที่ 4

สรุป

วิทยานิพนธ์นี้ เป็นการออกแบบและพัฒนาโปรแกรมเอดิเตอร์แบบเต็มจอภาพที่สามารถนำไปใช้ในการจัดสร้างโปรแกรมต้นฉบับ รวมทั้งใช้ในการจัดพิมพ์และแก้ไขเอกสารแบบง่าย ๆ ได้ โปรแกรมเอดิเตอร์ที่พัฒนาขึ้นนี้สามารถใช้งานได้ทั้งภาษาไทยและภาษาอังกฤษ และนอกเหนือจากคำสั่งที่ใช้ในการจัดการกับข้อความแล้ว โปรแกรมที่พัฒนายังได้เพิ่มคำสั่งในการจัดการกับเพิ่มข้อมูลที่เก็บอยู่ในหน่วยความจำสำรอง เช่น การทำสำเนาเพิ่มข้อมูล การลบเพิ่มข้อมูล และการเปลี่ยนชื่อเพิ่มข้อมูลนอกจากนี้ยังใช้เปลี่ยนชื่อสารบบย่อย (sub directory) ได้อีกด้วย

4.1 ผลการวิจัย

หลังจากใช้เวลาประมาณ 3 เดือน ในการศึกษาเอดิเตอร์แบบต่างๆที่มีให้อยู่บนเครื่องไมโครคอมพิวเตอร์ และจากนั้นได้ออกแบบและพัฒนาโปรแกรมเอดิเตอร์แบบเต็มจอภาพ PE ซึ่งได้รวมเอาลักษณะเด่นๆที่ผู้วิจัยชื่นชอบจากเอดิเตอร์ต่างๆที่ได้ศึกษาและทดลองใช้มา โดยอาจสรุปคุณลักษณะของโปรแกรมเอดิเตอร์ PE ได้ดังนี้

1. ลักษณะการใช้งาน เป็นแบบหน้าต่างประกอบด้วยรายการเลือกหลักในรูปรายการเลือกแบบดิ่งลง แต่ละรายการในรายการเลือกหลัก จะประกอบด้วยรายการย่อยให้ผู้เลือกใช้ได้ด้วยการใช้ปุ่มสัญลักษณ์ลูกศรบนแป้นพิมพ์ซึ่งไปยังรายการที่ต้องการ ซึ่งลักษณะการใช้งานนี้อำนวยความสะดวกให้กับผู้ใช้ให้ใช้โปรแกรมได้ง่ายมาก

2. เอดิเตอร์ PE สามารถทำงานทุกอย่างที่เอดิเตอร์ทั่วไปพึงทำได้ โดยปุ่มแป้นพิมพ์แทนคำสั่งที่ให้ผู้เลือกใช้จะเหมือนกับของเอดิเตอร์ที่นิยมใช้ทั่วไป เพื่อผู้ใช้จะได้ไม่ต้องเรียนรู้ใหม่ ทำให้ใช้งานได้ง่าย

3. โปรแกรม Thai driver ที่สามารถใช้ร่วมกับเอดิเตอร์ PE ได้แก่ VTHAI.COM ไทยทิพย์ และภาษาไทยของบริษัท International Research Corporation LTD. (IRC) ซึ่งเป็นที่นิยมใช้กันมากโดยทั่วไป

4.2 ปัญหาและอุปสรรค

ปัญหาส่วนใหญ่ที่เกิดขึ้นในการจัดทำวิทยานิพนธ์ จะเป็นปัญหาเกี่ยวกับซอฟต์แวร์ Thai driver ที่นำมาใช้ร่วมกับโปรแกรมที่พัฒนา ทั้งนี้เนื่องจากรหัสภาษาไทยที่มีใช้ในคอมพิวเตอร์ยังไม่มีความมาตรฐานที่แน่นอน อีกทั้งยังสร้างปัญหาบางประการให้กับการทำงานของเอดิเตอร์ PE นอก

จากนี้เนื่องจากภาษาไทยมีความแตกต่างกับภาษาอังกฤษบางประการในเรื่องของการจัดระดับคำ ซึ่งภาษาอังกฤษมีเพียงระดับเดียวในขณะที่ภาษาไทยมีอยู่ 4 ระดับ ดังนั้นในการนำเอาภาษาไทยมาใช้ในคอมพิวเตอร์โดยเฉพาะในด้านเกี่ยวกับการจัดพิมพ์ หรือ แก๊ไขเอกสาร จะต้องคำนึงถึงเรื่องการจัดระดับด้วย ซึ่งขึ้นอยู่กับความสามารถของ Thai driver ที่เลือกใช้ ทำให้เสียเวลาพอสมควรในการเลือกและทดลองใช้

4.3 ข้อเสนอแนะ

โปรแกรมที่พัฒนาขึ้นนี้สามารถนำไปใช้งานร่วมกับโปรแกรมระบบอื่นที่อนุญาตให้ผู้ใช้สามารถเลือกใช้เอดิเตอร์ที่ต้องการได้ เช่น dBASE นอกจากนี้ยังสามารถใช้เป็นพื้นฐานในการพัฒนาโปรแกรมที่มีเอดิเตอร์เป็นส่วนประกอบด้วย เช่น โปรแกรมประมวลผลคำ หรือ โปรแกรมตรวจสอบความถูกต้องของคำ เป็นต้น เนื่องจากโปรแกรมที่ผู้วิจัยพัฒนาขึ้นนี้ได้ครอบคลุมความสามารถพื้นฐานของเอดิเตอร์แบบเต็มจอภาพ และได้เพิ่มความสามารถในการจัดการกับแฟ้มข้อมูลด้วย ซึ่งจะเป็นการอำนวยความสะดวกให้กับผู้ที่ต้องการนำโปรแกรมดังกล่าวไปใช้งาน

เอดิเตอร์ PE ที่พัฒนาขึ้นนี้ สามารถจัดการกับแฟ้มข้อมูลครั้งละ 1 แฟ้มข้อมูล ดังนั้นการที่ทำให้เอดิเตอร์ PE สามารถจัดการกับแฟ้มข้อมูลได้มากกว่า 1 แฟ้มข้อมูล จึงเป็นแนวทางหนึ่งที่ทำให้ผู้ที่สนใจในโปรแกรมที่ผู้วิจัยทำการพัฒนาขึ้นนี้ สามารถใช้เป็นแนวทางในการพัฒนาต่อไป เพื่อให้มีประสิทธิภาพมากขึ้น

บรรณานุกรม

ทวีศักดิ์ กอนันต์กุล และ คณะทำงานร่างข้อกำหนดร่วมเพื่อการเขียนโปรแกรมซึ่งแสดงผลเป็นภาษาไทย, (2534), คอมพิวเตอร์กับภาษาไทย : การพัฒนามาตรฐานเบื้องต้นสำหรับเทคโนโลยีสารสนเทศของไทย, ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) กระทรวงวิทยาศาสตร์เทคโนโลยีและการพลังงาน, กรุงเทพฯ

บุญเลิศ เลี่ยมทัศนาศ, ยืน ภู่วรรณ และ สมนึก ศิริโต, (2529), โปรแกรมคอมพิวเตอร์ ภาษาซี, ซีเอ็ดดูเคชั่น จำกัด, กรุงเทพฯ

Biggerstaff, Ted J. (1986), System Software Tools, Prentice-Hall, A Division of Simon & Schuster, Inc., Englewood Cliffs, New Jersey.

Cox, Kevin and Walker, David (1993), USER-INTERFACE DESIGN, Simon & Schuster (Asia) Pte Ltd.

David, Gordon Bitter (1985), Management Information Systems, McGraw-Hill, Inc.

Dettmann, Terry (1989), DOS Programmer's Reference 2nd Edition, Que Corporation.

FitzGerald, Jerry (1973), Fundamentals of Systems Analysis, John Wiley & Sons, Inc.

Grauer, Robert T. and Sugrue, Paul K. (1987), Microcomputer Applications, McGraw-Hill, Inc.

Holzner, Steven (1989), C With Assembly Language, BRADY, Simon & Schuster, Inc., New York.

- Horspool, R. Nigel (1986), C Programming in the Berkeley UNIX Environment,
Prentice-Hall Canada Inc., Scarborough, Ontario.
- Hutchison, Robert C. and Just, Steven B. (1986), Programming Using The C
Language, McGraw-Hill, Inc.
- Jamsa, Kris (1987), DOS: The Complete Reference, McGraw-Hill, Inc.
- Kelley, All (1987), C By Dissection, The Benjamin/Cummings Publishing Company, Inc.
- Kernighan, Brian W. and Ritchie, Dennis M. (1988), The C Programming
Language, 2nd Ed., Prentice-Hall, Inc., Englewood Cliffs, N.J., U.S.A.
- Knecht, Ken (1989), Advanced Turbo C., Scott, Foresman and Company.
- Kruse, Robert L. (1987), Data Structures and Program Design, Prentice-Hall, Inc., A
Division of Simon & Schuster Englewood Cliffs, New Jersey.
- Lientz, Bennet P. and Rea, Kathryn P. (1987), Data Communications for Business,
Times Mirror/Mosby Colledge Publishing, St. Louis.
- Lucas, Henry C. (1985), The Analysis, Design, and Implementation of Information
System, McGraw-Hill.
-
- Mayhew, Deborah J. (1992), Principles and Guidelines in Software User Interface
Design, A Simon & Schuster Company, Prentice-Hall, Inc., Englewood Cliffs,
New Jersey.
- Pressman, Roger S. (1987), Software Engineering, McGraw-Hill, Inc.

Schildt, Herbert (1989), Born to Code in C, Osborne McGraw-Hill, Berkeley, California, U.S.A.

Schildt, Herbert (1989), C: Power User's Guide, Osborne McGraw-Hill, Berkeley, California, U.S.A.

Schildt, Herbert (1988), Using the Models 50 and 60, McGraw-Hill, Inc.

Silver, Gerald A. and Silver, Joan B. (1976), Introduction to System Analysis, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Stevens, Al (1989), Turbo C: Memory-Resident Utilities, Screen I/O and Programming Techniques, Management Information Source Inc.

Waterhouse, Shirley A. (1979), Word Processing Fundamentals, Harper & Row Publishers, Inc., New York.

Wright, Graham and Evans, David (1974), Commercial Computer Programming, NCC Publication.

ภาคผนวก

เอกสารสำหรับผู้ใช้

เอ็ดิเตอร์ PE ถูกออกแบบการทำงาน ให้การติดต่อระหว่างผู้ใช้และ เอ็ดิเตอร์ PE อยู่ในลักษณะที่ให้ผู้เลือกใช้คำสั่งจากรายการเลือก และจากการกดปุ่มแป้นพิมพ์ต่างๆบนแป้นพิมพ์แทนคำสั่ง ในเอกสารนี้จะอธิบายถึงวิธีการติดตั้งโปรแกรมเอ็ดิเตอร์ PE ขั้นตอนการใช้งาน ลักษณะการแสดงผลทางจอภาพ ปุ่มแป้นพิมพ์แทนคำสั่งการดำเนินงาน รูปแบบและความหมายของคำสั่งที่เป็นรายการย่อยของรายการต่างๆบนรายการหลัก

1. การติดตั้งเอ็ดิเตอร์ PE

โปรแกรมทั้งหมดของ PE ประกอบด้วยโปรแกรมต้นฉบับซึ่งเขียนด้วยภาษา C และโปรแกรมภาษาเครื่องทั้งหมด ถูกบรรจุอยู่บนแผ่นจานแม่เหล็กชนิดความหนาแน่นสูง (high density) ทั้งขนาด 5.25 นิ้ว หรือขนาด 3.25 นิ้ว จำนวน 1 แผ่น

การติดตั้งโปรแกรมภาษาเครื่องเพื่อใช้งานทำได้ค่อนข้างสะดวก เนื่องจากการทำงานของเอ็ดิเตอร์ PE ไม่จำเป็นต้องอาศัยโปรแกรมหรือแฟ้มข้อมูลอื่น สามารถใช้งานได้ทั้งใน ฟลอปปีดิสก์ และฮาร์ดดิสก์ เพียงแต่ผู้ใช้ใช้คำสั่ง "copy" ของระบบปฏิบัติการ DOS เพื่อทำการสำเนาเอ็ดิเตอร์ PE จากแผ่นฟลอปปีดิสก์ต้นฉบับไปยังฟลอปปีดิสก์หรือฮาร์ดดิสก์ที่ผู้ใช้ต้องการติดตั้งโปรแกรม สิ่งที่ต้องคำนึงในการติดตั้งเอ็ดิเตอร์ PE ก็คือเนื้อที่ที่มีอยู่ในฟลอปปีดิสก์ หรือฮาร์ดดิสก์ จะต้องมีอย่างน้อย 153K และจะต้องมีโปรแกรม Thai driver ที่สามารถใช้ร่วมกับเอ็ดิเตอร์ PE ได้แก่ VTHAI.COM ไทยทิพย์ หรือภาษาไทยของบริษัท International Research Corporation LTD. (IRC)

2. ขั้นตอนการใช้งาน

ก่อนการใช้งานเอ็ดิเตอร์ PE ถ้าต้องการใช้ภาษาไทยด้วย ให้เลือกใช้โปรแกรม Thai driver ดังนี้

```
C:\>VTHAI <Enter>
```

จากนั้นเรียกใช้เอ็ดิเตอร์ PE จากสารบบแฟ้มข้อมูลที่มีโปรแกรมภาษาเครื่อง "PE.EXE" การเรียกใช้เอ็ดิเตอร์ PE ทำได้ 2 รูปแบบดังนี้

รูปแบบที่ 1 C:\PE filename<Enter>

หน้าจอที่ได้หลังจากการเรียกใช้โปรแกรมเอ็ดิเตอร์ PE จะเป็นหน้าจอในรูปแบบที่ ก.1 โดยที่ถ้าชื่อแฟ้มข้อมูลที่เรียกใช้เป็นแฟ้มใหม่ ก็จะได้หน้าจอส่วนแสดงผลว่างเปล่า แต่ถ้าเป็นแฟ้มข้อมูลเก่า ก็จะมีข้อความของแฟ้มข้อมูลปรากฏตั้งแต่ต้นแฟ้มดังตัวอย่างในรูปแบบที่ ก.2

รูปแบบที่ 2 C:\>PE <Enter>

รูปแบบนี้ ผู้ใช้ไม่ได้บ่งบอกเพิ่มข้อมูลที่ใช้ จะทำให้ได้หน้าจอในรูปที่ ก.1 ซึ่งมีส่วนแสดงผลว่างเปล่าเช่นเดียวกันกับการใช้งานในรูปแบบที่ 1 ในกรณีที่เพิ่มข้อมูลเป็นแฟ้มใหม่ แต่สิ่งที่ต่างก็คือเอดิเตอร์ PE จะกำหนดชื่อเพิ่มข้อมูลใหม่ที่ใช้งานให้เป็น "NONAME" โดยอัตโนมัติ

3 ลักษณะการแสดงผลทางจอภาพ

จอภาพที่ปรากฏให้เห็นในลักษณะดังรูปที่ ก.1 และรูปที่ ก.2 จะเป็นจอภาพหลักที่ปรากฏขึ้นในขณะที่ผู้ใช้เรียกใช้หรือกำลังใช้งานเอดิเตอร์ PE ความหมายของข้อความบนส่วนต่างๆของหน้าจอมีดังนี้

ส่วนที่ 1 เป็นการแสดงข้อมูลเฉพาะของแฟ้มข้อมูลที่กำลังถูกใช้งาน ประกอบด้วยรายละเอียดดังนี้

Line	เป็นข้อมูลที่บอกให้ผู้ใช้ทราบว่าในขณะนั้นผู้ใช้กำลังจัดการกับข้อความที่ อยู่ ณ แถวเลขที่เท่าไรของจอภาพ
Col	เป็นข้อมูลที่บอกให้ผู้ใช้ทราบว่าในขณะนั้นผู้ใช้กำลังจัดการกับข้อความที่เริ่มต้น ณ สดมภ์เลขที่เท่าไรของจอภาพ
Ins	เป็นข้อมูลที่บอกให้ผู้ใช้ทราบว่าในการจัดพิมพ์ข้อความครั้งต่อไปนั้น จะเป็นการจัดพิมพ์ข้อความในลักษณะการพิมพ์แทรก (จะปรากฏสัญลักษณ์ "Ins" ณ แถวบนสุดของจอภาพ) หรือการพิมพ์ทับข้อความเดิม (ไม่ปรากฏสัญลักษณ์ "Ins")

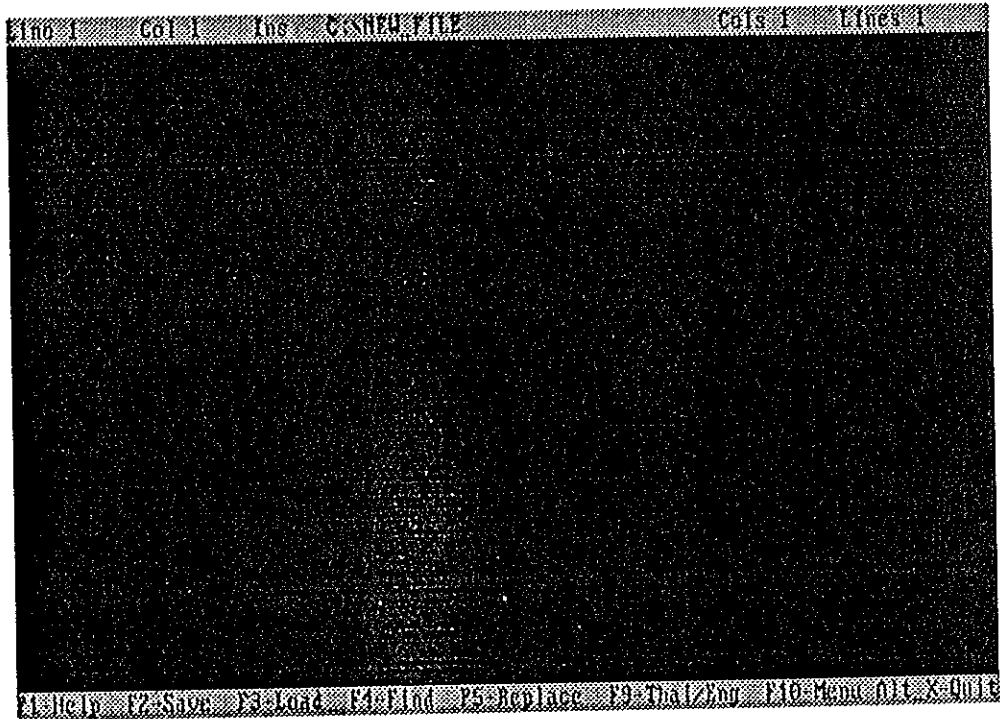
ชื่อเพิ่มข้อมูล เป็นข้อมูลที่บอกให้ผู้ใช้ทราบถึงชื่อเพิ่มข้อมูลที่กำลังถูกใช้งานอยู่

Cols เป็นข้อมูลที่บอกให้ผู้ใช้ทราบถึงจำนวนของอักขระทั้งหมดที่มีอยู่ในแถวที่เคอร์เซอร์อยู่ ตัวอย่างเช่น ในกรณีที่ข้อมูลดังกล่าวเป็น 86 ก็จะบอกให้ผู้ใช้ทราบว่าในแถวที่เคอร์เซอร์อยู่ นั้น มีจำนวนอักขระทั้งหมด 86 อักขระ

Lines เป็นข้อมูลที่บอกให้ผู้ใช้ทราบถึงจำนวนของแถวทั้งหมดที่มีอยู่ในแฟ้มข้อมูลที่กำลังถูกใช้ ตัวอย่างเช่น ในกรณีที่ข้อมูลดังกล่าวเป็น Lines 30 ก็จะบอกให้ทราบว่าจำนวนแถวในแฟ้มข้อมูลที่ผู้ใช้กำลังใช้งานอยู่นั้นมีอยู่ทั้งหมด 30 แถว

ส่วนที่ 2 เป็นเนื้อที่สำหรับให้ผู้ใช้จัดการกับข้อความ

ส่วนที่ 3 เป็นส่วนที่ใช้แสดงคำสั่งที่สำคัญๆในการใช้งานโปรแกรม ซึ่งเป็นการอำนวยความสะดวกให้กับผู้ใช้ไม่ต้องจดจำคำสั่งเอง



รูปที่ ก.1 จอภาพเริ่มต้นสำหรับการทำงานในกรณีที่ใช้งานเอดิเตอร์ PE
ด้วยชื่อแฟ้มข้อมูลใหม่ชื่อ NEW_FILE

ส่วนที่ 1

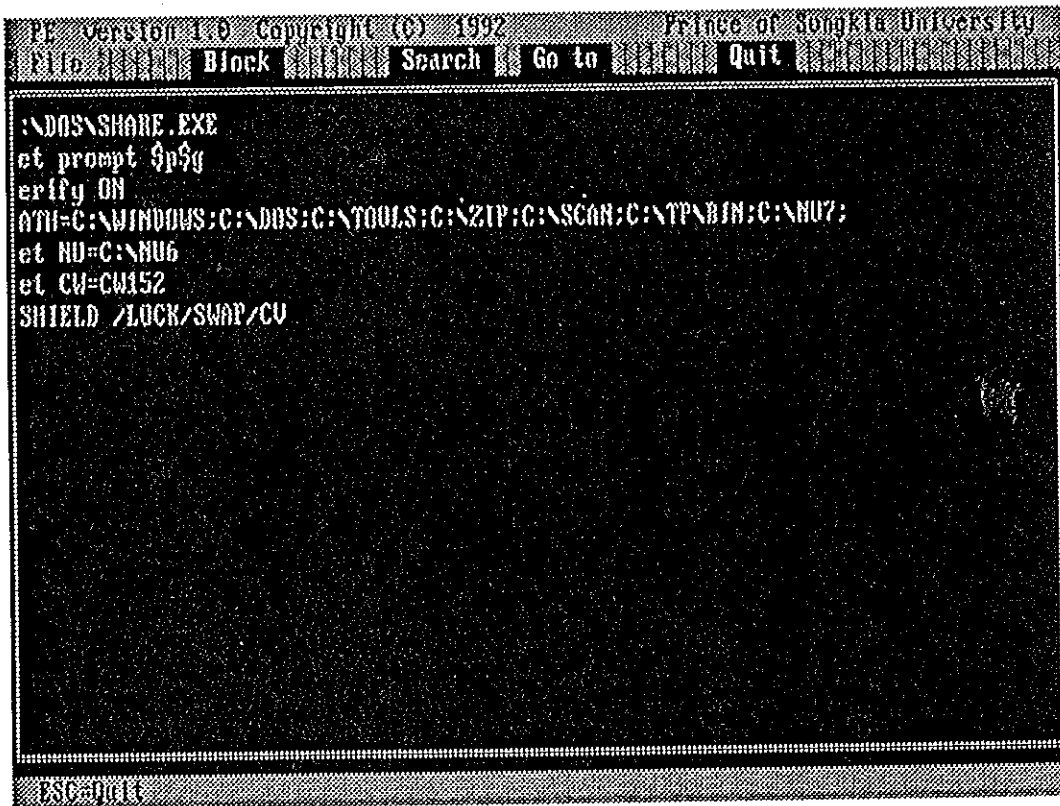
```
Line 1 Col 1 Ins C:\AUTOEXEC.BAT Cols 16 Lines 16
ECHO OFF
set comspec=c:\command.com
E:\DOS\SHARE.EXE
set prompt $p$g
verify ON
PATH=C:\WINDOWS;C:\DOS;C:\TOOLS;C:\ZIP;C:\SCAN;C:\ATP\BIN;C:\NU7;
set NU=C:\NU6
set CU=CW152
MSHIELD /LOCK/SHAP/CO
```

ส่วนที่ 2

ส่วนที่ 3

```
F1: Help F2: Save F3: Load F4: Find F5: Replace F9: That/End F10: Menu Alt X: Quit
```

รูปที่ ก.2 จอภาพเริ่มต้นสำหรับการทำงานในกรณีที่ใช้เอ็ดิเตอร์ PE ด้วยชื่อแฟ้มข้อมูลเก่า



```
Princo of Songkla University  
File Block Search Go to Quit  
:\DOS\SHARE.EXE  
et prompt 9p9g  
erifg 0H  
ATH=C:\WINDOWS;C:\DOS;C:\TOOLS;C:\ZIP;C:\SCAN;C:\TP\BIN;C:\NU7;  
et NU=C:\NU6  
et CU=CW152  
SHIELD /LOCK/SWAP/CU  
ESC=QUIT
```

รูปที่ ก.3 จอภาพที่ได้เมื่อผู้ใช้กดปุ่มฟังก์ชันคีย์ F10 จากหน้าจอรูปที่ ก.1 หรือ ก.2

รูปที่ ก.1 และ รูปที่ ก.2 เป็นจอภาพที่แสดงให้เห็นถึงความแตกต่างของการใช้งานเอดิเตอร์ PE กับเพิ่มข้อมูล "NEW_FILE" ซึ่งเป็นเพิ่มข้อมูลใหม่ที่ไม่มีข้อความบรรจุอยู่ และเพิ่มข้อมูล "AUTOEXEC.BAT" ซึ่งเป็นเพิ่มข้อมูลที่มีข้อความบรรจุอยู่ สิ่งเหมือนกันระหว่างจอภาพทั้งสองก็คือ ตำแหน่งของเคอร์เซอร์ ซึ่งเมื่อเริ่มต้นทำงานจะอยู่ ณ ตำแหน่ง สดมภ์ที่ 1 และแถวที่ 1 (แต่เป็นแถวที่สองของจอภาพ) เสมอ สำหรับสิ่งที่แตกต่างกันระหว่างจอภาพทั้งสองก็คือ ในการทำงานกับเพิ่มข้อมูลใหม่ เอดิเตอร์ PE จะทำให้จอภาพว่างเปล่า ส่วนในการทำงานกับเพิ่มข้อมูลที่มีข้อความบรรจุอยู่ เอดิเตอร์ PE นำข้อความดังกล่าวจากเพิ่มข้อมูลมาแสดงบนจอภาพ

จากรูปที่ ก.1 หรือ ก.2 เมื่อผู้ใช้กดปุ่มฟังก์ชันคีย์ F10 จอภาพจะเปลี่ยนเป็นจอภาพดังในรูปที่ ก.3 ซึ่งเป็นหน้ารายการหลักของเอดิเตอร์ PE ในรูปรายการเลือกแบบดิ่งลง การเลือกรายการหลักทั้ง 5 และรายการย่อยในแต่ละรายการหลัก สามารถทำได้โดยการใช้ปุ่มแป้นพิมพ์ดังกล่าวในหัวข้อต่อไป

4 ปุ่มแป้นพิมพ์แทนคำสั่งการดำเนินงาน

สัญลักษณ์ที่ใช้แทนคำสั่งในการดำเนินงานมีรูปแบบและความหมายแทนการทำงาน

ดังนี้

ปุ่มแป้นพิมพ์	การทำงาน
ข้อความบนพื้นสีเข้ม	เป็นรายการที่ถูกเลือกใช้ ณ ขณะนั้น
Alt + อักขระตัวเข้ม	เลือกรายการหลักพร้อมกับแสดงรายการย่อย
→	เลือกรายการหลักพร้อมกับแสดงรายการย่อยทางด้านขวาของรายการหลักปัจจุบัน
←	เลือกรายการหลักพร้อมกับแสดงรายการย่อยทางด้านซ้ายของรายการหลักปัจจุบัน
↑ หรือ ↓	เลือกรายการย่อยภายในรายการหลัก
รายการย่อยบนพื้นสีเข้ม	เลือกรายการย่อยนี้ภายในรายการหลักพร้อมกับให้ปฏิบัติงานตามคำสั่งนั้นๆ
Return หรือ Enter	ยืนยันการเลือกเพื่อให้ปฏิบัติงานให้

ในการพิมพ์ข้อความจะใช้ปุ่มแป้นพิมพ์อันได้แก่ ตัวอักขระ ตัวเลข เครื่องหมายวรรคตอน สัญลักษณ์ทางคณิตศาสตร์ และสัญลักษณ์พิเศษ เช่น @ หรือ # เป็นต้น ส่วนปุ่มฟังก์ชันคีย์ และ ปุ่มพิเศษอื่นๆ เช่น Home, End, Delete, Page Up, Page Down, Esc หรือ ปุ่มลูกศร จะนำไปใช้ในการแก้ไขและการแสดงข้อความบนจอภาพ

5 รายการหลัก และรายการย่อยบนรายการเลือกหลัก

รายการเลือกหลักของเอดิเตอร์ PE ประกอบด้วย 5 รายการหลัก คือ

- File บรรจुरายการย่อยซึ่งเป็นคำสั่งในการดำเนินงานกับเพิ่มข้อมูล
- Block บรรจुरายการย่อยซึ่งเป็นคำสั่งในการดำเนินงานกับข้อความในลักษณะเป็นกลุ่มของข้อความที่เรียกว่าบล็อก (block)
- Search บรรจुरายการย่อยซึ่งเป็นคำสั่งในการดำเนินงานค้นหา หรือแทนที่ข้อความ
- Go to บรรจुरายการย่อยซึ่งเป็นคำสั่งในการดำเนินงานกับการเคลื่อนที่ของเคอร์เซอร์
- Quit บรรจुरายการย่อยซึ่งเป็นคำสั่งในการดำเนินงานกับการติดต่อกับระบบปฏิบัติการ DOS ชั่วคราว หรือออกจากการใช้เอดิเตอร์ PE ชั่วคราว

รายการหลัก File

ในรายการหลัก File จะมีรายการย่อย Load, Save, Write to, Copy, Delete และ Rename ดังแสดงในรูปที่ ก.4 ซึ่งแต่ละรายการย่อยจะเป็นคำสั่งการทำงานเกี่ยวกับเพิ่มข้อมูล

รายการย่อย Load เป็นคำสั่งที่ใช้ในการอ่านข้อความจากเพิ่มข้อมูลที่ผู้ใช้งานต้องการใช้งานไปเก็บในความจำหลักที่จองไว้ จากนั้นจึงนำไปแสดงบนจอภาพ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Load จากรายการหลัก File หรือโดยการกดปุ่มฟังก์ชันคีย์ F3 ในกรณีที่เพิ่มข้อมูลมีขนาดใหญ่กว่าขนาดของความจำหลักที่จองไว้ เอดิเตอร์ PE จะไม่สามารถอ่านข้อความทั้งหมดจากเพิ่มข้อมูลไปเก็บยังความจำหลักที่จองไว้ได้ โดยจะแสดงข้อความบอกให้ผู้ใช้งานทราบ และภายหลังจากผู้ใช้งานกดปุ่มแป้นพิมพ์ตามที่เอดิเตอร์ PE กำหนดเพื่อแสดงว่ารับทราบปัญหาที่เกิดขึ้นแล้ว เอดิเตอร์ PE ก็จะยกเลิกคำสั่งอ่านข้อมูลจากเพิ่มข้อมูลดังกล่าว

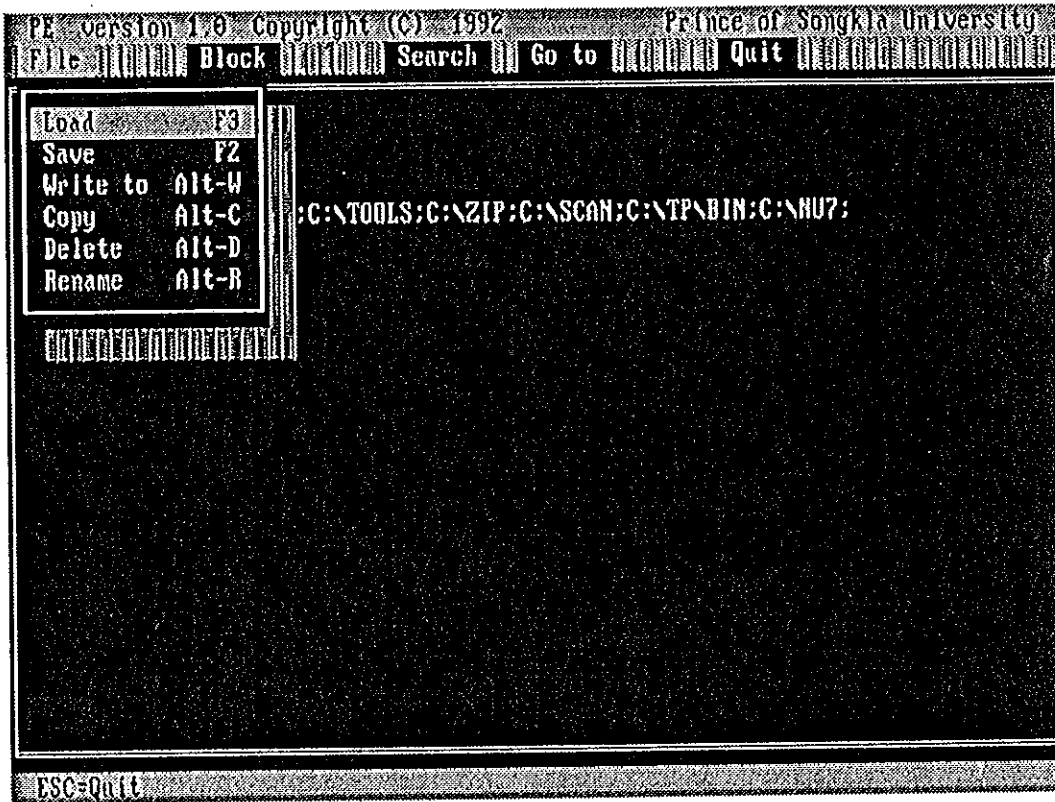
รายการย่อย Save เป็นคำสั่งที่ใช้ในการนำข้อความจากความจำหลักไปเก็บยังหน่วยความจำสำรอง โดยจะเก็บภายใต้ชื่อเพิ่มข้อมูลที่กำลังใช้งานอยู่ในขณะนั้น ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Save จากรายการหลัก File หรือโดยการกดปุ่มฟังก์ชันคีย์ F2

รายการย่อย Write to เป็นคำสั่งที่ใช้ในการนำข้อความจากความจำหลักไปเก็บยังหน่วยความจำสำรองภายใต้ชื่อเพิ่มข้อมูลของผู้ใช้ระบุ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Write to จากรายการหลัก File หรือโดยการกดปุ่มแป้นพิมพ์ Alt และปุ่มอักขระ W พร้อมกัน

รายการย่อย Copy เป็นคำสั่งที่ใช้ในการทำสำเนาเพิ่มข้อมูลในหน่วยความจำสำรอง ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Copy จากรายการหลัก File หรือโดยการกดปุ่มแป้นพิมพ์ Alt และ ปุ่มอักขระ C พร้อมกัน

รายการย่อย Delete เป็นคำสั่งที่ใช้ในการลบเพิ่มข้อมูลในหน่วยความจำสำรอง ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Delete จากรายการหลัก File หรือโดยการกดปุ่มแป้นพิมพ์ Alt และ ปุ่มอักขระ D พร้อมกัน

รายการย่อย Rename เป็นคำสั่งที่ใช้ในการเปลี่ยนชื่อเพิ่มข้อมูลในหน่วยความจำสำรอง ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Rename จากรายการหลัก File หรือโดยการกดปุ่มแป้นพิมพ์ Alt และ ปุ่มอักขระ R พร้อมกัน



รูปที่ ก.4 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก File

รายการหลัก Block

ในรายการหลัก Block จะมีรายการย่อย Begin, End, cOolumn, Copy, Delete, Move, Print, Read และ Write ดังแสดงในรูปที่ ก.5 ซึ่งแต่ละรายการย่อยจะเป็นคำสั่งการทำงานเกี่ยวกับการทำงานกับข้อความในรูปของบล็อก

รายการย่อย Begin เป็นคำสั่งที่ใช้ในการกำหนดจุดเริ่มต้นของบล็อก โดยการเลื่อนเคอร์เซอร์ไปยังอักขระตัวแรกของข้อความบนจอภาพที่ต้องการให้เป็นจุดเริ่มต้นของบล็อก จากนั้นจึงเลือกรายการย่อย Begin จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ B)

รายการย่อย End เป็นคำสั่งที่ใช้ในการกำหนดจุดสิ้นสุดของบล็อก โดยการเลื่อนเคอร์เซอร์ไปยังอักขระสุดท้ายของข้อความบนจอภาพที่ต้องการให้เป็นจุดสิ้นสุดของบล็อก ซึ่งจะต้องไม่อยู่ก่อนจุดเริ่มต้นของบล็อก จากนั้นจึงเลือกรายการย่อย End จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ K) ก็จะปรากฏแถบสว่างบนข้อความทั้งหมดที่ถูกกำหนดให้เป็นบล็อก

รายการย่อย cOolumn เป็นคำสั่งที่ใช้ในการกำหนดสถานะของบล็อก เป็นแบบแถวหรือแบบสดมภ์ภายหลังจากที่ผู้ใช้ได้จัดสร้างบล็อกเรียบร้อยแล้ว ลักษณะของบล็อกแบบสดมภ์จะสังเกตได้จากขอบซ้ายและขอบขวาของข้อความแต่ละแถวในบล็อกจะตรงกัน ซึ่งปกติแล้วสถานะของบล็อกจะเป็นแบบแถว แต่ผู้ใช้สามารถสลับเปลี่ยนสถานะของบล็อกด้วยการเรียกใช้คำสั่ง cOolumn ขึ้น การเรียกใช้คำสั่ง cOolumn สามารถทำได้ด้วยการเลือกรายการย่อย cOolumn จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ O)

รายการย่อย Copy เป็นคำสั่งที่ใช้ในการคัดลอกข้อความหนึ่งบล็อกไปยังตำแหน่งใด ๆ บนจอภาพ โดยการเลื่อนเคอร์เซอร์ไปยังตำแหน่งที่ต้องการนำข้อความของบล็อกมาได้ จากนั้นจึงใช้คำสั่ง Copy จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ C)

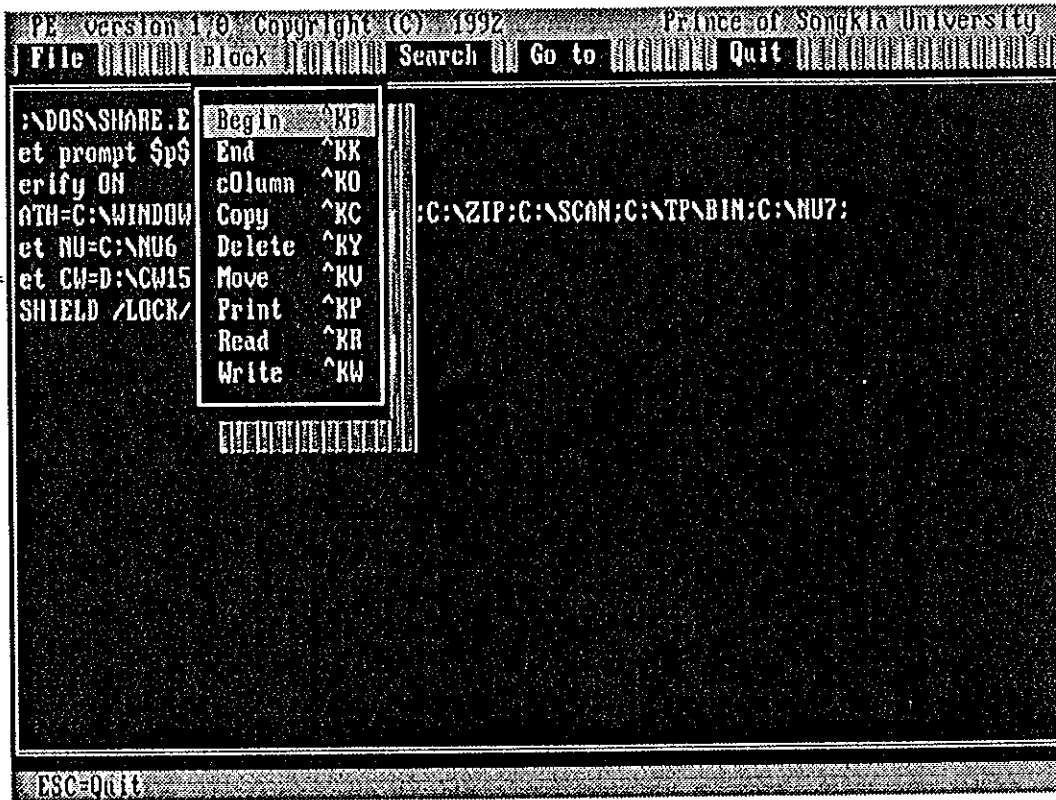
รายการย่อย Delete เป็นคำสั่งที่ใช้ในการลบบล็อก ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ ได้ด้วยการเลือกรายการย่อย Delete จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ Y)

รายการย่อย Move เป็นคำสั่งที่ใช้ในการเคลื่อนย้ายข้อความหนึ่งบล็อกไปยังตำแหน่งใด ๆ บนจอภาพ โดยการเลื่อนเคอร์เซอร์ไปยังตำแหน่งที่ต้องการเคลื่อนย้ายบล็อกมาได้ จากนั้นจึงใช้คำสั่ง Move จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ V)

รายการย่อย Print เป็นคำสั่งที่ใช้ในการคัดลอกข้อความหนึ่งบล็อกออกจากเครื่องพิมพ์ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ ได้ด้วยการเลือกรายการย่อย Print จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ P)

รายการย่อย Read เป็นคำสั่งที่ใช้ในการอ่านข้อความจากแฟ้มข้อมูลที่ต้องการมาแสดงบนจอภาพ ณ ตำแหน่งที่เคอร์เซอร์ปรากฏอยู่ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ ได้ด้วยการเลือกรายการย่อย Read จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ R) ในกรณีที่แฟ้มข้อมูลที่ต้องการใช้งานมีขนาดใหญ่กว่าขนาดของความจำหลักที่จองไว้ เอดิเตอร์ PE ก็จะไม่สามารถอ่านข้อมูลจากแฟ้มข้อมูลไปเก็บยังความจำหลักที่จองไว้ได้ และจะแสดงข้อความบอกให้ผู้ใช้ทราบ และภายหลังจากผู้ใช้กดปุ่มแป้นพิมพ์ตามที่เอดิเตอร์ PE กำหนดเพื่อแสดงว่ารับทราบปัญหาที่เกิดขึ้นแล้ว เอดิเตอร์ PE ก็จะยกเลิกคำสั่งอ่านข้อมูลจากแฟ้มข้อมูลดังกล่าว

รายการย่อย Write เป็นคำสั่งที่ใช้ในการคัดลอกข้อความหนึ่งบล็อกไปเก็บยังหน่วยความจำสำรองภายใต้ชื่อแฟ้มข้อมูลที่ผู้ใช้ระบุ ซึ่งผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ ได้ด้วยการเลือกรายการย่อย Write จากรายการหลัก Block (หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl ค้างอยู่และกดปุ่มอักขระ K และ W)



รูปที่ ก.5 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Block

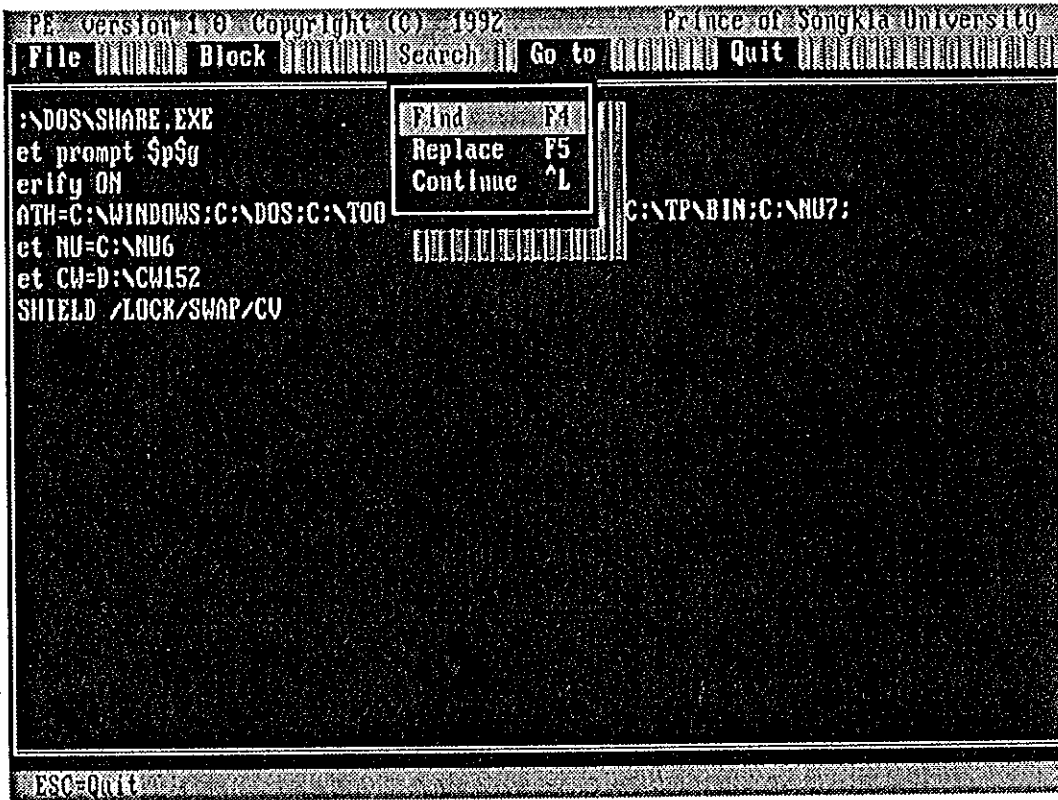
รายการหลัก Search

ในรายการหลัก Search จะมีรายการย่อย Find, Replace และ Continue ดังแสดงในรูปที่ ก.6 ซึ่งแต่ละรายการย่อยจะเป็นคำสั่งการทำงานเกี่ยวกับการค้นหาข้อความ

รายการย่อย Find เป็นคำสั่งที่ใช้ในการค้นหาข้อความที่ผู้ใช้ต้องการ โดยที่ผู้ใช้สามารถกำหนดเงื่อนไขของการค้นหาซึ่งมีอยู่ด้วยกัน 3 อย่างดังนี้ คือ ในการค้นหาไม่สนใจขนาดของตัวอักษรภาษาอังกฤษในข้อความ โดยถือว่าอักขระตัวใหญ่และตัวเล็กเหมือนกัน การค้นหาเริ่มจากตำแหน่งที่เคอร์เซอร์อยู่ไปจนกระทั่งจบเพิ่มข้อมูล และการค้นหาเริ่มจากตำแหน่งที่เคอร์เซอร์อยู่ย้อนกลับไปถึงต้นเพิ่มข้อมูล ในการเรียกใช้คำสั่ง Find สามารถทำได้ด้วยการเลือกรายการย่อย Find จากรายการหลัก Search หรือโดยการกดปุ่มฟังก์ชันคีย์ F4

รายการย่อย Replace เป็นคำสั่งที่ใช้ในการแทนที่ข้อความเดิมด้วยข้อความที่ผู้ใช้ต้องการ โดยที่ผู้ใช้สามารถกำหนดเงื่อนไขของการแทนที่ซึ่งมีอยู่ด้วยกัน 2 อย่างดังนี้ คือ ในการแทนที่ไม่สนใจขนาดของตัวอักษรภาษาอังกฤษในข้อความ โดยถือว่าอักขระตัวใหญ่และตัวเล็กเหมือนกัน และแทนที่ข้อความเดิมโดยไม่รอคำยืนยันจากผู้ใช้ ในการเรียกใช้คำสั่ง Replace สามารถทำได้ด้วยการเลือกรายการย่อย Replace จากรายการหลัก Search หรือโดยการกดปุ่มฟังก์ชันคีย์ F5

รายการย่อย Continue เป็นคำสั่งที่ใช้ในการค้นหาข้อความที่ผู้ใช้ต้องการซ้ำอีกครั้ง ด้วยการเลือกรายการย่อย Continue จากรายการหลัก Search หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl และปุ่มอักขระ L พร้อมกัน



รูปที่ ก.6 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Search

รายการหลัก Go to

ในรายการหลัก Go to จะมีรายการย่อย Begin of File, End of File, Top Screen, bOttom Screen และ Line ดังแสดงในรูปที่ ก.7 ซึ่งแต่ละรายการย่อยจะเป็นคำสั่งการทำงานเกี่ยวกับการเคลื่อนที่ของเคอร์เซอร์

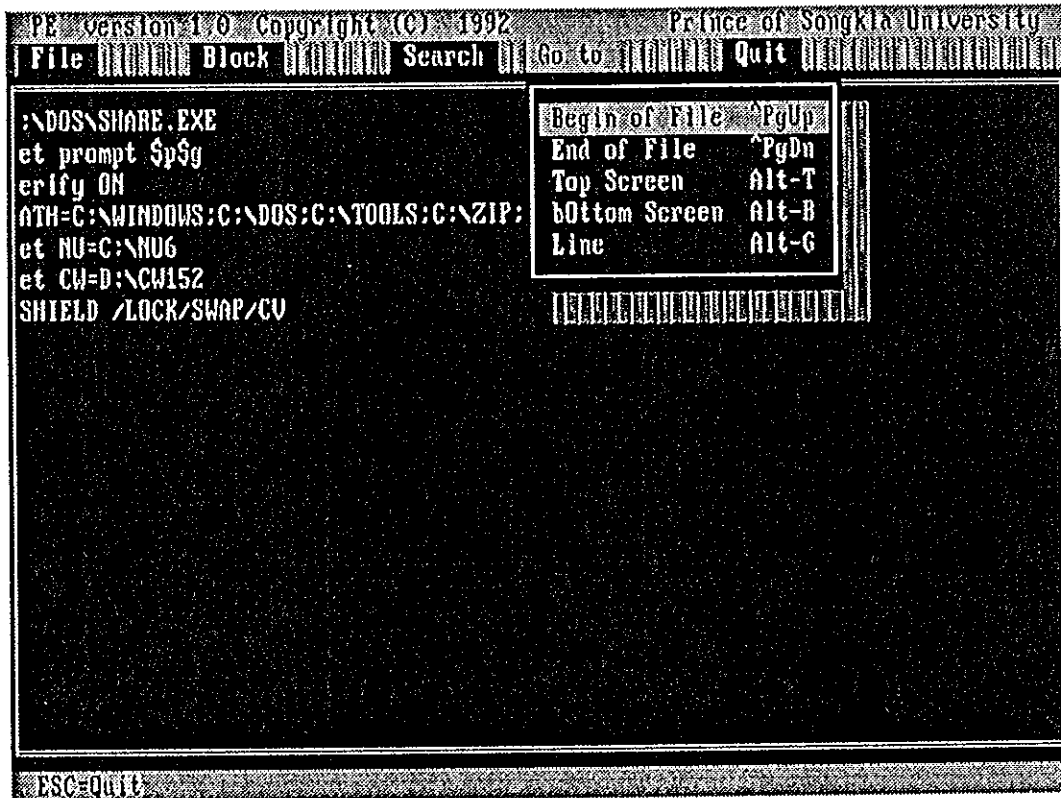
รายการย่อย Begin of File เป็นคำสั่งที่ใช้ในการเคลื่อนย้ายเคอร์เซอร์ไปยังต้นแฟ้มข้อมูล ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Begin of File จากรายการหลัก Go to หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl และ Page Up พร้อมกัน

รายการย่อย End of File เป็นคำสั่งที่ใช้ในการเคลื่อนย้ายเคอร์เซอร์ไปยังท้ายแฟ้มข้อมูล ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย End of File จากรายการหลัก Go to หรือโดยการกดปุ่มแป้นพิมพ์ Ctrl และ Page Down พร้อมกัน

รายการย่อย Top Screen เป็นคำสั่งที่ใช้ในการเคลื่อนย้ายเคอร์เซอร์ไปยังบรรทัดบนสุดของจอภาพ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Top Screen จากรายการหลัก Go to หรือโดยการกดปุ่มแป้นพิมพ์ Alt และปุ่มอักขระ T พร้อมกัน

รายการย่อย bOttom Screen เป็นคำสั่งที่ใช้ในการเคลื่อนย้ายเคอร์เซอร์ไปยังบรรทัดล่างสุดของจอภาพ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย bOttom Screen จากรายการหลัก Go to หรือโดยการกดปุ่มแป้นพิมพ์ Alt และปุ่มอักขระ B พร้อมกัน

รายการย่อย Line เป็นคำสั่งที่ใช้ในการเคลื่อนย้ายเคอร์เซอร์ไปยังบรรทัดใดๆตามที่ใช้ต้องการ ซึ่งหมายเลขบรรทัดที่ผู้ใช้ระบุนี้จะต้องไม่มากกว่าจำนวนบรรทัดทั้งหมดที่มีอยู่ในแฟ้มข้อมูล ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Line จากรายการหลัก Go to หรือโดยการกดปุ่มแป้นพิมพ์ Alt และปุ่มอักขระ G พร้อมกัน



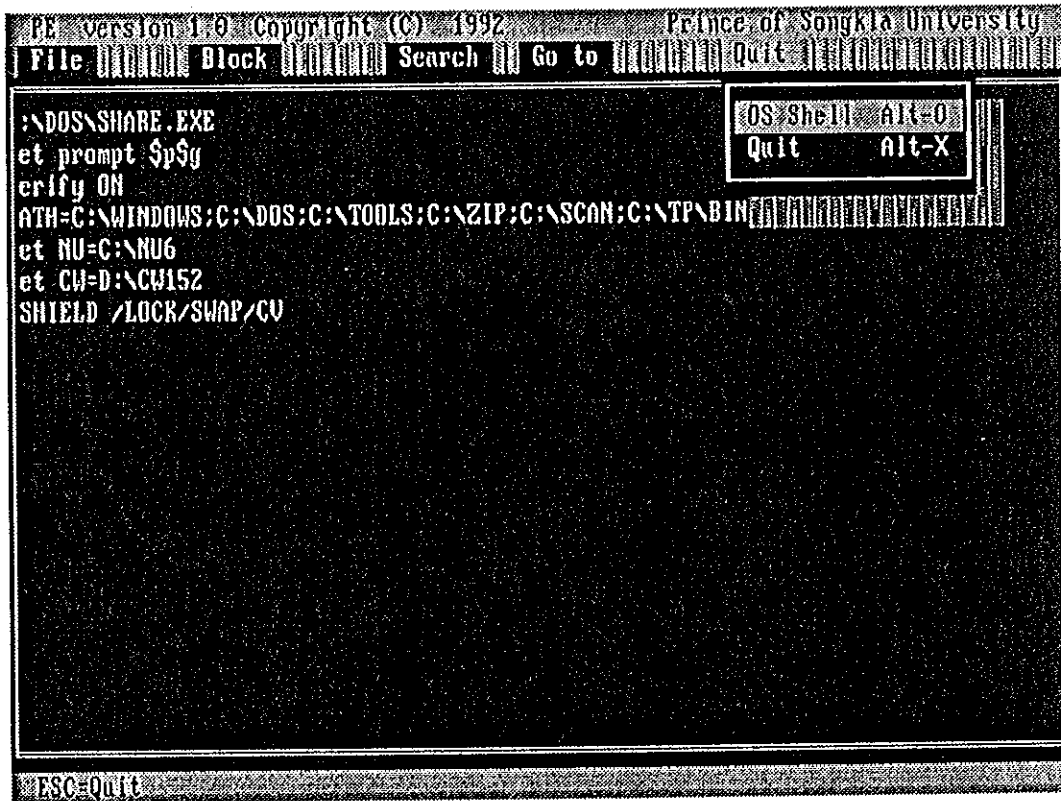
รูปที่ ก.7 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Go to

รายการหลัก Quit

ในรายการหลัก Quit จะมีรายการย่อย OS Shell และ Quit ดังแสดงในรูปที่ ก.8 ซึ่งแต่ละรายการย่อยจะเป็นคำสั่งการทำงานเกี่ยวกับการติดต่อกับระบบปฏิบัติการ

รายการย่อย OS Shell เป็นคำสั่งที่ใช้สำหรับหยุดการใช้งานเอดิเตอร์ PE ชั่วคราวเพื่อกลับไปสู่ระบบปฏิบัติการ DOS และถ้าหากผู้ใช้ต้องการกลับเข้าสู่เอดิเตอร์ PE อีกครั้ง ก็ให้พิมพ์คำว่า "Exit" ต่อจากเครื่องหมาย Prompt ขณะที่อยู่ที่ระบบปฏิบัติการ ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย OS Shell จากรายการหลัก Quit หรือโดยการกดปุ่มแป้นพิมพ์ Alt และปุ่มอักขระ O พร้อมกัน

รายการย่อย Quit เป็นคำสั่งที่ใช้สำหรับเลิกใช้งานเอดิเตอร์ PE และกลับไปสู่ระบบปฏิบัติการ DOS ผู้ใช้สามารถเรียกใช้คำสั่งดังกล่าวนี้ได้ด้วยการเลือกรายการย่อย Quit จากรายการหลัก Quit หรือโดยการกดปุ่มแป้นพิมพ์ Alt และปุ่มอักขระ X พร้อมกัน



รูปที่ ก.8 จอภาพแสดงรายการย่อยทั้งหมดของรายการหลัก Quit

```
Type EXIT to return to program . . .  
Microsoft(R) MS-DOS(R) Version 6.22  
(C)Copyright Microsoft Corp 1981-1994.  
C:\JRW>
```

รูปที่ ก.9 จอภาพภายหลังจากใช้คำสั่ง OS Shell

6. คำสั่งช่วยเหลือการปฏิบัติงาน

คำสั่งสำหรับช่วยเหลือการใช้งานเอดิเตอร์ PE สำหรับผู้ใช้ (help) ขณะกำลังใช้งานอยู่ เรียกใช้โดยการกดปุ่มฟังก์ชันคีย์ F1 จะเป็นคำสั่งที่ช่วยเหลือการใช้งานโปรแกรมของผู้ใช้ในกรณีที่ผู้ใช้ไม่ทราบ หรือเกิดหลงลืมคำสั่งการใช้งานบางคำสั่ง ซึ่งการเรียกใช้คำสั่งช่วยเหลือการใช้งานนี้ สามารถเรียกใช้ได้ตลอดเวลาในขณะที่อยู่ในสถานะการใช้งานโปรแกรมโดยตรง เมื่อผู้ใช้กดปุ่มฟังก์ชันคีย์ F1 จะปรากฏกรอบหน้าต่างบรรจุคำสั่งการใช้งานโปรแกรม ณ ที่กลางจอภาพดังรูปที่ ก.10

```

Line 1  Col 1  Ins  C:\AUTOEXEC.BAT  Col 10  Lines 10
ECHO OFF
set comspec=c:\command.com
C:\DOS\SHARE.EXE
set prompt $p$g
erify ON
PATH=C:\WINDOWS:
set NU=C:\NU6
set CW=D:\CW152
SHIELD /LOCK/SW

```

Help

Cursor Movement Commands

Character left	Ctrl-S or <--
Character right	Ctrl-D or -->
Word left	Ctrl-A or Ctrl + <--
Word right	Ctrl-F or Ctrl + -->
Line up	Ctrl-E or Up arrow
Line down	Ctrl-X or Down arrow
Scroll up	Ctrl-W
Scroll down	Ctrl-Z
Page up	Ctrl-B or PgUp
Page down	Ctrl-C or PgDn

PgDn

```

F1=Help  F2=Save  F3=Load  F4=Find  F5=Replace  F9=Thru/End  F10=Menu  Alt-X=Quit

```

รูปที่ ก.10 จอภาพขณะใช้คำสั่ง Help (F1)

```

Line 1 Col 1 Ins C:\AUTODEXC\BAT Cols 10 Lines 10
ECHO OFF
set comspec=c:\command.com
C:\DOS\SHARE.EXE
set prompt $p$g
verify ON
PATH=C:\WINDOWS:
set NU=C:\NU6
set CW=D:\CW152
MSHIELD /LOCK/SW

```

Help

Insert & Delete Commands

Insert mode on/off	Ins
Insert line	Ctrl-N
Delete line	Ctrl-Y
Yank line	Alt-Y
Delete to end of line	Ctrl-Q Y
Delete character left	Ctrl-H or Backspace
Delete character right	Ctrl-G or Del
Delete word right	Ctrl-T

PgUp/PgDn

```

F1=Help F2=Save F3=Load F4=Find F5=Replace F9=Tab/Esc F10=Menu Alt-X=Quit

```

รูปที่ ก.10(ต่อ)


```

Line 1      Col 1      Ins  C:\AUTOEXEC.BAT      Cols 10      Lines 10
ECHO OFF
set comspec=c:\command.com
C:\DOS\SHARE.EXE
set prompt $p$g
verify ON
PATH=C:\WINDOWS:
set NU=C:\NU6
set CW=D:\CW152
MSHIELD /LOCK/SW

```

Help

Block Commands

Mark block begin	Ctrl-K B
Mark block end	Ctrl-K K
Mark block column	Ctrl-K E
Copy block	Ctrl-K C
Delete block	Ctrl-K Y
Move block	
Read block from disk	Ctrl-K R
Write block to disk	Ctrl-K W
Hide/display block	Ctrl-K H
Print block	Ctrl-K P

PgUp

```

F1=Help F2=Save F3=Load F4=Find F5=Replace F9=Thru/Eng F10=Menu Alt-X=Quit

```

รูปที่ ก.10(ต่อ)

7. สรุปคำสั่งใช้งานโปรแกรมเอดิเตอร์ PE

ตารางที่ ก1 รวบรวมคำสั่งทั้งหมดสำหรับการใช้งานเอดิเตอร์ PE

คำสั่งที่ใช้	การทำงาน
^S, ←	เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง
^D, →	เลื่อนเคอร์เซอร์ไปทางขวา 1 ตำแหน่ง
^E, ↑	เลื่อนเคอร์เซอร์ขึ้นบน 1 บรรทัด
^X, ↓	เลื่อนเคอร์เซอร์ลงล่าง 1 บรรทัด
^F, ^ →	เลื่อนเคอร์เซอร์ไปคำถัดไป
^A, ^ ←	เลื่อนเคอร์เซอร์ไปคำที่แล้ว
^QR, ^PgUp	เลื่อนเคอร์เซอร์ไปต้นเพิ่มข้อมูล
^QC, ^PgDn	เลื่อนเคอร์เซอร์ไปท้ายเพิ่มข้อมูล
Home	เลื่อนเคอร์เซอร์ไปต้นบรรทัด
End	เลื่อนเคอร์เซอร์ไปท้ายบรรทัด
Alt-T	เลื่อนเคอร์เซอร์ไปบรรทัดแรกของหน้าจอ
Alt-B	เลื่อนเคอร์เซอร์ไปบรรทัดสุดท้ายของหน้าจอ
Alt-G	เลื่อนเคอร์เซอร์ไปบรรทัดที่ต้องการ
^W	เลื่อนจอขึ้น 1 บรรทัด
^Z	เลื่อนจอลง 1 บรรทัด
^R, PgUp	เลื่อนข้อความขึ้น 1 หน้าจอ
^C, PgDn	เลื่อนข้อความลง 1 หน้าจอ
Tab	เลื่อนเคอร์เซอร์ไปยังย่อหน้าถัดไป
^QF, F4	ค้นหาคำ/ข้อความที่ต้องการ
^QA, F5	แทนที่คำ/ข้อความที่ต้องการ
^L	ค้นหาคำ/ข้อความต่อไป
^G, Del	ลบตัวอักขระ ณ ตำแหน่งของเคอร์เซอร์
^H, BS	ลบตัวอักขระทางซ้ายของเคอร์เซอร์
^T	ลบคำ/ข้อความ
^Y	ลบข้อความทั้งบรรทัด

ตารางที่ ก1(ต่อ)

คำสั่งที่ใช้	การทำงาน
^QY	ลบข้อความจากตำแหน่งของเคอร์เซอร์ไปจนหมดบรรทัด
^N	แทรกบรรทัดว่าง
^V ,Ins	พิมพ์แทรก/ทับ
Alt-A+Y	ลบข้อความจากตำแหน่งปัจจุบันจนจบเพิ่มข้อมูล
Alt-Y	เอาข้อความที่ถูกลบไปกลับคืน
^KB	กำหนดตำแหน่งเริ่มต้นบล็อก
^KK	กำหนดตำแหน่งสิ้นสุดบล็อก
^KC	คัดลอกบล็อก
^KC	ลบบล็อก
^KV	เคลื่อนย้ายบล็อก
^KR	อ่านบล็อก
^KW	เขียนบล็อกไปยังเพิ่มข้อมูล
^KH	ซ่อน/แสดงบล็อก
^KP	พิมพ์บล็อกออกทางเครื่องพิมพ์
^KO	กำหนดบล็อกชนิดคอลัมน์
F10	แสดงรายการเลือกแบบดึงลงหลัก
Esc	ยกเลิกรายการเลือกแบบดึงลงและกลับไปสู่นำจอหลัก
F3	อ่านข้อความจากเพิ่มข้อมูล
F2	นำข้อความจากความจำหลักไปเก็บยังหน่วยความจำสำรองภายใต้เพิ่มข้อมูลปัจจุบัน
Alt-W	นำข้อความจากความจำหลักไปเก็บยังหน่วยความจำสำรองภายใต้ชื่อเพิ่มข้อมูลที่ใช้ระบุ
Alt-C	ทำสำเนาเพิ่มข้อมูลในหน่วยความจำสำรอง
Alt-D	ลบเพิ่มข้อมูลในหน่วยความจำสำรอง
Alt-R	เปลี่ยนชื่อเพิ่มข้อมูลในหน่วยความจำสำรอง

ตารางที่ ก1(ต่อ)

คำสั่งที่ใช้	การทำงาน
Alt-O	หยุดการใช้งานเอดิเตอร์ PE ชั่วคราวเพื่อกลับไปสู่ระบบปฏิบัติการ DOS
Alt-X	เลิกใช้งานเอดิเตอร์ PE และกลับไปสู่ระบบปฏิบัติการ DOS
F1	คำสั่งช่วยเหลือการปฏิบัติงาน

ประวัติผู้เขียน

ชื่อ นายจิรวุธ มุนินทร์พมาศ

เกิดวันที่ 12 มีนาคม 2509

วุฒิการศึกษา

วุฒิ

ชื่อสถาบัน

ปีที่สำเร็จการศึกษา

วิทยาศาสตรบัณฑิต (ศึกษาศาสตร์)

มหาวิทยาลัยสงขลานครินทร์

2531

วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

มหาวิทยาลัยสงขลานครินทร์

2539

ตำแหน่งและสถานที่ทำงาน นักวิเคราะห์ระบบ บริษัทดิจิทัล อินฟอร์เมชั่น ซิสเต็ม