

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องมือสร้างและแก้ไขข้อมูลโสมเพออย่างง่ายบนเซิร์ฟเวอร์

SIMPLE WEB EDITOR FOR AUTOMATIC CREATE AND UPDATE ON SERVER



โดย
นายภัทรพงศ์ น้อยเรือง
นายวีรวัฒน์ หวังอายุตวณิชย์

อาจารย์ที่ปรึกษา
อ. นวพร วรรณวิมลศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 37032
วัน, เดือน, ปี 30 ส.ค. 2543

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หากมีการเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องมือสร้างและแก้ไขข้อมูลโฮมเพจอย่างง่ายบนเซิร์ฟเวอร์

SIMPLE WEB EDITOR FOR AUTOMATIC CREATE AND UPDATE ON SERVER

ผู้จัดทำ

1. นาย ภัทรพงศ์ น้อยเรือง รหัสประจำตัว 39014384
2. นาย วีราวัฒน์ หวังอาทิตย์วิชัย รหัสประจำตัว 39014496



อาจารย์ที่ปรึกษา

(อ. นวพร วรรณวิมลศรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือสร้างและแก้ไขข้อมูลโฮมเพจอย่างง่ายบนเซิร์ฟเวอร์

นายภัทรพงศ์ น้อยเรือง 39014384

นายวีราวัฒน์ หวังอายุตวณิชย์ 39014496

อ. นวพร วรณวิมลศรี อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

บทคัดย่อ

ในปัจจุบันเทคโนโลยีทางด้านอินเทอร์เน็ต (INTERNET) มีการพัฒนาค่อนข้างรวดเร็ว ทั้งในแง่ของประสิทธิภาพในการทำงานที่เร็วมากขึ้น ทั้งในแง่ของความสามารถที่มากขึ้นนั้น ทำให้ได้รับการนำไปใช้งานกันค่อนข้างกว้างขวาง กลุ่มของผู้ใช้งานก็ขยายตัวกว้างขึ้นเรื่อยๆ เนื่องจากเป็นแหล่งที่สามารถให้ข้อมูล, ความรู้, ความบันเทิง, และความสะดวกรวดสบายต่างๆ บริการที่เป็นที่นิยมอย่างหนึ่งคือเวิร์ลไวด์เว็บ(World Wide Web) เป็นบริการที่มีผู้นิยมใช้เป็นจำนวนมาก

ปัญญานิพนธ์ฉบับนี้เป็นแนวทางหนึ่งในการสร้างเครื่องมือที่ใช้ในการสร้าง, แก้ไขและอัปเดตข้อมูลบนเว็บไซต์ เพื่อให้การทำงานเป็นไปอย่างสะดวกและมีประสิทธิภาพมากขึ้น อีกทั้งยังประยุกต์ใช้งาน CGI ในการสร้างเว็บบอร์ด (Webboard) ซึ่งเป็นกระดานข่าวโดยผู้ที่เข้ามาใช้งานสามารถที่จะดูข้อมูลที่นำเสนอหรือสร้างหัวข้อในการติดต่อสื่อสารกันได้และรอการตอบรับจากผู้ที่สนใจในหัวข้อนั้นๆ และผู้ดูแลระบบ (Administrator) สามารถที่จะจัดการกับระบบได้อย่างสะดวกโดยไม่ต้องเข้าไปยุ่งกับตัวโปรแกรมและฐานข้อมูล (Database) มากมายนัก โดยมีอุปกรณ์ให้ใช้งานในเว็บไซต์ได้เสีย และมีอุปกรณ์อำนวยความสะดวกอื่นเพิ่มเข้ามาอีกเช่น ตารางเวลานัดหมาย, การเตือน, ระบบตรวจสอบข้อความไม่สุภาพและอื่นๆ ซึ่งอุปกรณ์ต่างๆที่ได้กล่าวมานี้เพื่อเพิ่มประสิทธิภาพในการทำงานบนเว็บไซต์.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIMPLE WEB EDITOR FOR AUTOMATIC CREATE AND UPDATE ON SERVER

Pattarapong Noiraeng

Weeravat Wang-aryatwanich

Navaporn Warnwimonsri Advisor

ABSTRACT

Today's Internet technology has been rapidly progressive. The improvement in both efficiency and capability leads to an extensive implementation of Internet, and wider range of users, because the Internet is one of the most useful resources of information, knowledge and entertainment, especially the World Wide Web.

This project states a way to develop a tool used to create and update a website. CGI is applied to build a webboard for users to communicate and exchange information in the manner that each user can put up a topic they are interested in and wait for other user to put up his or her comments. Administrator can manage the system easily without having to deal directly with application programs and databases. The equipment to do these tasks is provided on the website. In addition, there are other utilities such as a time schedule, a warning system, a bad - language detection system.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญาานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ นวพร วรณวิมลศรี อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

ภัทรพงษ์ น้อยเรือง

วิราวัฒน์ หวังอายุวัฒน์ชัย

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 ระบบที่ใช้ในการทำงาน	4
2.2 หลักการทำงานของเว็บไซต์	4
2.2.1 ขั้นตอนแสดงถึงการทำงานขั้นต้น	4
2.2.2 เครื่องคอมพิวเตอร์ฝั่งไคลเอนต์	5
2.2.3 เครื่องคอมพิวเตอร์ฝั่งเซิร์ฟเวอร์	5
2.3 หลักการทำงานของซีจีไอ	5
2.3.1 การทำงานของโปรแกรมซีจีไอ	6
2.4 หลักการทำงานของ Perl	7
2.4.1 Perl	7
2.4.2 ตัวแปรภาษาของ Perl	8
2.4.3 เลือกใช้งานภาษา Perl เนื่องจากข้อดีต่างๆดังนี้	8
2.5 หลักการทำงานของ Linux	9
2.5.1 เลือกใช้งานระบบปฏิบัติการ Linux เนื่องจากข้อดีต่างๆ	9
2.6 หลักการทำงานของ APACHE	9
2.7 หลักการทำงานของ MySQL	10
2.7.1 เลือกใช้งานระบบฐานข้อมูล MySQL เนื่องจากข้อดีต่างๆ	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การคำนวณ การสร้าง การออกแบบ	11
3.1 หลักการทำงานของโปรแกรมสำหรับเปลี่ยนแปลงและแก้ไขข้อมูลผ่าน Browser (Update Data)	11
3.1.1 คุณสมบัติของโปรแกรม Update Data	11
3.1.2 หลักการทำงานของโปรแกรม Update Data	12
3.1.3 ไฟล์ที่เกี่ยวข้องและการทำงานของไฟล์แต่ละไฟล์	14
3.2 หลักการทำงานของโปรแกรมสร้างฟอร์มพื้นฐาน	15
3.2.1 ขั้นตอนและการทำงานของโปรแกรมสร้างฟอร์มพื้นฐาน	15
3.2.2 ไฟล์ที่เกี่ยวข้องและการทำงานของไฟล์แต่ละไฟล์	16
3.3 หลักการทำงานของโปรแกรมสร้างฟอร์ม	17
3.3.1 คุณสมบัติของ Form Editor	17
3.3.2 ขั้นตอนการทำงานของโปรแกรม	19
3.3.3 ไฟล์ที่เกี่ยวข้องและการทำงานของไฟล์แต่ละไฟล์	21
3.4 หลักการทำงานของกระดานข่าว	23
3.4.1 การออกแบบฐานข้อมูล	23
3.4.2 E-R Diagram ของกระดานข่าว	24
3.4.3 การทำงานของกระดานข่าวในรูปของ Flow Chart	25
3.4.4 แนวความคิดในการสร้างกระดานข่าว	26
3.4.5 การออกแบบและการทำงานในแต่ละส่วนของกระดานข่าว	27
3.4.5.1 การเข้าระบบ	27
3.4.5.2 การออกจากระบบ	28
3.4.5.3 แนวความคิดในการสร้าง Login / Logout	28
3.4.5.4 การสมัครสมาชิก	29
3.4.5.5 การแก้ไขข้อมูลผู้ใช้	30
3.4.5.6 แนวความคิดในการสร้างการแก้ไขข้อมูลผู้ใช้	30
3.4.5.7 การตั้งกระทู้	31
3.4.5.8 แนวความคิดในการสร้างการตั้งกระทู้	32
3.4.5.9 การตอบกระทู้	32
3.4.5.10 แนวความคิดในการสร้างการตอบกระทู้	33
3.4.5.11 การค้นหากระทู้	34
3.4.5.12 แนวความคิดในการสร้างการค้นหากระทู้	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.13	การทำงานของคู่มือกระดานข่าว	35
3.4.5.14	การลบกระทู้	36
3.4.5.15	แนวความคิดในการสร้างการลบกระทู้	36
3.4.5.16	การกำหนดค่าต่างๆของห้องกระทู้	36
3.4.5.17	การกำหนดค่าต่างๆของผู้ใช้	38
3.4.5.18	แนวความคิดในการสร้างการกำหนดค่าต่างๆของผู้ใช้	39
3.4.5.19	การกำหนดจำนวนกระทู้ที่แสดงต่อหน้า	40
3.4.5.20	แนวความคิดในการสร้างการกำหนดจำนวนกระทู้ที่แสดงต่อหน้า	40
3.4.5.21	ห้องกระทู้	40
3.4.5.22	แนวความคิดในการสร้างห้องกระทู้	41
3.4.5.23	โฮมเพจส่วนตัว	42
3.4.5.24	แนวความคิดในการสร้างโฮมเพจส่วนตัว	44
บทที่ 4	บทวิจารณ์ / สรุป	45
4.1	ข้อดีต่างๆในการทำงานของกระดานข่าว	45
4.2	ข้อเสียต่างๆในการทำงานของกระดานข่าว	46
บทที่ 5	ภาคผนวก	
5.1	การติดตั้งระบบที่ใช้งานต่างๆ	47
5.1.1	การติดตั้งระบบปฏิบัติการ Linux	47
5.1.1.1	เตรียมเครื่องคอมพิวเตอร์	47
5.1.1.2	จัดหาด้านแบบโปรแกรม Linux	47
5.1.1.3	บูทเป็นระบบ Linux เพื่อทำการติดตั้ง	48
5.1.1.4	เริ่มการติดตั้งและสำเนาไฟล์ต่างๆลงบนฮาร์ดดิสก์ของท่าน	48
5.1.1.5	เลือกวิธีการติดตั้ง	48
5.1.1.6	เลือกส่วนประกอบของ Linux	48
5.1.1.7	เริ่มการทำสำเนาไฟล์	48
5.1.2	การติดตั้งระบบเครือข่าย	48
5.1.3	การติดตั้ง WWW Server, Apache Server	48
5.1.3.1	WWW Server	48
5.1.3.2	การติดตั้ง Apache Server	49
5.1.4	การติดตั้งฐานข้อมูล	49
5.1.5	การติดตั้ง MySQL Grant Table	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2	พื้นฐานการใช้งาน Perl	50
5.2.1	ที่อยู่ของ Perl	50
5.2.2	การรันโปรแกรม Perl ในระบบ Unix	50
5.2.3	การเรียกใช้ Perl จาก Command line ของ DOS	51
5.2.4	สร้างโปรแกรม CGI อย่างง่ายๆ ให้แสดงผลที่หน้าจอบราวเซอร์	51
5.2.5	การลดปริมาณการใช้คำสั่ง Print	51
5.2.6	ชนิดของตัวแปร	52
5.2.7	หลักการกำหนดชื่อให้กับตัวแปร	54
5.2.8	การรับข้อมูลจากผู้ใช้	54
5.2.9	Perl และการคำนวณ	55
5.2.9.1	ลำดับของการคำนวณ	55
5.2.9.2	เงื่อนไขและการวนลูป	55
5.2.10	CGI Script	61
5.2.11	การจัดการไฟล์	67
5.2.12	การส่งข้อมูลจากฟอร์ม ไปยังที่อยู่ E-mail ของผู้รับ	70
5.3	พื้นฐานการใช้งาน MySQL	74
5.4	Java กับ Graphic	74
5.4.1	การ Compile และการรันโปรแกรม	75
5.4.2	การวาดภาพในวินโดว์	75
5.4.3	การสร้างเส้นตรง	76
5.4.4	การสร้างสี่เหลี่ยม	77
5.4.5	สี่เหลี่ยมมุมฉาก 3D	79
5.4.6	การสร้างวงกลมแต่ละส่วนโค้ง	80
	บรรณานุกรม	84

สารบัญภาพ

หน้าที่

รูปที่ 2-1	การส่งข้อมูลระหว่างเครื่องไคลเอ็นท์และเซิร์ฟเวอร์	4
รูปที่ 2-2	เป็นการขนถ่ายข้อมูลในส่วนของ โปรแกรมซีจีไอ	6
รูปที่ 2-3	โปรแกรมซีจีไอบนระบบปฏิบัติการ	7
รูปที่ 3-1	ข้อมูลของผู้ใช้งานที่ถูกเรียกขึ้นมาในขั้นแรก	12
รูปที่ 3-2	แสดงปุ่ม Update ข้อมูล	12
รูปที่ 3-3	แสดงข้อมูลของ User ขึ้นมาบน Block	13
รูปที่ 3-4	แสดงข้อมูลที่ถูกเปลี่ยนแปลงไปบน Block	13
รูปที่ 3-5	แสดงปุ่ม Submit	13
รูปที่ 3-6	แสดงข้อมูลที่ถูกเปลี่ยนแปลงไปแล้ว	14
รูปที่ 3-7	แสดงหน้าจอรับค่า Input File, HTML File และ CGI File	15
รูปที่ 3-8	แสดงตัวอย่างของฟอร์มที่สร้างโดยโปรแกรมนี้	16
รูปที่ 3-9	แสดงหน้าจอของฟอร์มแบบต่างๆที่มีให้เลือก	19
รูปที่ 3-10	แสดงหน้าจอตัวอย่างของฟอร์มที่ใช้โปรแกรมนี้สร้าง	20
รูปที่ 3-11	แสดงหน้าจอที่แสดงข้อความที่ได้กรอกเข้าไปในฟอร์ม	20
รูปที่ 3-12	แสดงหน้าจอหลักของกระดานข่าว	23
รูปที่ 3-13	E-R Diagram ของกระดานข่าว	24
รูปที่ 3-14	แสดง Flow Chart การทำงานของผู้ใช้งานกระดานข่าวทุกๆไป	25
รูปที่ 3-15	แสดง Flow Chart การทำงานของผู้ดูแลกระดานข่าว	26
รูปที่ 3-16	แสดงหน้าจอการเข้าระบบของกระดานข่าว	27
รูปที่ 3-17	แสดงหน้าจอการสมัครสมาชิกของกระดานข่าว	29
รูปที่ 3-18	แสดงหน้าจอการแก้ไขข้อมูลส่วนตัว	30
รูปที่ 3-19	แสดงหน้าจอเพื่อให้ผู้ใช้ยืนยันการตั้งกระทู้อีกครั้ง	31
รูปที่ 3-20	แสดงหน้าจอการตอบกระทู้ของแต่ละหัวข้อกระทู้	33
รูปที่ 3-21	แสดงหน้าจอการค้นหาข้อมูลของกระดานข่าว	34
รูปที่ 3-22	แสดงหน้าจอการลบกระทู้	36
รูปที่ 3-23	แสดงหน้าจอการลบห้องกระทู้	37
รูปที่ 3-24	แสดงหน้าจอการแก้ไขชื่อห้องกระทู้	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-25	แสดงหน้าจอการลบผู้ใช้ออกจากสมาชิก	39
รูปที่ 3-26	แสดงห้องกระทู้และกระทู้ที่อยู่ในห้องกระทู้นั้น	40
รูปที่ 3-27	แสดงหน้าจอการเลือกรูปแบบพื้นหลังของโฮมเพจ	42
รูปที่ 3-28	แสดงหน้าจอตัวอย่างของโฮมเพจส่วนตัวของผู้ใช้	44
รูปที่ 5-1	โปรแกรม CGI ในรูป sub parseform	66
รูปที่ 5-2	รูปแสดงผลการรันโปรแกรม	77
รูปที่ 5-3	รูปแสดงผลการทำงานของโคดโปรแกรมในข้างต้น	78
รูปที่ 5-4	ภาพแสดงความกว้างและความสูงของมุมมอง	78
รูปที่ 5-5	แสดงผลจากการรันบนโปรแกรม	79
รูปที่ 5-6	รูปแสดงผลการรันสร้างรูปสี่เหลี่ยมแบบ 3D	80
รูปที่ 5-7	ผลของการรันโปรแกรมสร้างวงกลม	81
รูปที่ 5-8	ผลการรันโปรแกรมแสดง	82
รูปที่ 5-9	จากการรันโปรแกรม	83

สารบัญตาราง

	หน้าที่	
ตารางที่ 2-1	แสดงความแตกต่างของตัวแปรภาษาทั้งสองประเภท	8
ตารางที่ 5-1	แสดงเครื่องหมายเปรียบเทียบระหว่างตัวแปรและข้อความ	56



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันความก้าวหน้าในเทคโนโลยีทางด้านอินเทอร์เน็ตค่อนข้างสูง ทั้งในด้านความเร็วในการทำงานที่มากขึ้นและประสิทธิภาพในการทำงานที่สูงขึ้น อีกทั้งกลุ่มผู้ใช้งานค่อยๆ ขยายตัวกว้างขึ้นเรื่อยๆ เนื่องมาจากประโยชน์ของแหล่งข้อมูล, ความรู้, ความบันเทิง และการติดต่อสื่อสารในโลกที่ไร้พรมแดน โดยวิธีการสำหรับการเชื่อมต่อหรือติดต่อสื่อสารของเครือข่ายอินเทอร์เน็ต เรียกว่า โพรโตคอล (Protocol) โดยโพรโตคอลที่เป็นพื้นฐานสำหรับการเชื่อมโยงสื่อสารของเครือข่ายอินเทอร์เน็ตใช้ TCP/IP (Transmission Control Protocol / Internet Protocol) โพรโตคอลนี้ถือเป็นโพรโตคอลมาตรฐานในการกำหนดรายละเอียดการทำงาน ทำให้สามารถเชื่อมโยงคอมพิวเตอร์ที่มีความแตกต่างกันได้

ระบบเครือข่ายเวิร์ดไวด์เว็บ เริ่มด้วยข้อมูลที่มีลักษณะ Interactive Hypermedia หรือกล่าวอีกอย่างว่า รูปแบบหรือเอกสารที่ใช้งานที่เรียกว่า ไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นเอกสารที่สามารถ เชื่อมโยงกับเอกสารต่างๆที่มีความสัมพันธ์กัน โดยภาษาที่ใช้เป็นข้อกำหนดในการสร้างเอกสารรูปนี้คือภาษา HTML (Hypertext Markup Language) ภาษา HTML มีการกำหนดส่วนที่เรียกว่ามาร์คอัพ (Markup) หรือ จุดที่จะเชื่อมโยง ส่วนเอกสารต่างๆไปยังเอกสาร หรือแหล่งข้อมูลอื่นๆ และวิธีการหรือรายละเอียด ข้อกำหนดในการรับ-ส่ง ข้อมูลของระบบเว็บ จะอาศัยโพรโตคอล HTML (Hypertext Transfer Protocol) และตัวรูปแบบข้อมูลจะเรียกว่า ไฮเปอร์มีเดีย (Hypermedia) ทั้งนี้เพราะข้อมูลมีความหลากหลายรูปแบบ การใช้งานของตัวข้อมูล ไม่ว่าจะเป็น เท็กซ์ (Text), กราฟฟิกหรือรูปภาพ (Graphics, Image), เสียง (Audio), วิดีโอ (Video) และอื่นๆ การแสดงผลหากเป็นการใช้งานแบบเดิมหรือในยุคแรกๆของอินเทอร์เน็ต จะแสดงผลเป็นเท็กซ์อย่างเดียว โดยใช้ Lynx ซึ่งเป็นคำสั่งใช้งานบนระบบปฏิบัติการยูนิกซ์ (UNIX) เป็นตัวค้นหาข้อมูลหรือเอกสาร (แบบเดียวกับกับบราวเซอร์) แต่ปัจจุบันสามารถใช้งานผ่านโปรแกรมจำพวกบราวเซอร์ (Browser) ที่มีขีดความสามารถสูงทำให้สามารถใช้งาน ได้กับข้อมูลที่มีรูปแบบหลากหลายได้ โดยการจัดรูปแบบการนำเสนอยังต้องอาศัยภาษา HTML ในการกำหนดและสร้างเอกสาร

เมื่อเกิดระบบเครือข่ายเวิร์ดไวด์เว็บใช้งานจนเป็นที่นิยมก็มีการพัฒนาแอปพลิเคชันในลักษณะเหมือนโปรแกรมประยุกต์ที่องค์กรใช้งานอยู่โดยอาศัยหลักการของ CGI ในการพัฒนาซึ่ง CGI (Common Gateway Interface) เป็นหลักการหรือวิธีการของการพัฒนาแอปพลิเคชัน ที่ทำหน้าที่เหมือนประตู (Gateway) เชื่อมโยงการติดต่อกับการทำงานอื่นๆเพื่อให้เกิดการทำงานที่หลากหลายในการใช้งาน โดยอาศัยพื้นฐานของระบบเว็บหรือจะกล่าวได้ว่า การทำงานควบคู่กับเว็บเซิร์ฟเวอร์ เพราะบราวเซอร์ไม่สามารถติดต่อส่วนอื่นๆโดยตรงได้ เช่นการติดต่อกับฐานข้อมูลเป็นต้น จำเป็นต้องติดต่อผ่านเซิร์ฟเวอร์ไปยังส่งของ CGI ดังนั้นเรามักจะเรียกว่า CGI โปรแกรม ด้วยเหตุนี้จึงเห็นได้ว่า CGI เป็นแอปพลิเคชันประเภท เซิร์ฟเวอร์ แอปพลิเคชัน (Server Application) หรือแอปพลิเคชันที่ทำงานอยู่ที่ฝั่งเซิร์ฟเวอร์ โดยมีส่วนที่ทำหน้าที่ติดต่อกับ ผู้ขอใช้บริการ หรือ ไคลเอ็นต์ (Client) คือเว็บเซิร์ฟเวอร์ และไคลเอ็นต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านเว็บเซิร์ฟเวอร์โดยตรง พอมีวิธีการของ CGI ทำให้เห็นได้ว่าจากรูปแบบของโฮมเพจที่เปลี่ยนไปจากเดิมเคยเป็นแค่ Static Hypermedia Document ซึ่งเป็นเอกสารที่แสดงโดยไม่มี การเปลี่ยนแปลง ไปเป็นเอกสารที่สามารถเปลี่ยนแปลงรูปแบบได้ ตลอดจนเห็นเป็นโฮมเพจที่สามารถโต้ตอบได้ หรือเป็น อินเตอร์แอคทีฟ (Interactive) เหมือนส่วนของอินเตอร์เฟส (Interface) ของ CGI แอปพลิเคชัน ที่แปรเปลี่ยนตลอดเหมือนการใช้งาน โปรแกรมประยุกต์นั่นเอง.

เครื่องมือสร้างและแก้ไขข้อมูลโฮมเพจอย่างง่ายบนเซิร์ฟเวอร์ เป็นเครื่องมือที่พัฒนาขึ้นมาจากการเขียนโปรแกรมภาษา Perl บน CGI ซึ่งทำงานบน Server โดยเครื่องมือที่เขียนขึ้นมาี้มีความสามารถในการสร้าง, แก้ไขและอัปเดตข้อมูลประเภทต่างๆบนเว็บไซต์ เพื่อให้การทำงานบนเว็บไซต์นั้นเป็นไปอย่างสะดวกสบายและมีประสิทธิภาพมากขึ้น และได้สร้างเว็บบอร์ด ซึ่งเป็นกระดานข่าวโดยผู้ที่เข้ามาใช้งานสามารถที่จะดูข้อมูลที่สนใจ หรือสร้างหัวข้อในการติดต่อสื่อสารกันได้และรอการตอบรับจากผู้ที่สนใจในหัวข้อนั้นๆ และผู้ดูแลระบบ สามารถที่จะจัดการกับระบบได้โดยที่ไม่ต้องเข้าเขียน โปรแกรมหรือจัดการในส่วนของไฟล์เพิ่มขึ้น แต่จัดการบนหน้าจอของเว็บไซต์ได้เลย ซึ่งเครื่องมือที่สร้างขึ้นมา ทั้งเครื่องมือสร้างและแก้ไขข้อมูลโฮมเพจอย่างง่ายบนเซิร์ฟเวอร์กับเว็บบอร์ด จะสร้างขึ้นโดยเน้นการทำงานที่สะดวกขึ้นโดยที่ผู้ใช้งานสามารถจัดการการทำงานต่างๆได้บนเว็บไซต์ โดยไม่ต้องใช้ความรู้เกี่ยวกับโปรแกรมมากมายนัก และไม่ต้องยุ่งเกี่ยวกับส่วนของฐานข้อมูลเลย ซึ่งเครื่องมือต่างๆบนเว็บไซต์จะทำงานโดยเข้าไปจัดการในส่วนต่างๆ แทนให้ และมีอุปกรณ์อำนวยความสะดวกอื่นๆเพิ่มขึ้นมา เช่น ตารางเวลานัดหมายของผู้ใช้งาน, การเตือน, การตรวจสอบและป้องกันข้อมูลที่ไม่สุภาพและอื่นๆ ซึ่งอุปกรณ์เหล่านี้สร้างขึ้นมาเพื่อเพิ่มประสิทธิภาพในการทำงานบนเว็บไซต์.

จากที่กล่าวมาทั้งหมดนี้ ผู้อ่านคงจะเห็นได้ว่า เครื่องมือต่างๆที่พัฒนาขึ้นมานั้นสามารถเพิ่มความสะดวกสบายให้กับการทำงานบนเว็บไซต์ให้มากยิ่งขึ้น เพื่อให้ผู้ใช้งานไม่จำเป็นจะต้องมีความรู้ในเรื่องของระบบเซิร์ฟเวอร์ การเขียน โปรแกรม CGI และการทำงานของฐานข้อมูล ซึ่งเครื่องมือต่างๆจะช่วยจัดการระบบต่างๆให้อยู่แล้ว และสามารถนำเครื่องมือเหล่านี้ไปประยุกต์ใช้งานกับการทำงานต่างๆได้เพื่อประโยชน์ในการพัฒนาต่อไปในอนาคต.

1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 สร้างอุปกรณ์ต่างๆบนเว็บไซต์ด้วยภาษา CGI ขึ้นมาเพื่ออำนวยความสะดวกและเพิ่มประสิทธิภาพในการทำงานให้มากขึ้น โดยผู้ใช้สามารถเข้ามาใช้งานบนเว็บไซต์ได้โดยไม่ต้องมีความรู้เกี่ยวกับภาษาที่ใช้เขียนมากนัก และไม่ต้องเข้าไปยุ่งเกี่ยวกับส่วนของระบบและฐานข้อมูลโดยตรง แต่ใช้งานผ่านบราวเซอร์แทน
- 1.2.2 ศึกษาความเป็นไปได้ในการนำเอาภาษา CGI มาพัฒนารูปแบบของเว็บไซต์ ว่าสามารถพัฒนาไปในรูปแบบต่างๆ ได้รูปแบบใดบ้าง
- 1.2.3 เครื่องมือและอุปกรณ์ต่างๆที่สร้างขึ้นมา สามารถถูกพัฒนาให้มีประสิทธิภาพมากขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะสร้างเครื่องมือขึ้นมาโดยใช้ภาษา Perl เขียนบนโปรแกรม CGI ซึ่งเครื่องมือที่ถูกสร้างขึ้นจะถูกเก็บอยู่บนเซิร์ฟเวอร์ ใช้ Apache Server เป็นเซิร์ฟเวอร์ มี Linux เป็นระบบปฏิบัติการ การเก็บข้อมูลจะเก็บลงในส่วน Database โดยใช้ MySQL ในการเก็บ

เครื่องมือต่างๆที่สร้างขึ้นมา สามารถทำงานได้โดยใช้บราวเซอร์เรียกทำงาน (Internet Explorer และ Netscape Navigator) โดยไฟล์ต่างๆที่เขียนขึ้นมาจะอยู่ในรูปของ CGI ทั้งหมด

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งจะมีเรื่องหลักๆคือ หลักการทำงานของเว็บไซต์, หลักการทำงานของ CGI-Perl, ระบบปฏิบัติการ Linux, เว็บเซิร์ฟเวอร์ Apache, ฐานข้อมูล MySQL ซึ่งมีรายละเอียดดังในบทที่ 2 โดยแยกอธิบายออกเป็นหัวข้อต่างๆ จากนั้นก็จะนำเอาความรู้ที่ได้ศึกษาทั้งหมดมาออกแบบเครื่องมือสร้างและแก้ไขข้อมูลโสมเพจอย่างง่ายบนเซิร์ฟเวอร์ ซึ่งมีรายละเอียดของเครื่องมือในบทที่ 3 โดยจะมีเครื่องมือหลักๆอยู่ 3 แบบคือ เครื่องมือ Update Data, เครื่องมือสร้างฟอร์ม, โปรแกรม Webboard. โดยจะแสดงรูปแบบของโปรแกรมที่ถูกสร้างขึ้น และการทำงานของไฟล์ต่างๆที่เกี่ยวข้องกัน รวมถึงระบบที่ใช้งานอยู่ด้วย

บทที่ 4 จะเป็นการทดสอบระบบรวมทั้งหมด บทที่ 5 จะเป็นการสรุปการทำงาน ผลที่ได้รับจากงานวิจัยนี้ และแนวทางในการนำไปประยุกต์ใช้ บทที่ 6 จะเป็นภาคผนวก ซึ่งจะอธิบายถึงการติดตั้งโปรแกรมต่างๆ และวิธีการใช้งานขั้นพื้นฐาน

บทที่ 2

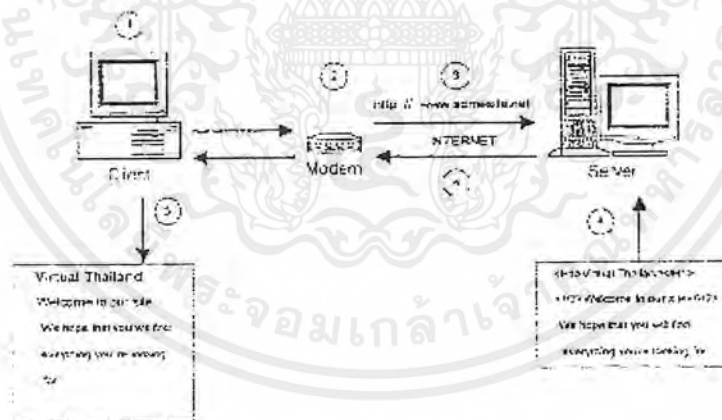
ทฤษฎีและหลักการ

2.1 ระบบที่ใช้ในการทำงาน

โปรแกรม (Program)	=	CGI-PERL, JAVA SCRIPT
ระบบปฏิบัติการ (Operating System)	=	LINUX SERVER
เว็บเซิร์ฟเวอร์ (Web Server)	=	APACHE WEB SERVER
ฐานข้อมูล (Database)	=	MYSQL

2.2 หลักการทำงานของเว็บไซต์

บริการเวิร์ดไวด์เว็บจะใช้สถาปัตยกรรมแบบไคลเอ็นต์เซิร์ฟเวอร์ (Client / Server) โดยเว็บเบราว์เซอร์ (Web Browser) ที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ฝั่งไคลเอ็นต์ (Client) จะส่งการร้องขอข้อมูลที่อยู่ในฝั่งของเครื่องเซิร์ฟเวอร์ (Server) โดยเซิร์ฟเวอร์ก็จะนำข้อมูลที่ได้ออกมา ส่งผ่านเครือข่ายไปยังเครื่องของผู้รับ แล้วเว็บเบราว์เซอร์ของผู้รับก็จะทำการแปลและแสดงข้อมูลนั้นออกทางหน้าจอ



รูปที่ 2-1 การส่งข้อมูลระหว่างเครื่องไคลเอ็นต์และเซิร์ฟเวอร์

2.2.1 ขั้นตอนแสดงถึงการทำงานข้างต้น

2.2.1.1 ผู้รับเริ่มการทำงานของโปรแกรมเว็บเบราว์เซอร์บนเครื่องของผู้รับ

2.2.1.2 ผู้รับทำการเชื่อมต่อเข้าสู่อินเทอร์เน็ตซึ่งอาจจะเป็นที่บ้าน, ที่ทำงาน โดยผ่านบริษัทผู้ให้บริการอินเทอร์เน็ตหรือเป็นการเชื่อมต่อโดยตรงก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.2.1.3 ผู้รับร้องขอข้อมูลต่างๆจากไชต์บนเว็บไซต์โดยบราวเซอร์จะส่งข้อความร้องขอไปผ่าน
ไปบน Internet ด้วยข้อความดังนี้
- โพรโตคอลที่ใช้ในการส่งถ่ายข้อมูล ได้แก่ Http://
 - ตามด้วยตำแหน่งบนหน่วยความจำ (Address) หรือ URL (Uniform Resource Locator)
- 2.2.1.4 เมื่อเซิร์ฟเวอร์ได้รับข้อความร้องขอแล้ว ก็จะทำการส่งเว็บเพจที่ร้องขอผ่านไบน
อินเทอร์เน็ตไปยังเครื่องคอมพิวเตอร์ ของผู้รับ
- 2.2.1.5 โปรแกรม Web Browser ก็จะได้รับข้อมูลนั้น และทำการแสดงผลข้อมูลให้ผู้รับเห็น

2.2.2 เครื่องคอมพิวเตอร์ฝั่งไคลเอ็น

ผู้ใช้ที่ต้องการใช้งานบนเว็บไซต์จะต้องมีคอมพิวเตอร์แบบไคลเอ็น ซึ่งต้องการคุณสมบัติในการ
เชื่อมต่อสู่อินเทอร์เน็ตและต้องสามารถใช้งาน โปรแกรมเว็บเบราว์เซอร์ได้

การเชื่อมต่อเข้าสู่อินเทอร์เน็ต อาจทำได้แบบฮาร์ดไวร์ (Hard-Wired) หรือแบบใช้โมเด็ม
(Modem) ผ่านบริษัทที่ให้บริการอินเทอร์เน็ตก็ได้ ซึ่งข้อแตกต่างของทั้งสองอย่างก็เป็นเพียงเรื่องของ
ความเร็วนอกจากนั้นก็มีการทำงานเหมือนกัน

ปัจจุบันมีโปรแกรมเว็บเบราว์เซอร์สำหรับ คอมพิวเตอร์ในรูปแบบ (Platform) ต่างๆอยู่มากมาย
เช่น เน็ตเคปเนวิกเตอร์ (Netscape Navigator) สำหรับ วินโดว (Windows), WebSurfer สำหรับ OS2,
MacWeb ของแมคอินทอช

2.2.3 เครื่องคอมพิวเตอร์ฝั่งเซิร์ฟเวอร์

เครื่องคอมพิวเตอร์แบบเซิร์ฟเวอร์จะต้องมีคุณสมบัติเช่นเดียวกับเครื่องคอมพิวเตอร์ฝั่งไคลเอ็น
นั่นคือ ต้องการการเชื่อมต่อทางอินเทอร์เน็ต และต้องสามารถเรียกการใช้งาน โปรแกรมเว็บเบราว์เซอร์ได้

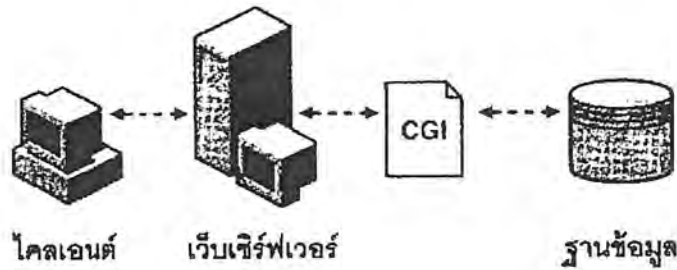
อย่างไรก็ตามเว็บเซิร์ฟเวอร์ต้องการการเชื่อมต่อทางอินเทอร์เน็ตที่มีความสามารถสูง โดย
เซิร์ฟเวอร์ควรจะทำาการเชื่อมต่อเข้าสู่อินเทอร์เน็ตได้ตลอดเวลา มิฉะนั้นผู้ใช้จะไม่สามารถเข้าสู่เว็บไซต์
นั้นๆ ได้

2.3 หลักการทำงานของซีจีไอ

CGI ย่อมาจาก Common Gateway Interface เป็นเสมือนตัวกลางในการเชื่อมต่อระหว่างผู้ใช้กับ
เซิร์ฟเวอร์ โดยทั่วไปแล้วการเชื่อมต่อแอปพลิเคชันกับข้อมูลที่ HTTP Server มีอยู่ 2 ลักษณะคือ แบบสแต
ติก (Static) และแบบไดนามิกซ์ (Dynamic)

- แบบ สแตติกเป็นการอ้างถึงเว็บเพจ ธรรมดาจะ ได้ข้อมูลที่คงที่ไม่เปลี่ยนแปลง
- แบบ ไดนามิกเป็นการอ้างถึงโปรแกรมซีจีไอ ซึ่งต้องมีการทำงานของโปรแกรมแบบเวลาใน
ขณะนั้น (Real Time) ซึ่งจะทำได้ผลลัพธ์ที่มีการเปลี่ยนแปลงไปตามสภาวะปัจจุบัน เช่น การเรียกดูข้อ
มูลจากฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 เป็นการขนถ่ายข้อมูลในส่วนของโปรแกรมซีจีไอ

โดยทั่วไปแล้วโปรแกรมซีจีไอที่สร้างไว้จะถูกเก็บในไดเรกทอรี (Directory) พิเศษซึ่งอยู่ภายใต้การดูแลของเว็บมาสเตอร์การเขียนโปรแกรม CGI สามารถเขียนได้ 2 ลักษณะคือ

- เขียน โดยภาษาที่ใช้เขียน สคริป (Script) หรือที่เรียกว่า Scripting Language เช่น Perl, Unix, Shell Script, TCL ซึ่งสามารถนำไปใช้ได้เลยโดยไม่ต้องมีข้อมูลฐาน (Source Code) เก็บแยกต่างหากทำให้ง่ายและสะดวกต่อการแก้ไขในภายหลัง
- เขียน โดยภาษาโปรแกรม (Programming Language) เช่น ภาษา C / C++ , Fortan, VB ซึ่งการเขียนด้วยภาษาเหล่านี้ เราต้องทำการประมวลผลข้อมูลฐาน (Compile Source Code) ก่อนจึงจะสามารถนำไปใช้งานได้ส่วน Source Code นั้น เราจะต้องเก็บไว้ในไดเรกทอรี ที่มีชื่อว่า /cgi-src เมื่อเราทำการเปลี่ยนแปลงโปรแกรม เราต้องทำการประมวลผล (Compile) ใหม่ทุกครั้ง ซึ่งอาจทำให้เสียเวลาเพราะฉะนั้นจึงเป็นที่นิยมน้อยกว่าแบบแรก

2.3.1 การทำงานของโปรแกรมซีจีไอ

เมื่อผู้ใชร้องขอความต้องการขณะที่อยู่ที่บราวเซอร์ตัวเซิร์ฟเวอร์จะทำการประมวลผลโปรแกรมซีจีไอ แบบ Real-Time แล้วส่งผลลัพธ์ที่ได้ซึ่งอาจเป็นการเรียกดูข้อมูลที่อยู่เซิร์ฟเวอร์ผลลัพธ์ที่ได้เหล่านี้จะถูกส่งกลับจากเซิร์ฟเวอร์ไปที่ฝั่งผู้ใช้ โดยผ่านทางโปรแกรมซีจีไอซึ่งต้องมีการกำหนดรูปแบบของผลลัพธ์ ที่จะส่งกลับเสียก่อนในรูปของการกำหนดรูปแบบของเนื้อหา (Content - Type) ที่ส่วนหัวของโปรแกรมซีจีไอและต้องตามด้วยบรรทัดว่าง 1 บรรทัด ก่อนที่จะเริ่มส่วนของ ผลลัพธ์จริงๆ เช่น

เป็นการกำหนดว่าผลลัพธ์ที่จะส่งกลับมา เป็นข้อมูลในรูปแบบของเอกสาร HTML เราเรียกส่วนนี้ว่า MIME Type ส่วนใน HTML ที่เรียกใช้โปรแกรมซีจีไอต้องมีการกำหนด <FORM ACTION = ... METHOD = ...> ในส่วนของ ACTION คือเส้นทาง (Path) ที่อยู่ของโปรแกรมซีจีไอ ซึ่งต้องการให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ไปทำการประมวลผล และในส่วนของ METHOD เป็นการกำหนดวิธีการในการส่งข้อมูล ซึ่งมี 2 แบบ คือ “GET “ และ “ POST “

```

#!/usr/bin/perl
use Socket;

my $port = 8080;
my $proto = "tcp";

my ($rhost, $raddr, $rproto) = getpeerbyaddr(0);
my ($laddr, $lproto, $lport) = getsockbyname($port);

print "Listening on port $port\n";

socket(SERVERSOCK, AF_INET, SOCK_STREAM, $proto) || die "Socket error: $!";
my $saddr = pack("S", AF_INET, $port);

bind(SERVERSOCK, $saddr) || die "Bind error: $!";
listen(SERVERSOCK, 5) || die "Listen error: $!";

while(1) {
    print "Waiting for connection\n";

```

รูปที่ 2-3 โปรแกรมซีจีโอบนระบบปฏิบัติการ

- วิธีแบบ “GET “ จะทำการส่งข้อมูลเข้า (Input) ผ่านทางตัวแปรภายนอก (Environment Variable) ที่ชื่อ Query_String ต่อท้าย URL ซึ่งอยู่ในรูปของกลุ่มชื่อและค่า name–value และอาจมีได้หลายคู่ โดยมีเครื่องหมาย & เป็นตัวแยกระหว่างคู่ นั่นคือ

URL ? name1 = value1 & name2 = value2 & name3 = value3

แต่การส่งข้อมูลแบบนี้มีจำนวนจำกัดทำให้ส่งได้น้อย

- วิธีแบบ “ POST “ ข้อมูลเข้าจะถูกส่งผ่านทาง Standart Input ซึ่งต้องส่ง Content_Length เพื่อบอกจุดสิ้นสุดของข้อมูลที่ส่งด้วยการส่งวิธีนี้จะทำให้สามารถส่งข้อมูล ได้เป็นจำนวนมากกว่าแบบแรก

2.4 หลักการทำงานของ Perl

2.4.1 Perl

ย่อมาจาก Practical Extraction and Reporting Language เป็นภาษาหนึ่งที่ถูกบัญญัติขึ้นโดยนาย Larry Wall ในปี 1986 ซึ่งเริ่มใช้สำหรับเขียนโปรแกรมในระบบยูนิกซ์ ก่อนที่จะถูกพัฒนาให้ใช้กับระบบปฏิบัติการอื่น ๆ ได้ เช่น WindowsNT Perl เป็นโปรแกรมภาษาที่เขียนอยู่ในรูปของสคริปต์ (Scripts เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Langeage) และเก็บอยู่ในรูปของเท็กซ์ไฟล์ จึงทำให้การแก้ไขและการทำความเข้าใจได้ง่ายกว่า ภายในไฟล์โปรแกรมนั้นจะประกอบไปด้วยคำสั่งของ Perl ซึ่งลักษณะของคำสั่งและรูปแบบการเขียนนั้น จะคล้าย ๆ กับภาษาซี แต่มีรูปแบบที่ง่ายกว่าและมีความยืดหยุ่นภายในตัวภาษาเอง Perl ถูกออกแบบมาเพื่อรองรับกับการเขียนโปรแกรมบนเว็บ (Web programming) ซึ่งใช้สำหรับสร้าง CGI (Common Gateway Interface) สำหรับจัดการข้อมูลให้กับ Web เช่น การกรอกแบบสอบถาม (Form) หรือ การสร้างเว็บเพจใหม่ (Dynamic Webpage) เป็นต้น

2.4.2 ตัวแปลภาษาของ Perl

เมื่อเราศึกษาภาษาใด เราก็มักจะรู้ด้วยว่าภาษานั้น ใช้ตัวแปลภาษาแบบใด ซึ่งโดยทั่ว ๆ ไปแล้วเราจะแบ่งออกเป็น 2 ประเภท นั่นก็คือ อินเตอร์พรีเตอร์ (Interpreter) และ คอมไพเลอร์ (Compiler) ความ

<p>1. อินเตอร์พรีเตอร์ การทำงานของตัวแปลภาษาแบบนี้ จะทำอ่านซอร์สโค้ดโปรแกรมทีละบรรทัด การประมวลผลเป็นบรรทัดไป หากเกิดข้อผิดพลาด ตัวแปลภาษาจะหยุดประมวลผลและแสดงข้อผิดพลาดทันที... ภาษาที่ใช้ตัวแปลแบบนี้ก็เช่น ภาษาเบสิก</p>
<p>2. คอมไพเลอร์ การทำงานของตัวแปลภาษาแบบคอมไพเลอร์ จะเริ่มประมวลผลโดยอ่าน และตรวจเช็คข้อผิดพลาด ภายในซอร์สโค้ดโปรแกรมทั้งหมดและทำการแปลงให้อยู่ในรูปของไฟล์เอ็กซิกิวต์ (Execute) จากนั้นจึงทำการรัน ตัวอย่างภาษาที่ใช้ตัวแปลแบบนี้ก็คือ ภาษาซี</p>

ตารางที่ 2-1 แสดงแตกต่างของตัวแปลภาษาทั้งสองประเภท

สำหรับตัวแปลภาษาของ Perl นั้น อยู่ระหว่างความเป็น อินเตอร์พรีเตอร์ และ คอมไพเลอร์ เพราะเมื่อคุณรันโปรแกรมหรือสคริปต์ ตัวแปลภาษาของ Perl จะคอมไพล์ซอร์สโค้ดโปรแกรมให้อยู่ในรูปของไฟล์เอ็กซิกิวต์และเก็บไว้ในหน่วยความจำ จากนั้นจึงทำการรันโปรแกรม ทำการให้ประมวลผลนั้นมีความรวดเร็วยิ่งขึ้น เพราะไม่มีการเขียนไฟล์เอ็กซิกิวต์ลงบนดิสก์ เหมือนกับคอมไพเลอร์ภาษาซี

2.4.3 เลือกใช้งานภาษา Perl เนื่องจากข้อดีต่างๆดังนี้

- Perl ง่ายต่อการศึกษา เพราะโครงสร้างของภาษาไม่ซับซ้อน
- ตัวภาษามีความยืดหยุ่นสูง (ที่มีลักษณะเหมือนกันภาษาซี)
- การเขียน Perl จะเขียนอยู่ในรูปของเท็กซ์ไฟล์ โดยใช้โปรแกรม NotePad (บน Windows) หรือใช้โปรแกรม Vi
- การประมวลผลหรือรันโปรแกรมที่เขียนด้วย Perl ไม่จำเป็นต้อง Compile และ Link ให้เป็น Machine-Code เช่นเดียวกับภาษาซี เพราะการรันโปรแกรมจะต้องใช้ตัวแปลภาษาของ Perl เป็นตัวประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 หลักการทำงานของลินุกซ์ (LINUX)

เป็นระบบปฏิบัติการแบบ 32 บิต ที่พัฒนาขึ้นมาจากระบบยูนิกซ์ (UNIX) สนับสนุนการทำงานแบบ Multitasking, Multi – User และ Multiprocessor เป็นระบบติดต่อกับผู้ใช้แบบกราฟฟิก (Graphic) ใช้มาตรฐานการสื่อสารแบบ TCP/IP บรรดาผู้ใช้งานบนลินุกซ์นั้นมีได้หลายระดับ ไม่ว่าจะเป็นระดับ Kernel Hacker ซึ่งจะทำการศึกษาเกี่ยวกับการทำงานแบบลึกถึง Kernel ซึ่งผู้ใช้สามารถที่จะใช้ Linux ได้หลายอย่าง ไม่ว่าจะเป็นการศึกษาระบบ Unix หรือใช้เขียนโปรแกรม หากต้องการที่จะใช้ Application บน Dos กับ Windows ก็มี DosEmu และ WINE ไว้ให้สามารถใช้ได้

ในการสื่อสารกับอินเทอร์เน็ตสามารถทำ BBS ส่วนตัว, ทำระบบงาน Back Office, Mail Server, File Server, Internet Server, Web Server ก็ได้ ในการทำงานนี้เราจะใช้โปรแกรม Apache Web Server ในการทำงาน และตั้ง Server ให้สำหรับการทำงานขึ้นมา

2.5.1 เลือกใช้งานระบบปฏิบัติการ Linux เนื่องจากข้อดีต่าง ๆ ดังนี้

- Linux เป็นระบบปฏิบัติการแบบหลายงานและหลายผู้ใช้ (Multitasking and Multiuser) ที่สมบูรณ์แบบ
- Linux มีความเข้ากันได้ (Compatible) กับระบบ UNIX ส่วนมากในระดับซอร์สโค้ด
- ความสามารถอื่นๆ อันได้แก่ POSIX job control (ซึ่งถูกใช้ในโปรแกรม shell ต่างๆ) เทอร์มินอลเสมือน คอนโซลเสมือน
- Linux สนับสนุนระบบไฟล์หลายชนิด
- Linux สนับสนุนระบบเครือข่าย TCP/IP อย่างสมบูรณ์
- เคอร์เนลของ Linux มีความสามารถในการจำลองการทำงานของโปรเซสเซอร์ช่วยประมวลผลทางคณิตศาสตร์ 80387
- สนับสนุนการดีบั๊กโปรแกรม

2.6 หลักการทำงานของอะปาเช (APACHE)

Apache นั้นเป็นระบบ Server ที่ได้รับความนิยมสูงสุดระบบหนึ่งเมื่อเทียบกับโปรแกรมบน Server ตัวอื่นๆ แล้ว

การติดตั้ง Apache นั้น เป็นการติดตั้งในลักษณะของการคอมไพล์ซอร์สโค้ด เพื่อให้ได้ตัวไบนารีโค้ด หรือรหัสโปรแกรมที่สามารถทำงานได้ แต่ถ้าต้องการหาในแบบของ Package ของ Apache ที่อยู่ในรูปของ RPM (Redhat Package Management) มาเพิ่มเติมในการติดตั้งสามารถทำได้สะดวกและง่ายขึ้น การติดตั้งในรูปแบบของ Package มีข้อดีก็คือ ไม่จำเป็นจะต้องยุ่งยากในการกำหนดคอนฟิกของโปรแกรม และการคอมไพล์ Source Code ทั้งหมด แต่อาจจะมีปัญหาอยู่บ้างในกรณี ที่ระบบ Linux ของเรามีข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดในการติดตั้ง เช่นจะต้องมี Library ของโปรแกรมที่ใช้งานหรือมีโปรแกรมที่มีการอ้างอิงในการใช้งาน ดังนั้นหากจะติดตั้ง Apache ในรูปแบบของ Package RPM บนระบบ Linux ให้ใช้คำสั่งในการติดตั้งคือ

```
$ rpm -I apache - 1.2.4-4.i386.rpm
```

ระบบไฟล์ที่ใช้งานใน Apache นั้น โดยปรกติแล้ว Apache จะติดตั้งไฟล์ที่ใช้ในการ Setup ตัวโปรแกรม Apache และไฟล์ที่จำเป็นในการใช้งานอื่นๆไว้ในไดเรกทอรี /usr/local/etc/httpd ซึ่งเรียกได้ว่าเป็น Server Root ก็ว่าได้

ไฟล์ที่ใช้กำหนดข้อมูลทั่วไปของ Apache	:	httpd.conf
ไฟล์ที่ใช้กำหนดข้อมูลที่เกี่ยวข้องกับชื่อต่างๆ	:	srn.conf
ไฟล์ที่ใช้กำหนดเรื่องของการเข้าถึงข้อมูลในระดับต่างๆ	:	access.conf
ไฟล์ที่ใช้กำหนดเกี่ยวกับเรื่อง MIME	:	mime.types
ไฟล์ที่เก็บข้อมูลเกี่ยวกับข้อผิดพลาด	:	error_log
ไฟล์ที่เก็บข้อมูลว่ามีการดึงข้อมูลจากไฟล์ใดบ้าง	:	access_log
ไฟล์ที่เก็บหมายเลขไพรเซสของโปรแกรมเว็บเซิร์ฟเวอร์	:	httpd.pid

2.7 หลักการทำงานของมายเอสคิวแอล (MySQL)

MYSQL เป็นภาษาในการจัดการกับฐานข้อมูลซึ่งมีโครงสร้างของภาษาใกล้เคียงกับภาษามนุษย์ จึงนับว่าเป็นภาษาในลักษณะชั้นสูง (High-Level Language) ระบบจัดการฐานข้อมูล ในปัจจุบันจะสนับสนุนการใช้งานกับฐานข้อมูลโดยใช้ภาษา SQL แทบทั้งสิ้น ดังนั้นในการพัฒนาโปรแกรมต่าง ๆ นั้น ระบบจะเป็นระบบเปิดในส่วนของงานจัดการฐานข้อมูล ซึ่งจะให้ประโยชน์ในหลายๆด้าน เช่น การลดค่าใช้จ่ายในการพัฒนาระบบ หรือการปรับปรุงโปรแกรม, ง่ายต่อการบำรุงรักษาระบบรวม

2.7.1 เลือกใช้งานระบบฐานข้อมูล MySQL เนื่องจากข้อดีต่างๆดังนี้

- ลักษณะโครงสร้างเข้าใจง่าย
- มีทฤษฎีทางคณิตศาสตร์รองรับ
- การจัดการเรื่อง Concurrency Control หรือ การใช้งานกับฐานข้อมูลเดี่ยวพร้อมๆกัน
- การจัดการเรื่องการป้องกัน และกู้ระบบฐานข้อมูล
- การจัดการเรื่อง Query Optimization
- การจัดการเรื่องความปลอดภัย

การพัฒนา CGI ติดต่อกับฐานข้อมูลนั้น เราต้องหาเครื่องมือช่วยในการใช้งานฐานข้อมูลกับโปรแกรมภาษาเพิร์ล ซึ่งจะอาศัยการติดต่อผ่าน DBI (Database Interface) ซึ่งจะทำหน้าที่เสมือน ตัวเชื่อมการติดต่อระหว่างโปรแกรมหรือสคริปต์ กับฐานข้อมูล ด้วยเหตุนี้จึงทำให้แต่ละส่วนของ CGI นั้น สามารถแยกออกจากกันได้อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณ การสร้าง การออกแบบ

3.1 หลักการทำงานของโปรแกรมสำหรับเปลี่ยนแปลงและแก้ไขข้อมูลผ่าน Browser (Update Data)

เป็นโปรแกรมที่เขียนขึ้นมาเพื่อให้ User แต่ละคนสามารถที่จะเปลี่ยนแปลงข้อมูลได้ด้วยตัวเอง โดยผ่าน Browser

3.1.1 คุณสมบัติของโปรแกรม Update Data

เมื่อทำการเลือกการทำงานให้ Update จะทำการเปลี่ยนแปลงเพิ่มข้อมูลโดยในขั้นแรกจะขึ้นหน้าจอให้แสดงผลของ ข้อมูลเก่าก่อนว่าเป็นอย่างไร ถ้าต้องการเปลี่ยนแปลงก็ให้กด Update จะขึ้นมาที่ โดยแต่ละ Block จะแสดงข้อมูลต่างๆแยกกัน เมื่อทำการแก้ไขข้อมูลเรียบร้อยแล้วก็ให้กด Submit ก็จะขึ้นมาที่ฟอร์มข้อมูลว่าข้อมูลที่เปลี่ยนแปลงเรียบร้อยแล้วเป็นอย่างไร



3.1.2 หลักการทำงานของโปรแกรม Update Data

ในขั้นแรกจะทำการเรียกข้อมูลซึ่งเป็นการจะทำการเปลี่ยนแปลงขึ้นมาก่อน โดยจะสมมุติให้เป็นไฟล์ Somsakm2.html จะขึ้นหน้าจอดังนี้

ประวัติส่วนตัว

- **Name** นายสมศักดิ์ นามสกุล มีตะกา
- **Address** 301/8 หมู่ 10 หมู่บ้านรุ่งอรุณ 2 ถนน - แขวง ลำปลาทิว เขต ลาดกระบัง จังหวัด กรุงเทพมหานคร Telephone -
- **Company** ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
Telephones 326-996

ตำแหน่งในปัจจุบัน

- **Expertise** ผู้ช่วยศาสตราจารย์
- **Executive** หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
- **Others** -

ประวัติการศึกษา

ปี พ.ศ. ที่จบ	คุณวุฒิเดิมและต่อ	สาขาวิชาเอก	สถาบัน
ปริญญาตรี 2529	อุตสาหกรรมศาสตรบัณฑิต(อ.ส.บ) คณะวิศวกรรมศาสตร์	เทคโนโลยีโทรทัศน์	KMITL
ปริญญาโท 2537	วิศวกรรมศาสตรมหาบัณฑิต(ว.ศ.ม.)	วิศวกรรมไฟฟ้า (คอมพิวเตอร์)	KMITL

สาขาที่มีประสบการณ์ในการสอนในสถาบันอุดมศึกษา

ระหว่าง พ.ศ.-พ.ศ	ชื่อสถาบัน	ชื่อสาขา/รายวิชาที่สอน
2529-ปัจจุบัน	อ.จ.ล.	Digital System design
2529-2541	อ.จ.ล.	Microprocessor & Microcontroller
2529-2540	อ.จ.ล.	Interfacing

รูปที่ 3-1 ข้อมูลของผู้ใช้งานที่ถูกเรียกขึ้นมาในขั้นแรก

หาก User ต้องการที่จะทำการเปลี่ยนแปลงข้อมูลก็จะให้ไปกดที่ปุ่ม Update

ประสบการณ์ด้านอื่นที่สำคัญและเกี่ยวข้องกับสาขาที่จบการศึกษา

- 2535-2537 เป็นวิทยากรพิเศษ การทำอากาศยานแห่งประเทศไทย (ทอท) แผนกสื่อสาร
- เป็น อาจารย์ ชูญ ลาดกระบัง

UPDATE DATA

รูปที่ 3-2 แสดงปุ่ม Update ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะทำการเรียก ไฟล์ Update.cgi เพื่อทำการแสดงผลออกมาเป็น Block เพื่อให้เปลี่ยนแปลงข้อมูล

Please Update Your Data

นายสมศักดิ์ นามสกุล มิตะถา	
3018 หมู่ 10 หมู่บ้านทุ่งสูง 2 ถนน -	
ภาควิชาวิศวกรรมคอมพิวเตอร์และวิศวกรรมศาสตร์	
สถาบันเทคโนโลยี	
ผู้ช่วยศาสตราจารย์	

รูปที่ 3-3 แสดงข้อมูลของ User ขึ้นมาบนบล็อก



Please Update Your Data

นาย สิทธิชัย นามสกุล	
3018 หมู่ 10 หมู่บ้านทุ่งสูง 2 ถนน -	

รูปที่ 3-4 แสดงข้อมูลที่ถูกเปลี่ยนแปลงไปใน Block

จากตรงลูกศร เราได้ทำการเปลี่ยนแปลงข้อมูลจาก นาย สมศักดิ์ มิตะถา เป็น นาย สิทธิชัย นิมศิริ ตระกูล เมื่อทำการเปลี่ยนแปลงข้อมูลเรียบร้อยแล้ว ให้กด Submit

2542-ปัจจุบัน หัวหน้าภาควิชา วิศวกรรมคอมพิวเตอร์
2535-2537
เป็น อาจารย์ ประจำคณะ
SUBMIT

รูปที่ 3-5 แสดงปุ่ม Submit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะทำการเรียก ไฟล์ Update2.cgi จะทำการเปลี่ยนแปลงข้อมูลแล้วแสดงขึ้นมาที่หน้าจออีกที

ภัทรพงศ์ นื่องเรือง

ประวัติส่วนตัว

- Name นาย สิทธิพร นิมศรีตระกูล
- Address 301/8 หมู่ 10 หมู่บ้านรุ่งอรุณ 2 ถนน - แขวง ลำปาดาทิว เขต ลาดกระบัง จังหวัด กรุงเทพมหานคร Telephone -
- Company ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
Telephones 326-996

รูปที่ 3-6 แสดงข้อมูลที่ถูกเปลี่ยนแปลงไปแล้ว

3.1.3 ไฟล์ที่เกี่ยวข้องและการทำงานของไฟล์แต่ละไฟล์

- ไฟล์ Somsakm2.html เป็นไฟล์ html ที่เป็นส่วนที่เก็บข้อมูลของ User แต่ละคนว่ามีข้อมูลเป็นอย่างไรบ้าง ในตอนแรกจะแสดงไฟล์นี้ ถ้าต้องการจะเปลี่ยนแปลงข้อมูลก็กด submit
- ไฟล์ Update.cgi เป็นไฟล์ที่ทำการ Load ไฟล์ข้อมูลนั้นมาเก็บไว้ที่ตัวแปร \$DATA โดยไฟล์ข้อมูลนี้ ข้อมูลต่างๆที่เราทำการเก็บไว้ เช่น NAME , ADDRESS , COMPANY , POSITION , EXPERIENCE จะทำการทำสัญลักษณ์ไว้เพื่อประโยชน์ในการ SEARCH ว่าตัวใดเป็นข้อมูลให้แสดงผลอย่างเดียว และตัวใดเป็นข้อมูลที่สามารถเปลี่ยนแปลงได้ ซึ่งต้นข้อความจะใช้ <C*> และท้ายข้อความจะใช้ <CO*> เมื่อทำการ Load ข้อมูลมาแล้วก็จะใส่ข้อมูลแต่ละตัวลงบน Block (Textarea) การหาข้อมูลแต่ละตัวออกมานั้นเราจะทำการหาโดยใช้ตำแหน่ง \$POS1 = INDEX(\$DATA , "C*") และ \$POS2 = INDEX(\$DATA , "CO*") แล้วหาความยาวของข้อมูลโดย \$LEN = \$POS2 - \$POS1 ; แล้วก็ข้อมูลตัวนั้นลงตัวแปร \$NEWSTR โดยใช้คำสั่ง \$NEWSTR = \$SUBSTR(\$DATA, \$POS1+4, \$LEN-4) ; แสดงผลแยกประเภทออกมา ถ้ามีข้อมูลตัวใดถูกเปลี่ยนแปลง ก็ให้กดปุ่ม SUBMIT ก็จะส่งข้อมูลไป
- ไฟล์ UPDATE2.CGI ซึ่งจะทำการรับข้อมูลที่ถูกเปลี่ยนแปลงไปและนำไปใส่ในไฟล์ข้อมูล โดยที่ UPDATE2.CGI จะมีโปรแกรมย่อยอยู่ 2 โปรแกรมคือ UPDATE2 และ V_UPDATE โปรแกรมย่อย UPDATE2 นั้นเมื่อรับข้อมูลมาแล้วจะนำไปใส่ในข้อมูลเดิมโดยใช้คำสั่ง \$DATA =~ S/\$NEWSTR/\$MYDATA {++++} / ; โปรแกรมย่อย V_UPDATE ก็จะนำไฟล์ข้อมูลนั้นมาแสดงผลให้เห็นว่าข้อมูลที่ถูกเปลี่ยนแปลงไปแล้วเป็นอย่างไร ถ้าต้องการจะเปลี่ยนแปลงข้อมูลเพิ่มเข้าไปอีกก็ให้กดปุ่ม UPDATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 หลักการทำงานของโปรแกรมสร้างฟอร์มพื้นฐาน (Make Form)

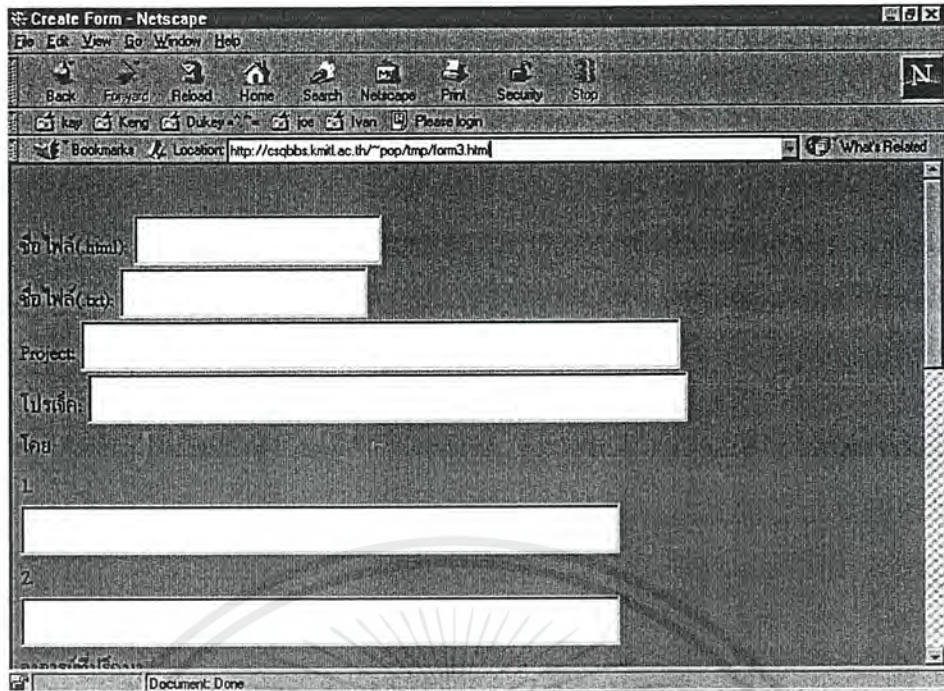
เป็นโปรแกรมที่สร้างฟอร์มให้โดยการป้อนอินพุตเข้าไฟล์ โดยอินพุตที่ป้อนจะเป็นไฟล์ภาษา Html ที่เปลี่ยนแปลงรูปแบบเพื่อให้ใช้กับโปรแกรมได้

3.2.1 ขั้นตอนและการทำงานของโปรแกรมสร้างฟอร์มพื้นฐาน

ในขั้นแรกจะเข้าสู่หน้าจอให้กรอกว่าอินพุตคืออะไร และให้กรอกชื่อไฟล์ html และ cgi ที่จะให้สร้าง หลังจากกดปุ่ม submit ไฟล์ mk_html.html จะไปทำการเรียกไฟล์ pp_test.cgi มาประมวลผล หลังจากนั้นไฟล์ pp_test.cgi ก็จะทำงานดังที่ได้กล่าวมาแล้วด้านบน

รูปที่ 3-7 แสดงหน้าจอรับค่า Input File, HTML File และ CGI File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 แสดงตัวอย่างของฟอร์มที่สร้างโดยโปรแกรมนี้

3.2.2 ไฟล์ที่เกี่ยวข้องและการทำงานของไฟล์แต่ละไฟล์

- somsakm3.html เป็นไฟล์ html ที่เปลี่ยนแปลงรูปแบบเพื่อให้ใช้งานกับโปรแกรมนี้ได้ โดยขั้นแรกไฟล์ที่ได้มาจะเป็นไฟล์ html ธรรมดา โดยลักษณะของการแก้ไขจะเป็นไปตามลักษณะของรูปแบบของประเภทฟอร์มที่ต้องการให้เป็น คือ

- textfield ให้เปลี่ยนข้อความที่ต้องการให้เติมใน textfield เป็น `` แล้วตามด้วย `!`, ตัวแปรที่จะใช้ `!`, `maxlength`, `size`, ข้อความที่จะให้ขึ้นหน้า textfield และ `&#@` ตามลำดับ
- textarea ให้เปลี่ยนข้อความที่ต้องการให้เติมใน textarea เป็น `&#ta` แล้วตามด้วย `!`, ตัวแปรที่จะใช้ `!`, จำนวนแถว, จำนวนคอลัมน์, ข้อความที่จะให้ขึ้นหน้า textarea และ `&#@` ตามลำดับ
- text ในกรณีที่เราต้องการให้มีข้อความอยู่ในฟอร์ม โดยที่ในไฟล์ html มีข้อความนั้นอยู่แล้ว ให้แทนที่ข้อความนั้นด้วย `&#tt` แล้วตาม `!`, ข้อความที่จะแสดง และ `&#@` ตามลำดับ
- pp_test.cgi เป็นไฟล์ที่เขียนด้วยภาษา perl โดยไฟล์ๆนี้จะทำการสร้างไฟล์ขึ้นมา 2 ไฟล์ โดยไฟล์แรกจะเป็นไฟล์ภาษา html สำหรับฟอร์มที่ได้จากไฟล์อินพุต และไฟล์ที่ 2 เป็นไฟล์ cgi เพื่อมาใช้ประมวลผลข้อมูลที่ได้จากฟอร์ม ซึ่งการทำงานของไฟล์ๆนี้จะเป็นอย่างนี้ คือ จะทำการเปิดไฟล์ form.html ขึ้นมาเพื่อเขียนไฟล์ลงไป ในส่วนแรกจะให้ไฟล์ form.html นี้มีฟิลด์อยู่ 2 ฟิลด์ คือ textfield ที่รับค่าชื่อไฟล์ html ที่จะแสดงข้อมูลที่กรอกไป ส่วนฟิลด์ที่ 2 เป็น textfield ที่รับชื่อไฟล์ .txt ซึ่งไฟล์ๆนี้จะเก็บค่าข้อมูลที่กรอกลงไปในแต่ละฟิลด์ในลักษณะเรียงต่อกันไป โดยมีเครื่องหมาย `!` คั่นระหว่างค่าแต่ละค่า ส่วนฟิลด์ต่อมาจะได้จาก ไฟล์อินพุตที่ป้อนเข้ามา โดยเมื่อเจอเครื่องหมาย `&` แล้วโปรแกรมจะแยกพิจารณาเป็น 4 ลักษณะคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าเจอ #ch ก็จะทำให้การสร้าง checkbox โดยการทำงานจะทำการแยกค่าออกจากกันโดยใช้เครื่องหมาย ! เป็นตัวแยก ในฟิลด์แรกคือ ch จะไม่ใช้ ฟิลด์ต่อมาจะเป็นชื่อตัวแปรของ checkbox นั้นๆ ต่อมาจะเป็นฟิลด์ที่เก็บค่าของตัวแปรของตัวเลือกนั้นๆ
- ถ้าเจอ #tf ก็จะทำให้การสร้าง textfield การทำงานจะเหมือนกัน ก็มีเครื่องหมาย ! เป็นตัวแยก โดยในฟิลด์แรกคือ tf จะไม่ใช้ ฟิลด์ต่อมาจะเป็นชื่อของตัวแปรของ textfield นั้นๆ ต่อมาเป็น size , maxLength และตามด้วยข้อความที่จะใช้วางข้างหน้า textfield ตามลำดับ
- ถ้าเจอ #ta จะทำการสร้าง textarea การทำงานจะแบ่งเป็นฟิลด์เช่นกันคือ ฟิลด์แรกคือ ta จะไม่ใช้ ต่อมาเป็นชื่อตัวแปรของ textarea นั้นๆ , จำนวนแถว , จำนวนคอลัมน์ และตามด้วยข้อความที่จะใช้วางหน้า textarea ตามลำดับ
- ถ้าเจอ #tt จะเป็นการนำข้อความที่อยู่หลังเครื่องหมาย ! มาแสดงโดยจะทำการค้นหาไปเรื่อยๆจนหมดไฟล์ หลังจากนั้น ก็จะสร้างปุ่ม submit และก็ปิดไฟล์ เป็นอันจบการเขียนไฟล์ form.html

ต่อมาจะเป็นการสร้างไฟล์ form.cgi เพื่อมาประมวลผลข้อมูลจากไฟล์ form.html โดยการทำงานของไฟล์ form.cgi จะทำการเปิดไฟล์อินพุตขึ้นมาพิจารณา โดยจะลบเครื่องหมายที่เราเพิ่มเข้าไปออกแล้วแทนด้วยข้อความที่กรอกลงไป ในฟิลด์แต่ละฟิลด์แทน ซึ่งผลที่ได้นี้จะเก็บลงไฟล์ที่กำหนด แล้วสุดท้ายก็จะนำแสดงออกหน้าจอ

นอกจากนี้ยังมีอีกโปรแกรมหนึ่งคือ ลักษณะการทำงานจะคล้ายกัน คือ จะทำงานผ่าน browser โดยสามารถที่จะกำหนดชื่อ ไฟล์อินพุต , ชื่อไฟล์ html และ ชื่อไฟล์ cgi ที่ต้องการจะสร้างได้

3.3 หลักการทำงานของโปรแกรมสร้างฟอร์ม (Form Editor)

ในโปรแกรมที่แล้วจะเป็น โปรแกรมสร้างฟอร์มที่ต้องอาศัยไฟล์อินพุตที่ป้อนเข้ามา และต้องมีการแก้ไขไฟล์ html ที่จะป้อนเข้ามา เพื่อให้ใช้งานกับโปรแกรมนี้นี้ได้ ดังนั้นผู้ที่แก้ไขไฟล์นี้ได้ต้องมีความรู้ภาษา html พอสมควร แต่ในส่วนของโปรแกรมนี้นี้จะสามารถสร้างฟอร์มได้โดยที่ผู้ใช้ไม่ต้องมีความรู้ในภาษา html เลย เพียงแค่ click ปุ่มเพื่อเลือกประเภทของ form ที่จะใช้ โปรแกรมจะทำการสร้างไฟล์ html ที่เป็นตัวฟอร์ม และไฟล์ cgi เพื่อมาประมวลผลข้อมูลที่กรอกเข้าไปอีกที นอกจากนี้ โปรแกรมยังสามารถเลือกรูปแบบของหน้าจอที่จะใช้แสดงผลของข้อมูลที่กรอกไปได้คร่าวๆอีกด้วย

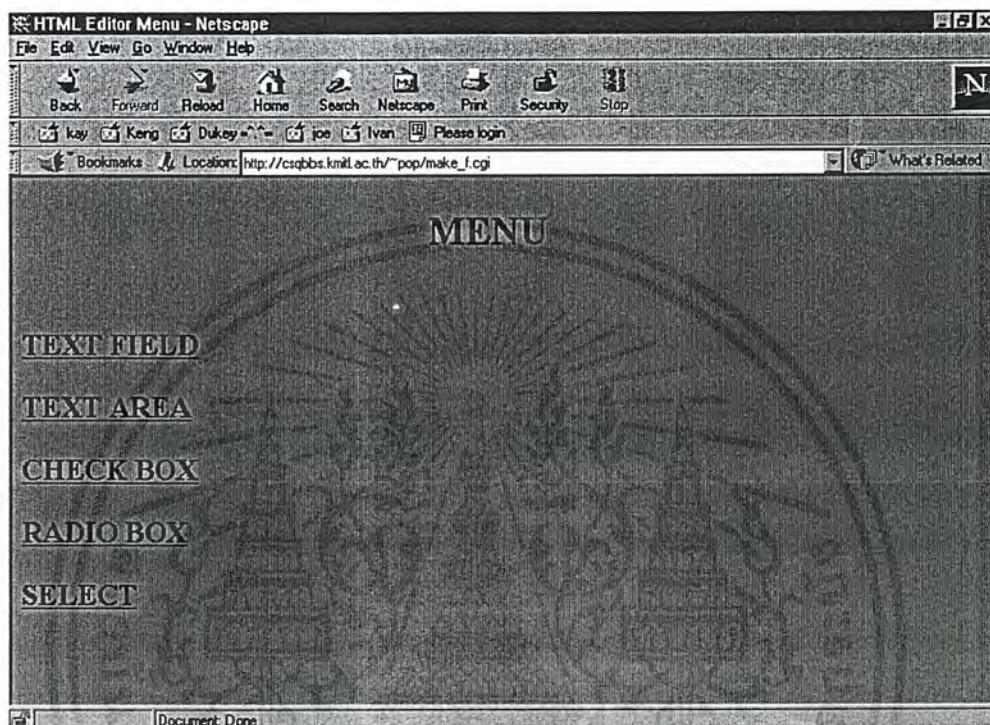
3.3.1 คุณสมบัติของ Form Editor

- มีช่องกรอกข้อความให้ใส่ชื่อไฟล์ html
- มีช่องกรอกข้อความให้ใส่ชื่อไฟล์ cgi
- มีประเภทของฟอร์มประเภทต่างๆให้เลือก โดยแบ่งเป็นประเภทต่างๆดังนี้
 - I. textfield จะมี ช่องกรอกข้อความให้กรอกดังนี้
 - text เพื่อให้กรอกข้อความที่จะเขียนหน้า textfield
 - name ให้กรอกชื่อตัวแปรที่จะใช้สำหรับ textfield นั้นๆ

- size ให้กรอกขนาดของ textfield นั้นๆ
 - maxlength ให้กรอกจำนวนตัวอักษรสูงสุดที่จะสามารถกรอกได้
2. textarea จะมีช่องกรอกข้อความให้กรอกดังนี้
- text เพื่อให้กรอกข้อความที่จะเขียนหน้า textarea
 - name ให้กรอกชื่อตัวแปรที่จะใช้แทน textarea นั้นๆ
 - rows ให้กรอกจำนวนแถวของ textarea นั้นๆ
 - columns ให้กรอกจำนวนคอลัมน์ของ textarea นั้นๆ
3. checkbox จะมีช่องให้กรอกข้อความดังนี้
- ในหน้าแรก จะให้กรอกหัวข้อของตัวเลือกเหล่านั้น
 - ในหน้าต่อมาจะให้กรอกรายละเอียดของตัวเลือกแต่ละตัว โดยจะมีรายละเอียดของตัวเลือกแต่ละตัวดังนี้
 - text ให้กรอกว่าตัวเลือกนั้นคืออะไร
 - name ให้กรอกชื่อตัวแปรของตัวเลือกนั้นๆ
 - value ให้กรอกค่าของตัวเลือกนั้นๆ
 - check ให้ระบุว่าตัวเลือกนั้นจะให้มีการถูกเลือกก่อนหรือไม่
4. radiobox มีช่องให้กรอกข้อความดังนี้
- ในหน้าแรก จะให้กรอกหัวข้อและชื่อตัวแปรของตัวเลือกเหล่านั้น
 - ในหน้าต่อมาจะให้กรอกรายละเอียดของตัวเลือกแต่ละตัว โดยมีรายละเอียดของตัวเลือกแต่ละตัวดังนี้
 - text ให้กรอกว่าตัวเลือกนั้นคืออะไร
 - value ให้กรอกค่าของตัวแปรของตัวเลือกนั้นๆ
 - check ให้ระบุว่า จะให้มีการเลือกตัวเลือกนั้นก่อนหรือไม่
5. select มีช่องให้กรอกข้อความดังนี้
- ในหน้าแรกจะให้กรอกหัวข้อและชื่อตัวแปรของตัวเลือกเหล่านั้น
 - ในหน้าต่อมาจะให้กรอกรายละเอียดของตัวเลือกแต่ละตัว โดยมีรายละเอียดดังนี้
 - text ให้กรอกว่าตัวเลือกนั้นคืออะไร
 - value ให้กรอกค่าของตัวแปรของตัวเลือกนั้นๆ
6. upload file ให้เลือกว่าจะมีการ upload file หรือไม่ โดยจะมี textfield ให้กรอกหัวข้อของการ upload file
- เมื่อทำการเลือกแต่ละตัวจนพอแล้ว ก็จะเข้าสู่หน้าจอให้เลือกรูปแบบของหน้าจอแสดงผลของหน้า html ที่จะแสดงข้อมกรอกข้อความไป ซึ่งมีรูปแบบให้เลือกดังนี้
 - สี Background โดยจะมีสีตามสีมาตรฐาน
 - สีตัวอักษร โดยจะมีสีตามสีมาตรฐานเช่นกัน
 - เลือกว่าจะแสดงผลในรูปแบบของตารางหรือไม่

3.3.2 ขั้นตอนการทำงานของโปรแกรม

ในขั้นแรกจะเข้าหน้าจอที่ให้กรอกว่าจะสร้างไฟล์ html และไฟล์ cgi ชื่อว่าอะไร เมื่อคลิกปุ่ม submit แล้ว จะไปเรียกไฟล์ menu_editor.html ซึ่งจะมีรายการของฟอร์มให้เลือกว่าจะเลือกสร้างฟอร์มแบบใด ถ้าต้องการเลือกฟอร์มแบบใดก็ click รายการนั้น



รูปที่ 3-9 แสดงหน้าจอของฟอร์มแบบต่างๆที่มีให้เลือก

หลังจากนั้นก็เข้าสู่หน้าจอที่ให้กรอกรายละเอียดของฟอร์มแต่ละแบบ ในช่วงนี้จะสามารถเพิ่มฟอร์มแบบต่างๆได้เรื่อยๆ เมื่อเพิ่มก็ให้เลือก SAVE เมื่อ click ปุ่ม SAVE แล้ว ณ จุดนี้ทั้งไฟล์ html และ ไฟล์ cgi จะถูกสร้างเรียบร้อยแล้ว จากนั้นก็จะเข้าสู่หน้าจอที่ให้เลือกรูปแบบของหน้าจอแสดงผล ดังรูป

เมื่อทำการเลือกรูปแบบของหน้าจอแสดงผลแล้ว โปรแกรมก็จะเก็บค่าต่างๆที่ได้เลือกไว้ลงไฟล์ pattern.txt ใน subdirectory tmp จากนั้นก็จะเรียกไฟล์ html ที่เป็นฟอร์มที่ได้สร้างไว้มาแสดง

FORM - Netscape

File Edit View Go Window Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Locations http://csqpbs.kmit.ac.th/~pop/pattern.cgi Who's Related

ชื่อ (name):

นามสกุล:

ชื่อ:

ที่อยู่:

อาหารที่ชอบ

พิซซ่า

แมงเม็กซิกัน

ไทชอย

Document: Done

รูปที่ 3-10 แสดงหน้าจอตัวอย่างของฟอร์มที่ใช้โปรแกรมนี้สร้าง

หลังจากนี้ถ้าทำการกรอกข้อมูลแล้วทำการกดปุ่ม submit แล้วโปรแกรมจะไปเรียก ไฟล์ cgi ที่ได้ทำการสร้างไว้มาประมวลผล จะได้ผลลัพธ์ดังรูป

Show Form - Netscape

File Edit View Go Window Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Locations http://csqpbs.kmit.ac.th/~pop/tmp/pop.cgi Who's Related

ชื่อ:	ภิศรพงศ์ น้อยเรือง
ที่อยู่:	นนทบุรี

Document: Done

รูปที่ 3-11 แสดงหน้าจอที่แสดงข้อมูลที่ได้กรอกเข้าไปในฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ไฟล์ที่เกี่ยวข้องและการทำงานของไฟล์แต่ละไฟล์

1. create_form.html เป็นไฟล์ที่จะทำการรับค่าชื่อไฟล์ที่ต้องการจะสร้าง โดยแยกเป็นชื่อไฟล์ html ซึ่งเป็นไฟล์ที่ใช้สร้างฟอร์มตามที่เราจะเลือก ส่วนอีกไฟล์เป็นไฟล์.cgi ซึ่งเขียนด้วยภาษา perl หลังจากนั้นเมื่อกดปุ่ม submit
2. make_f.cgi เป็นไฟล์ที่จะทำการเก็บค่าชื่อไฟล์ทั้ง 2 ไฟล์ลงในไฟล์ชื่อ temp.cgi ซึ่งอยู่ใน subdirectory tmp หลังจากนั้นไฟล์ make_f.cgi ก็จะเริ่มทำการสร้างไฟล์ .html และ .cgi ตามที่กำหนด โดยเริ่มจากไฟล์ .html จะเริ่มสร้าง tag form ก่อน และก็จะสร้าง textfield เพื่อให้กรอกชื่อไฟล์ html ที่เก็บข้อมูล และ textfield สำหรับให้กรอกหัวข้อของหน้าฟอร์มว่าเป็นเรื่องเกี่ยวกับอะไร หลังจากนั้นไฟล์ make_f.cgi ก็จะทำการสร้างไฟล์ cgi ต่อ โดยไฟล์ cgi จะเปิดไฟล์ html ที่เก็บข้อมูล แล้วให้เขียนหัวข้อออกหน้าจอ หลังจากนั้นก็จะปิดไฟล์ทั้ง 2 ไฟล์ แล้วเรียกไฟล์ menu_editor.html ขึ้นมา
3. menu_editor.html จะเป็นไฟล์ html ที่จะเป็น menu ให้เลือก คือ textfiled , textarea , checkbox , radio box , select และ upload file โดยแต่ละตัวเลือกจะ link ไปยังหน้าต่างๆของแต่ละหัวข้อ
4. tf.html เป็นไฟล์ html ซึ่งจะให้กรอกรายละเอียดต่างๆของ textfiled นั้นๆ หลังจากกด submit แล้ว ก็จะเรียกไฟล์ change.cgi ขึ้นมาประมวลผล
5. change.cgi จะทำการเขียนไฟล์ html และไฟล์ cgi ที่กำหนดต่อ โดยแยกตามประเภทของฟอร์มนั้นๆ คือ
 - textfiled ในส่วนของไฟล์ html จะทำการเขียน tag ของ textfiled ตามที่กำหนด ส่วนไฟล์ cgi ไฟล์นี้จะทำการเขียนหัวข้อของ textfiled นั้นๆ และข้อมูลที่ได้กรอกลงไปไปใน textfiled
 - textarea ในส่วนของไฟล์ html จะทำการเขียน tag ของ textarea ตามที่กำหนด ส่วนไฟล์ cgi ไฟล์นี้จะทำการเขียนหัวข้อของ textarea นั้นๆ รวมทั้งข้อมูลที่กรอกลงไปไปใน textarea
 หลังจากนั้นก็จะทำการเรียกไฟล์ return.html ขึ้นมา
6. ta.html เป็นไฟล์ html ที่จะให้กรอกรายละเอียดของ textarea นั้นๆ และเมื่อกดปุ่ม submit ก็จะไปเรียกไฟล์ change.cgi ขึ้นมาประมวลผล
7. ch_top.html เป็นไฟล์ html ที่ให้กรอกหัวข้อของ checkbox นั้นๆ
8. ra_top.html เป็นไฟล์ html ที่ให้กรอกหัวข้อของ radio box นั้นๆ
9. select_top.html เป็นไฟล์ html ที่ให้กรอกหัวข้อของ select นั้นๆ
10. ch_top.cgi เมื่อทำการกรอกหัวข้อของแต่ละฟอร์มแล้ว ไฟล์ๆนี้จะทำการสร้างไฟล์ html ต่อ โดยจะเขียนหัวข้อที่ได้กรอกไปลงไฟล์ html ในส่วนของไฟล์ cgi จะเขียนให้ไฟล์ cgi นี้เขียนหัวข้อลงไฟล์ html เช่นกัน ต่อมาก็จะแยกเป็นหัวข้อตามประเภทของฟอร์มแต่ละแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยฟอร์มประเภท radio box และ select จะทำการเขียนชื่อตัวแปรลงไฟล์ ra.txt และ select.txt ตามลำดับ และจะเรียกไฟล์ที่จะทำการสร้างตัวเลือกของฟอร์มแต่ละแบบ

11. ch.html เป็นไฟล์ html ที่ให้กรอกรายละเอียดของ checkbox
12. ra.html เป็นไฟล์ html ที่ให้กรอกรายละเอียดของ radio box
13. select.html เป็นไฟล์ html ที่ให้กรอกรายละเอียดของ select
14. ch.cgi เป็นไฟล์ cgi ที่ทำงานแยกตามประเภทของฟอร์มแต่ละแบบ คือ
 - check box จะสร้างไฟล์ html โดยจะทำการสร้างตัวเลือกของ checkbox นั้นๆ ในส่วนของไฟล์ cgi ไฟล์ cgi นี้จะทำการเช็คว่าคุณเลือกหรือไม่ ถ้าเลือกก็จะเขียนค่าตัวเลือกนั้นลงไฟล์ html
 - radio box จะสร้างไฟล์ html โดยจะทำการสร้างตัวเลือกของ radio box นั้นๆ ส่วนของไฟล์ cgi ไฟล์ cgi นี้จะทำการเช็คว่าคุณเลือกหรือไม่ ถ้าเลือกก็จะเขียนค่าตัวเลือกนั้นลงไฟล์ html
 - select จะสร้างไฟล์ html โดยจะทำการสร้างตัวเลือกของ select นั้นๆ ส่วนของไฟล์ cgi ไฟล์ cgi นี้จะทำการเช็คว่าคุณเลือกหรือไม่ ถ้าเลือกก็จะเขียนค่าตัวเลือกนั้นลงไฟล์ html

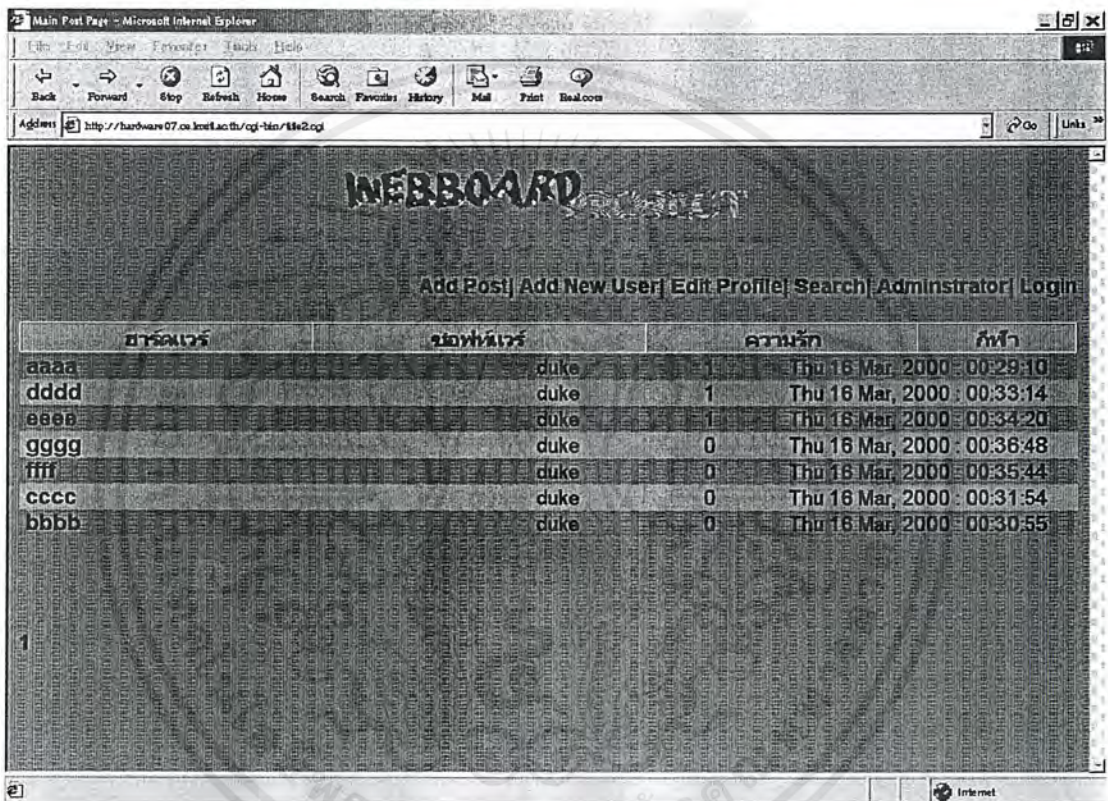
หลังจากนั้นก็ทำการสร้าง link เพื่อให้เลือกว่าจะเพิ่มตัวเลือกต่อ หรือจะทำการเพิ่มฟอร์มแบบอื่นแทน

15. ok_select.cgi เป็นไฟล์ที่เขียนไฟล์ html เพื่อปิด tag select และเรียกหน้า return.html ขึ้นมา
16. return.html เป็นหน้าที่จะทำให้ทำการเลือกว่าจะทำการเพิ่มฟอร์มแบบอื่นๆต่อ หรือว่าจะพอแค่นี้แล้ว save ค่าทั้งหมด
17. end.cgi เป็นไฟล์ที่เขียนด้วยภาษา perl โดยไฟล์นี้จะทำการเขียนไฟล์ html โดยจะสร้างปุ่ม submit และปุ่ม clear และปิดไฟล์ ส่วนไฟล์ cgi จะไปตั้งให้ unlink ไฟล์ pattern.txt, temp.cgi, ra.txt และ select.txt และจะทำการเรียกไฟล์ html ที่เก็บข้อมูลที่กรอกไปออกมาแสดงผล
18. pattern.html เป็นไฟล์ที่ให้เลือกรูปแบบของหน้าจอแสดงผล เมื่อกดปุ่ม submit ก็จะไปเรียกไฟล์ pattern.cgi
19. pattern.cgi เป็นไฟล์ที่จะทำการเขียนข้อมูลที่ได้ออกไว้ในไฟล์ pattern.html ลงไฟล์ pattern.txt ใน subdirectory tmp หลังจากนั้นก็จะเรียกไฟล์ html ของฟอร์มที่ได้สร้างไว้ออกมาแสดงผล

หลังจากนั้น เมื่อแสดงหน้าฟอร์มนั้นแล้วมีการกดปุ่ม submit ตัวไฟล์ cgi ก็จะทำการเขียนข้อมูลทั้งหมดที่กรอกไปลงไฟล์ที่ได้กำหนดไว้ แล้วแสดงผลออกหน้าจอ

3.4 หลักการทำงานของกระดานข่าว (Webboard)

กระดานข่าวเป็นโปรแกรมที่ให้ผู้ใ้ใช้ที่มีคำถามหรือต้องการที่จะประกาศข่าวสารต่าง ๆ บนอินเทอร์เน็ตที่สามารถมาประกาศข้อความต่างๆได้ กระดานข่าวที่ให้เห็นทั่วไปยังมีความสามารถในการทำงานที่ยังน้อยอยู่ หรือถ้าจะหากระดานข่าวที่มีความสามารถมากก็จะต้องดาวน์โหลดจากเว็บไซต์ต่างประเทศ ซึ่งแนวคิดที่ได้เขียนกระดานข่าวนี้นี้ขึ้นมาเพื่อต้องการกระดานข่าวที่มีความสามารถมาก แต่เป็นฝีมือของคนไทย ซึ่งความสามารถหรือคุณสมบัติของกระดานข่าวนี้นี้จะได้กล่าวพร้อมกับแนวคิด และการออกแบบในลำดับต่อไป



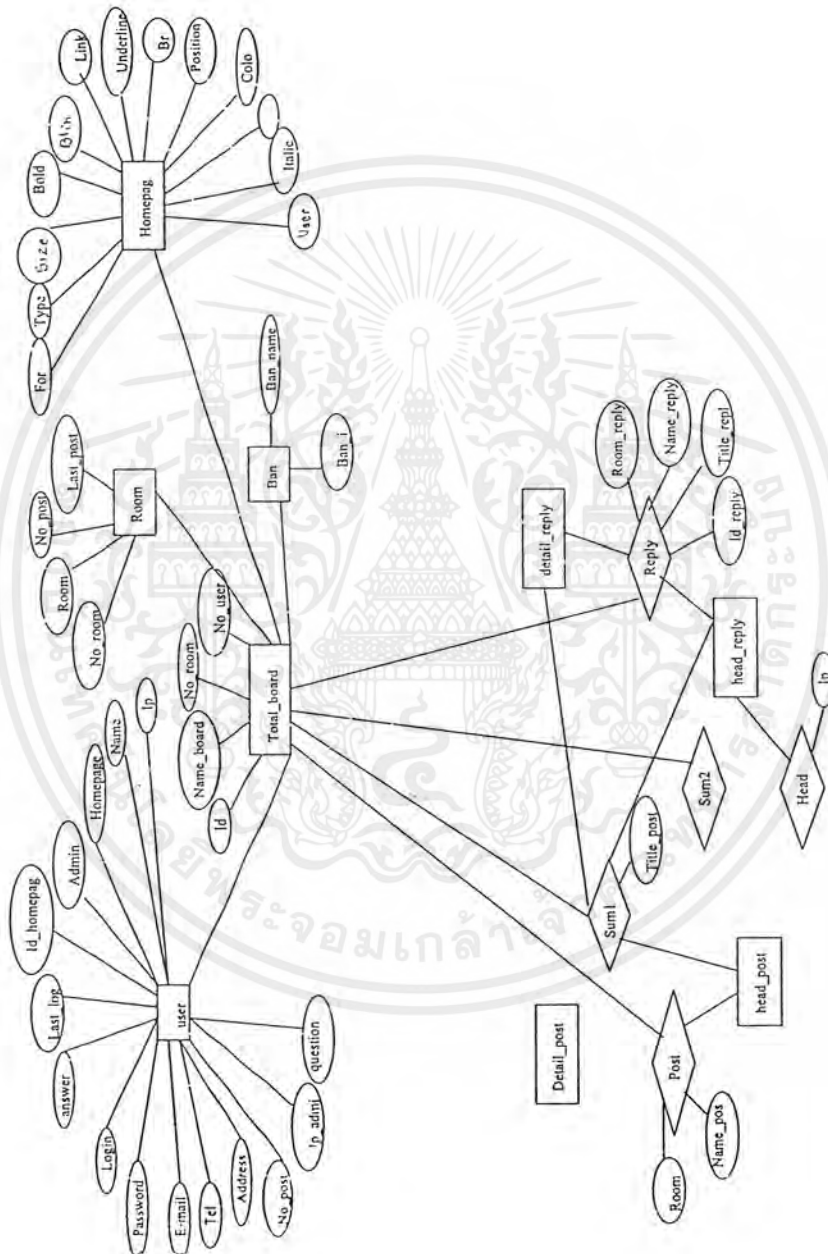
รูปที่ 3-12 แสดงหน้าจอหลักของกระดานข่าว

3.4.1 การออกแบบฐานข้อมูล

ในขั้นแรกก่อนจะทำการเขียนโปรแกรมจะต้องทำการออกแบบระบบฐานข้อมูลก่อน ซึ่งการออกแบบระบบฐานข้อมูลนี้ในขั้นแรกต้องนึกถึงก่อนว่าเราต้องการให้กระดานข่าวของเราสามารถทำอะไรได้บ้าง จากนั้นก็ทำการกำหนดว่าในแต่ละการทำงานของกระดานข่าวที่เราต้องการนั้นต้องมีการเก็บข้อมูลอะไรบ้าง แล้วทำการสร้างตารางเก็บข้อมูลที่เกี่ยวข้องกันอยู่ด้วยกัน

3.4.2 E-R Diagram ของกระดานข่าว

E-R Diagram ของข้อมูลบนฐานข้อมูลจะได้ดังนี้



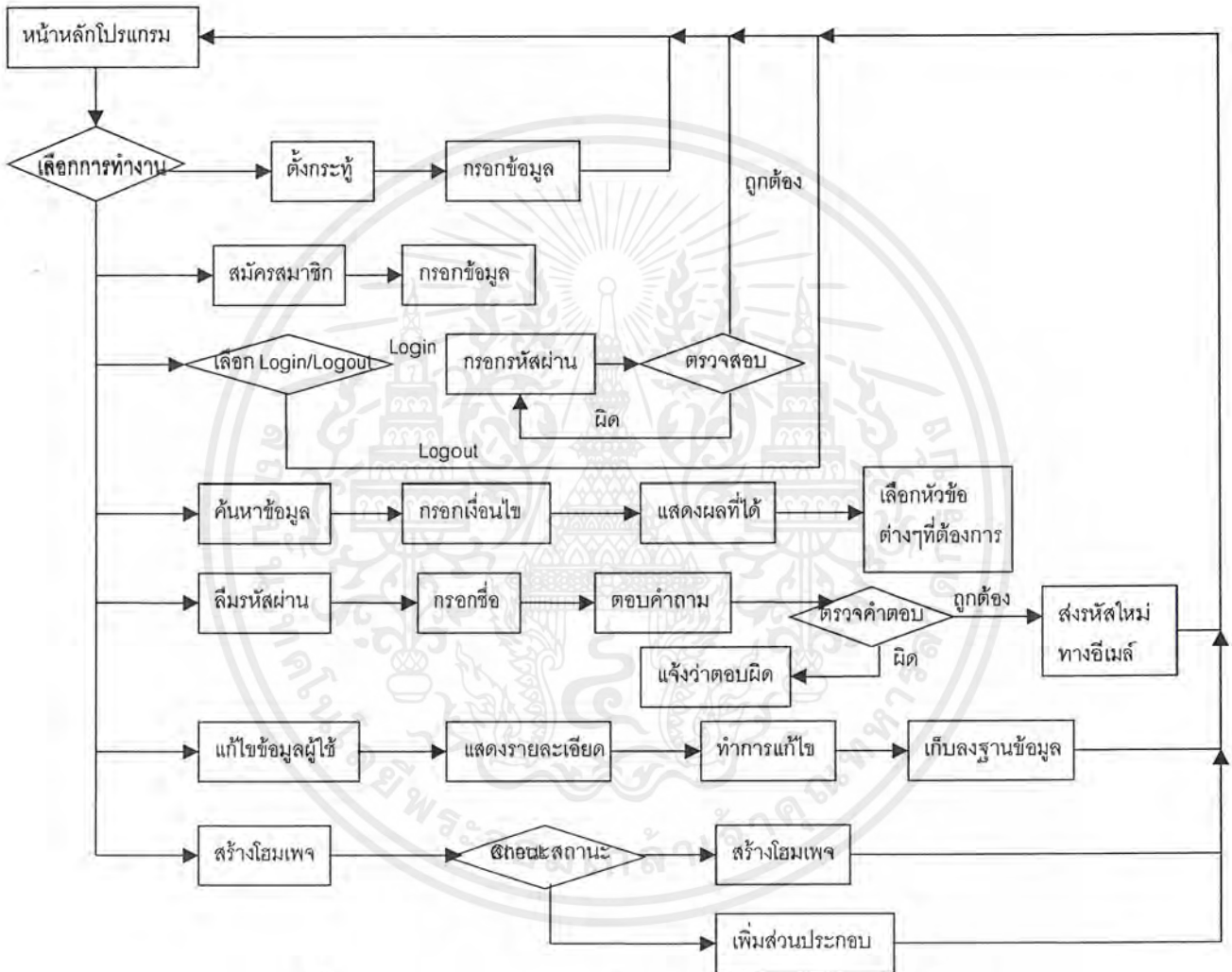
รูปที่ 3-13 E-R Diagram ของกระดานข่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การทำงานของกระดานข่าวในรูปของ Flow Chart

การทำงานของกระดานข่าวเขียนออกมาเป็น Flow Chart โดยที่จะแสดงออกมาเป็นสองส่วนหลักๆด้วยกัน

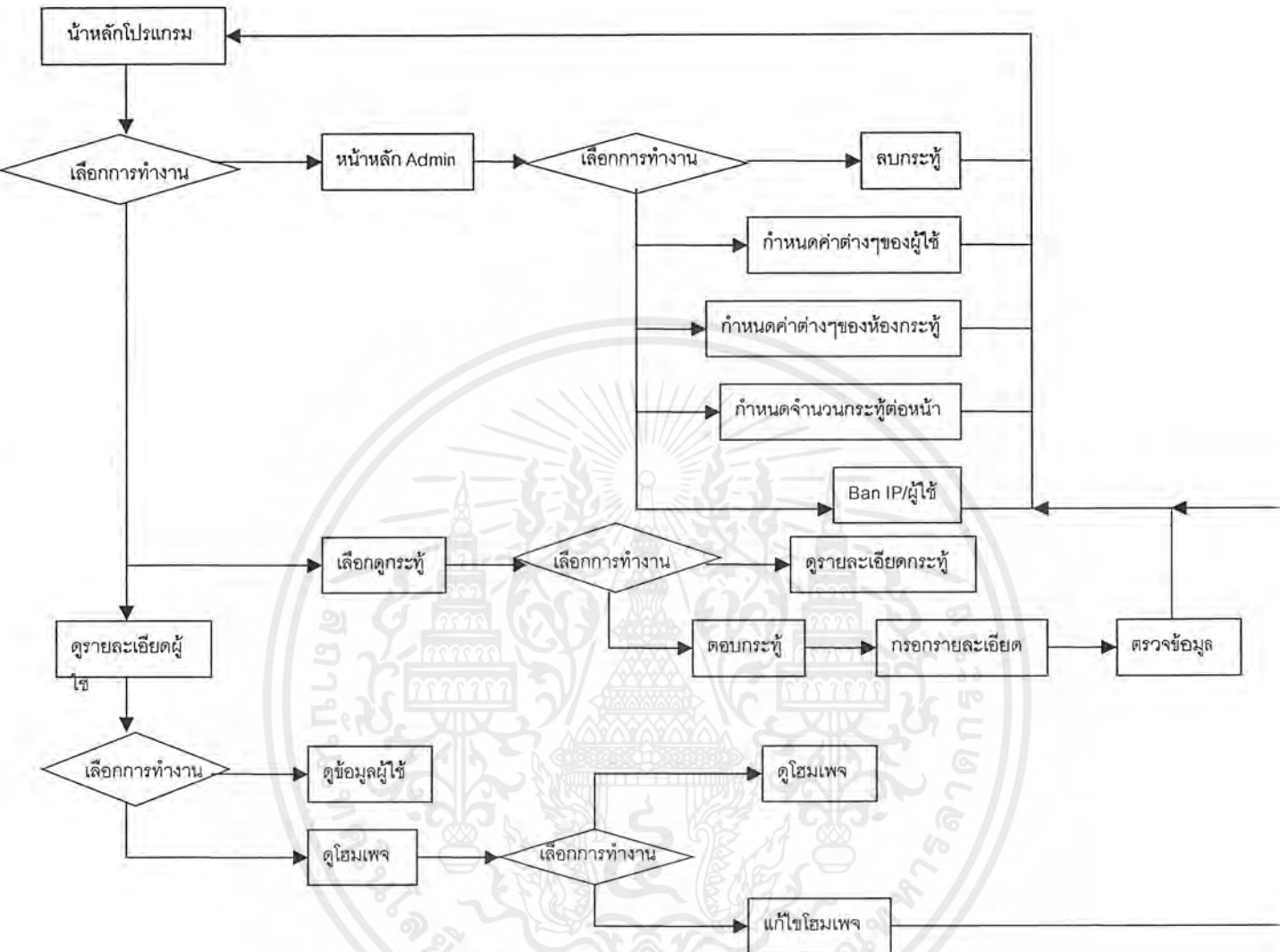
ในส่วนแรกของ Flow Chart นั้นจะแสดงถึงการทำงานของผู้ใช้งานกระดานข่าวทั่วไปที่ไม่ได้เป็นผู้ดูแลกระดานข่าว (administrator)



รูปที่ 3-14 แสดง Flow Chart การทำงานของผู้ใช้งานกระดานข่าวทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนที่สองจะแสดงถึงการทำงานของผู้ดูแลระบบ (administrator)



รูปที่ 3-15 แสดง Flow Chart การทำงานของผู้ดูแลระบบ

3.4.4 แนวความคิดในการสร้างกระดานข่าว

- กระดานข่าวที่ได้สร้างขึ้นมานี้ มีจุดประสงค์หลักคือต้องการเพิ่มความสะดวกสบายให้กับผู้ใช้งาน ทั้งผู้ใช้งานกระดานข่าวทั่วไป และผู้ดูแลกระดานข่าว มีความปลอดภัยในการใช้งานกระดานข่าว และเพิ่มการใช้งานในลักษณะอื่นที่กระดานข่าวทั่วไปไม่มี
- ในส่วนของความสะดวกสบายนั้น กระดานข่าวที่สร้างขึ้นจะสร้างโดยพยายามให้ผู้ใช้สามารถทำงานต่างๆ ได้โดยผ่านขั้นตอนให้น้อยที่สุด ไม่ว่าจะเป็นการตั้งกระทู้ การสมัครสมาชิก การแก้ไขข้อมูลผู้ใช้ และการเก็บข้อมูลจะเก็บลงฐานข้อมูล ทำให้ผู้ใช้สามารถที่จะค้นหาและจัดการข้อมูลต่างๆ ได้
- ในส่วนของความปลอดภัยของกระดานข่าว ผู้ใช้งานถ้าต้องการจะตั้งกระทู้ จะต้องทำการ Login ก่อน และจะมีรหัสสำหรับผู้ใช้งานแต่ละคน เมื่อทำงานเรียบร้อยแล้วจะต้อง Logout ออก ซึ่งผู้ใช้จะไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถไปยุ่งกับข้อมูลอื่นๆได้ ถ้าไม่ได้เป็นผู้ดูแลระบบ นอกจากนี้จะมีการตรวจสอบกระตุ้ที่ไม่เหมาะสม และไม่แสดงออกมาที่กระดานข่าว เช่น การตรวจสอบคำหยาบ

- ในส่วนของงานอื่นที่เพิ่มขึ้นมา กระดานข่าวนี้อาจทำให้ผู้ที่ใช้งานสามารถที่จะสร้างโฮมเพจส่วนตัวได้ โดยสร้างจากอุปกรณ์ที่เราสร้างขึ้น

3.4.5 การออกแบบและการทำงานในแต่ละส่วนของกระดานข่าว

กระดานข่าวที่ได้ทำการสร้างนี้มีการออกแบบและลักษณะการทำงานดังต่อไปนี้

3.4.5.1 การเข้าระบบ (Login)

กระดานข่าวนี้อาจจะไม่ให้ผู้ที่ไม่ได้เป็นสมาชิกสามารถมาตั้งกระตุ้ได้ ผู้ที่จะเข้ามาตั้งกระตุ้นั้นจะต้องเป็นสมาชิกเสียก่อน โดยจะต้องเข้าไปสมัครสมาชิกที่ Add New User แต่ผู้ที่ไม่ได้เป็นสมาชิกนั้นก็ยังสามารถที่จะเข้าไปดูกระตุ้ได้ เพียงแต่จะไม่สามารถตอบกระตุ้หรือตั้งกระตุ้ได้เท่านั้น



รูปที่ 3-16 แสดงหน้าจอการเข้าระบบของกระดานข่าว

ในขั้นแรกเมื่อผู้ใช้ออกล็อกอิน และรหัสผ่านเรียบร้อยแล้ว โปรแกรมจะทำการตรวจสอบก่อนว่าล็อกอินที่กรอกมาเป็นช่องว่างหรือไม่และทำการตรวจสอบในฐานข้อมูลว่ารหัสผ่านที่กรอกมามีล็อกอินอะไรบ้าง แล้วเก็บใส่อาเลย์ จากนั้นก็จะนำมาค่าอาเรย์นั้นมาเทียบค่าล็อกอินที่ผู้ใช้ออกเข้าไปถ้าไม่เท่ากันก็จะให้กรอกใหม่ แต่ถ้าเท่ากันก็แสดงว่ารหัสผ่านถูกต้องก็จะเก็บข้อมูล IP และเวลาเข้าระบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันลงฐานข้อมูล แล้วทำการเรียกหน้าจอหลักขึ้นมา แต่ในคราวนี้ส่วนของล็อกอินนี้จะเปลี่ยนเป็น ล็อกเอาต์แล้ว เพราะผู้ใช้ได้เข้าสู่ระบบแล้ว

3.4.5.2 การออกจากระบบ (Logout)

เมื่อผู้ใช้ต้องการที่จะเลิกใช้กระดานข่าวแล้ว ก็จะทำการออกจากระบบ ในขั้นนี้ตัวโปรแกรมจะทำการ เปลี่ยนข้อมูลในคอลัมน์ IP เป็นช่องว่าง (คอลัมน์นี้จะเก็บค่า IP Address ของเครื่องของผู้ใช้ เมื่อผู้ใช้เข้าระบบ) แต่ถ้าผู้ใช้ปิดโปรแกรมขณะที่ยังไม่ได้ทำการออกจากระบบ ในคราวต่อไปถ้าผู้ใช้มาใช้กระดานข่าวที่เครื่องเดิมอีกครั้ง กระดานข่าวนั้นก็ยังคงอยู่ในสถานะล็อกอินอยู่ เนื่องจากการตรวจสอบสถานะนั้น โปรแกรมจะตรวจสอบจากค่า IP ในฐานข้อมูลแล้วนำมาเทียบกับ IP ของเครื่องถ้าตรงกัน ก็จะกำหนดให้อยู่ในสถานะอยู่ในระบบ แต่ถ้าไม่ตรงกันก็จะอยู่ในสถานะนอกระบบ

3.4.5.3 แนวความคิดในการสร้าง Login / Logout

กระดานข่าวที่สร้างขึ้น กำหนดให้มีส่วน Login / Logout เนื่องจาก บางกระดานข่าวจะไม่มีส่วนนี้ทำให้ ใครก็ตามที่เข้ามาใช้งานสามารถที่จะตั้งกระทู้ได้หมด และเข้ามายุ่งเกี่ยวกับระบบมากเกินไป ทำให้เกิดปัญหาขึ้น ถ้ามีส่วนนี้แล้ว สามารถที่จะรักษาความปลอดภัยของกระดานข่าวได้ขั้นหนึ่ง และทำให้ระบบไม่รับ resource มากเกินไป อีกทั้งสามารถจัดการกับข้อมูลของสมาชิกแต่ละคนได้อย่างมีประสิทธิภาพ

3.4.5.4 การสมัครสมาชิก (Add New User)

การสมัครนี้เพื่อสมาชิกที่ต้องการที่จะมีสิทธิในการตอบหรือตั้งกระทู้ในกระดานข่าวนี้ ซึ่งโปรแกรมจะให้ผู้ใช้กรอกข้อมูลต่างๆของตัวผู้ใช้เองดังรูป

The screenshot shows a Microsoft Internet Explorer window titled "Add New User". The address bar contains the URL "http://hardware07.co.kr/act/cp.cgi-bin/sp_user.cgi". The main content area displays a registration form with the following fields: Name, Login, E-mail, Telephone, Pager, Address, Occupation, Homepage, and ICQ Number. Below the form, there is a message: "*** I will send your password by E-mail, Please fill your correct E-mail ***". At the bottom of the form, there are two buttons: "Submit" and "Clear". The browser's status bar at the bottom shows "Done" and "Internet".

รูปที่ 3-17 แสดงหน้าจอการสมัครสมาชิกของกระดานข่าว

จากนั้นโปรแกรมจะทำการตรวจสอบข้อมูลบางตัวเช่น ล็อกอิน ถ้าผู้ใช้ไม่ได้กรอกก็จะให้ผู้ใช้ทำการกรอกใหม่ ถ้าผู้ใช้กรอกครบถ้วนก็จะนำค่าล็อกอินมาตรวจสอบว่าซ้ำกับที่มีอยู่หรือไม่ ถ้าซ้ำก็จะให้กรอกใหม่ แต่ถ้าไม่ซ้ำโปรแกรมจะทำการสุ่มค่าตัวเลขขึ้นมา 8 หลัก เพื่อเป็นรหัสผ่านให้ผู้ใช้แต่ละคน จากนั้นก็จะนำข้อมูลต่างๆเก็บลงฐานข้อมูล ส่วนรหัสผ่านนั้นก็จะส่งไปให้ผู้ใช้ผ่านอีเมล

3.4.5.5 การแก้ไขข้อมูลผู้ใช้ (Edit Profile)

เมื่อผู้ใช้ต้องการที่จะแก้ไขข้อมูลที่ได้กรอกไว้ในตอนสมัครสมาชิก ก็สามารถที่จะทำได้โดยการเลือกหัวข้อ Edit Profile แต่ถ้าผู้ใช้ยังไม่เข้าสู่ระบบโปรแกรมก็จะเตือนผู้ใช้ให้เข้าสู่ระบบก่อน ถ้าผู้ใช้เข้าสู่ระบบแล้ว โปรแกรมก็จะไปดึงข้อมูลของผู้ใช้คนนั้นจากฐานข้อมูล โดยการทำจะรู้ว่าจะไปดึงข้อมูลของผู้ใช้คนไหนมาให้แก้ไขนั้น โปรแกรมจะดูจากค่า IP ของเครื่อง ถ้าค่านี้ไปตรงกับค่า IP ในฐานข้อมูลของผู้ใช้คนไหนก็จะไปดึงข้อมูลของผู้ใช้คนนั้นออกมา (ค่า IP ของแต่ละเครื่องจะไม่ซ้ำกัน) จากนั้นเมื่อผู้ใช้ทำการแก้ไขข้อมูลแล้ว กดปุ่มเพื่อตกลงแก้ไขข้อมูลแล้ว โปรแกรมจะทำการแก้ไขข้อมูลในฐานข้อมูลของผู้ใช้คนนั้นใหม่ แต่ในที่นี้ค่าล็อกอินจะเป็นค่าเพียงค่าเดียวที่ผู้ใช้ไม่สามารถแก้ไขได้

The screenshot shows a web browser window titled "Edit Profile - Microsoft Internet Explorer". The address bar shows the URL "http://hardware07.cs.kmitl.ac.th/cgi-bin/profile.cgi". The main content area displays the "Edit Your Profile" form with the following fields and values:

Name:	Nutapong Sassenanun
Password:(maxlength 8 characters):	
E-mail:	duke@csqbbs.kmitl.ac.th
Telephone:	739-0059
Pager:	152-314549
Address:	
Occupation:	student
Homepage:	csqbbs.kmitl.ac.th/~duke
ICQ Number:	

At the bottom of the form are "Submit" and "Clear" buttons. The browser's status bar at the bottom shows "Done" and "Internet".

รูปที่ 3-18 แสดงหน้าจอการแก้ไขข้อมูลส่วนตัว

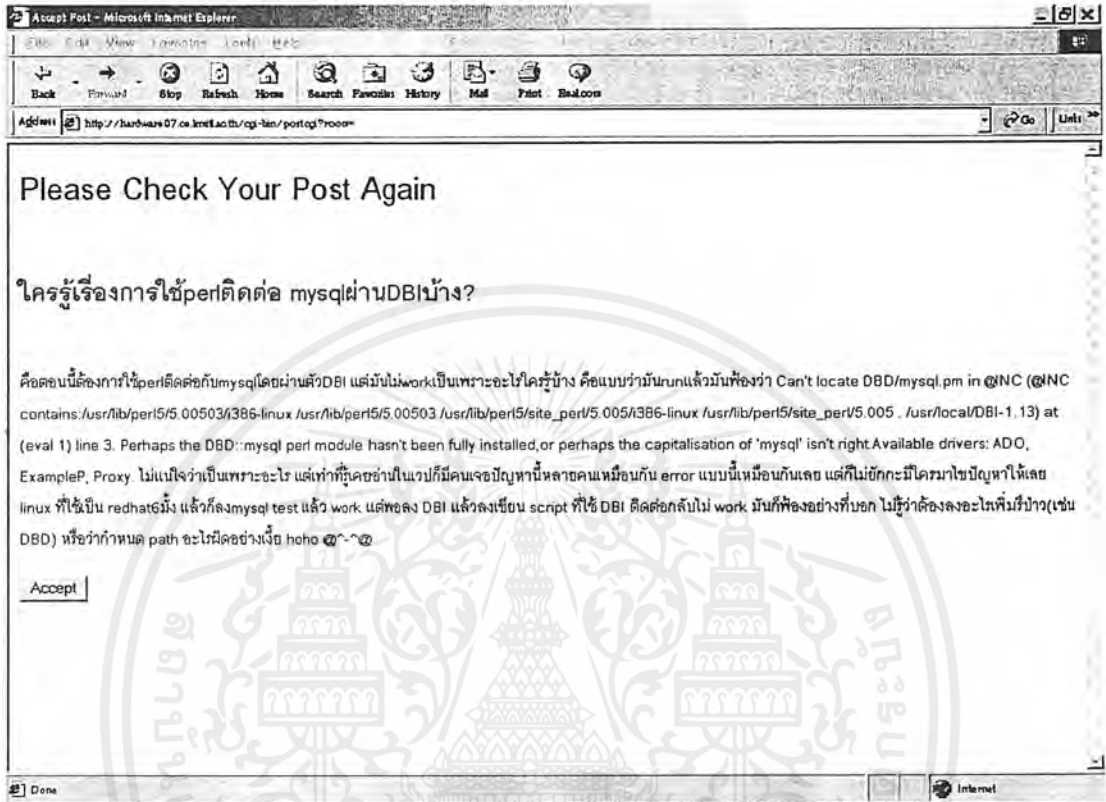
3.4.5.6 แนวความคิดในการสร้างการแก้ไขข้อมูลผู้ใช้

กระดานข่าวจะอนุญาตให้ผู้ที่สมัครเข้ามาเป็นสมาชิกแล้วสามารถเปลี่ยนแปลงข้อมูลของตนเองได้ เนื่องจากรหัสลับที่ส่งไปให้กับผู้ใช้นั้นเป็นตัวเลข 8 หลักซึ่งจะทำให้ผู้ใช้จำรหัสลับได้ยาก ดังนั้นจึงให้ผู้ใช้งานสามารถเปลี่ยนรหัสของตนเองให้จำได้ง่ายขึ้น และข้อมูลต่างๆของผู้ใช้นั้นหากมีการเปลี่ยนแปลงไป เมื่อผู้ใช้เข้ามาเปลี่ยนแปลงให้เรียบร้อยแล้ว จะเป็นการอัปเดตข้อมูล ทำให้ข้อมูลที่เก็บนั้นมีความทันสมัย และทำให้ติดต่อกับผู้ใช้งานได้ง่ายขึ้นในกรณีต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.7 การตั้งกระทู้ (Add Post)

เมื่อผู้เข้าสู่ระบบแล้วก็จะสามารถที่จะตั้งกระทู้ได้โดย โปรแกรมจะให้ผู้ใช้กรอกหัวข้อ และรายละเอียด



รูปที่ 3-19 แสดงหน้าจอเพื่อให้ผู้ใช้นั้นการตั้งกระทู้อีกครั้ง

ของกระทู้ที่จะตั้ง หลังจากนั้นโปรแกรมก็จะถามผู้ใช้เพื่อยืนยันความถูกต้องของข้อมูลอีกที ซึ่งถ้าผู้ใช้กรอกข้อมูลผิดก็สามารถที่จะกลับไปแก้ไขได้ แต่ถ้าข้อมูลถูกต้องผู้ใช้ก็จะกดปุ่มยอมรับเพื่อสร้างกระทู้นั้น จากนั้นโปรแกรมก็ทำการตรวจสอบต่างๆเป็นลำดับดังต่อไปนี้

- ตรวจสอบค่าล็อกอินโดยตรวจสอบจาก IP แล้วทำการเก็บค่าล็อกอินนั้นไว้
- แก้ไขค่าลำดับข้อความของกระดานข่าว ซึ่งค่านี้จะเพิ่มขึ้นทีละ 1 เมื่อผู้ใช้มีการตอบหรือตั้งกระทู้ในแต่ละครั้ง นั่นคือกระทู้หรือการตอบกระทู้แต่ละครั้งจะมีค่าลำดับ (Id) ของตัวมันเอง
- แก้ไขค่า no ในฐานข้อมูลซึ่งค่านี้จะเป็ค่าที่แสดงถึงลำดับการแสดงผลบนหน้าจอของหัวข้อกระทู้แต่ละตัว โดยกระทู้ที่มีการตั้งหรือตอบล่าสุดจะอยู่บนสุดแล้วไล่ลงมาตามลำดับ
- เพิ่มข้อมูลหัวข้อและรายละเอียดของกระทู้ที่ตั้งลงฐานข้อมูล
- แก้ไขจำนวนกระทู้ของห้องที่ทำการสร้างกระทู้
- แก้ไขเวลาล่าสุดที่ทำการสร้างหรือตอบกระทู้ของห้องนั้น

เมื่อทำการตรวจสอบรวมทั้งแก้ไขค่าต่างๆเรียบร้อยแล้ว ก็จะเรียกหน้าหลักของโปรแกรมขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

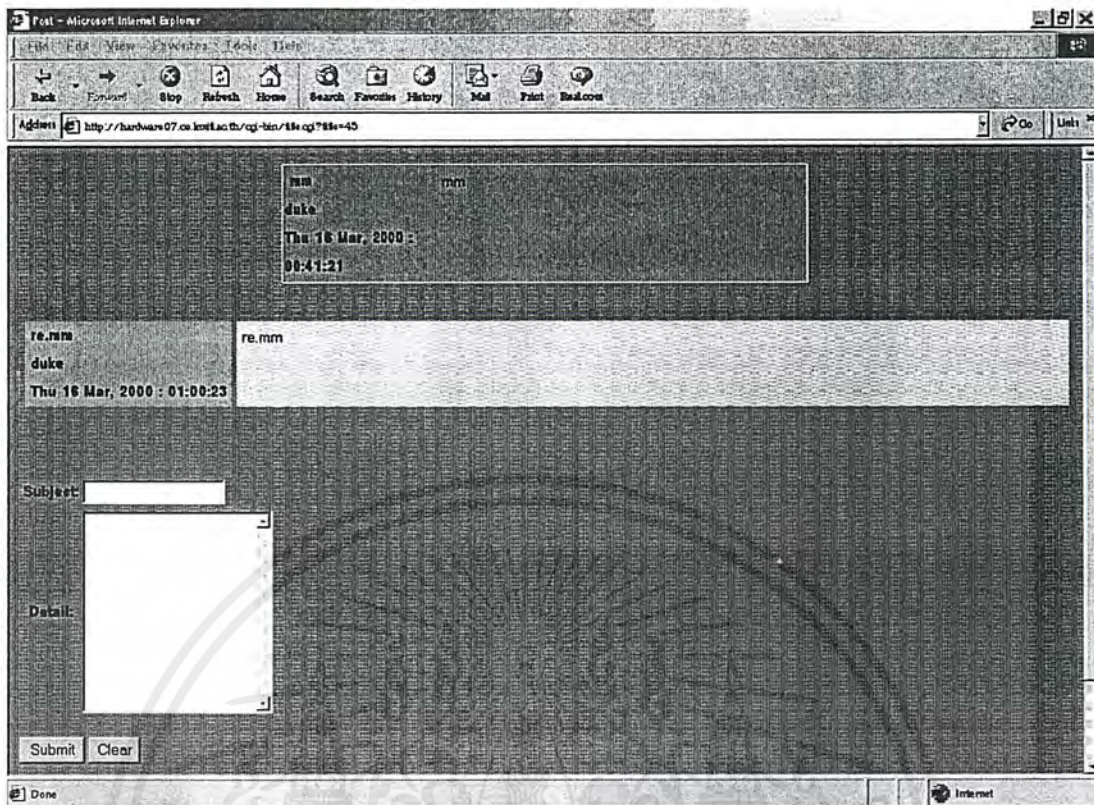
3.4.5.8 แนวความคิดในการสร้างการตั้งกระทู้

การตั้งกระทู้ของกระดานข่าวนั้น เมื่อผู้ใช้งานทำการกรอกข้อความเรียบร้อยแล้ว จะทำการแสดงหน้าจอของข้อความอีกที ซึ่งหน้าจอกี้แสดงหน้านี้จะใกล้เคียงกับที่จะแสดงบนกระดานข่าวให้ผู้ใช้งานอื่นอ่านจริงๆ เพื่อให้ผู้ที่ตั้งกระทู้ทำการตรวจสอบเพื่อความถูกต้อง และตัดสินใจอีกทีว่า กระทู้นี้สมบูรณ์อย่างที่ตั้งใจแล้วหรือไม่ ซึ่งจะได้ตรงตามที่ผู้ตั้งกระทู้ต้องการจริงๆ การตั้งกระทู้จะใส่ข้อมูลของผู้ตั้งลงไป ด้วยเพื่อบ่งบอกว่าใครเป็นผู้ตั้ง และทำให้สามารถตรวจสอบได้ หากกระทู้ที่ตั้งนั้นเป็นการไม่สมควร ก็จะถอนสมาชิกนั้นออกจากกระดานข่าว

3.4.5.9 การตอบกระทู้ (Reply)

ที่หน้าหลักของโปรแกรมจะเป็นหน้าที่แสดงหัวข้อกระทู้ของแต่ละห้อง เมื่อผู้ใช้คลิกที่หัวข้อกระทู้เพื่อดูรายละเอียด โปรแกรมจะเข้าสู่หน้าเพื่อแสดงรายละเอียดของกระทู้รวมทั้ง คำตอบที่เคยมีผู้ใช้งานอื่นเคยตอบมา ในส่วนล่างของหน้านี้จะให้ผู้ใช้สามารถตอบกระทู้เพิ่มเติมได้ โดยผู้ใช้จะต้องกรอกหัวข้อและรายละเอียดของการตอบกระทู้ เมื่อผู้ใช้กรอกเรียบร้อยแล้ว โปรแกรมก็จะให้ผู้ใช้ยืนยันรายละเอียดของข้อมูลที่ได้กรอกไปอีกที เมื่อผู้ใช้กดปุ่มเพื่อยืนยันข้อมูลแล้ว โปรแกรมจะทำการตรวจสอบและแก้ไขข้อมูลดังต่อไปนี้

- ตรวจสอบว่าผู้ใช้เข้าสู่ระบบแล้วรึยัง ถ้ายังก็จะให้ผู้ใช้เข้าสู่ระบบก่อน แต่ถ้าอยู่ในระบบแล้วก็จะเก็บค่าล็อกอินของผู้ใช้คนนั้นไว้
- แก้ไขลำดับข้อความของกระดานข่าว ซึ่งค่านี้จะเพิ่มขึ้นทีละ 1 เมื่อผู้ใช้มีการตอบหรือตั้งกระทู้ในแต่ละครั้ง
- เพิ่มข้อมูลหัวข้อและรายละเอียดของการตอบกระทู้ลงฐานข้อมูล จากนั้นก็จะเข้าหน้าจอแสดงรายละเอียดการตอบกระทู้อีกครั้ง



รูปที่ 3-20 แสดงหน้าจอการตอบกระทู้ของแต่ละหัวข้อกระทู้

3.4.5.10 แนวความคิดในการสร้างการตอบกระทู้

การตอบกระทู้ของกระดานข่าว ให้ผู้ตอบกระดานข่าวทำการเลือกจากหัวข้อที่จะตอบโดยคลิกที่หัวข้อโดยตรง เนื่องจากเป็นวิธีที่เร็วและง่ายที่สุด หลักจากนั้นจะแสดงข้อความของผู้ที่ตอบกระทู้มาก่อนหน้านี้ ซึ่งทำให้สามารถที่จะอ่านข้อความเหล่านั้นได้ และเป็นแนวทางในการตอบกระทู้ต่อไป ซึ่งจะใส่ข้อมูลของผู้ตอบลงไปด้วย ทำให้สามารถถอนผู้ตอบคนนั้นออกจากสมาชิกได้ หากคำตอบของกระทู้ นั้นไม่เหมาะสม

3.4.5.11 การค้นหาข้อมูล (Search)

การค้นหาข้อมูลในกระดานข่าวจะช่วยผู้ใช้ที่ต้องการหาข้อความที่ต้องการ โดยที่ไม่ต้องไปเปิดดูกระทู้ทุกกระทู้ ซึ่งสามารถที่จะหาได้จากหลายทาง ไม่ว่าจะเป็นจากชื่อผู้ตอบหรือตั้งกระทู้ หัวข้อกระทู้ และข้อความในกระทู้ รวมทั้งยังสามารถกำหนดช่วงของการค้นหาได้ด้วยว่าต้องการกระทู้ที่เพิ่งตั้งหรือตอบในช่วงเวลาไหน

เมื่อผู้ใช้เลือกที่จะค้นหาข้อมูลแล้ว โปรแกรมจะเข้าสู่หน้าที่ให้ผู้ใช้ป้อนข้อมูลที่ต้องการค้นหา โดยผู้ใช้จะต้องเลือกว่าจะค้นหาจากอะไร ชื่อผู้ตอบหรือตั้งกระทู้ หัวข้อกระทู้ หรือรายละเอียดของกระทู้ ได้คำที่ต้องการค้นหา กำหนดห้องที่ต้องการจะค้นหาว่าต้องการค้นหาในห้องกระทู้ไหน และระบุช่วงเวลาการตอบหรือสร้างของกระทู้ที่ต้องการค้นหา ดังรูป

รูปที่ 3-21 แสดงหน้าจอการค้นหาข้อมูลของกระดานข่าว

จากนั้นเมื่อคลิกปุ่มเพื่อทำการค้นหาแล้ว โปรแกรมก็จะทำการค้นหาข้อมูลจากฐานข้อมูล โดยมีหลักการทำงาน คือจะทำการค้นหาที่ละครั้งคือ ผู้ใช้สามารถเลือกรูปแบบการค้นหาได้หลายรูปแบบ ไม่ว่าจะเป็นจากชื่อ หัวข้อกระทู้ หรือรายละเอียด โปรแกรมก็จะทำการค้นหาที่ละรูปแบบตามข้อมูลที่กรอกเข้ามา จากนั้นก็จะแสดงผลการค้นหาออกมา โดยผลที่ได้นี้แต่ละกระทู้สามารถที่จะเชื่อมโยงไปดูรายละเอียดของแต่ละกระทู้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.12 แนวความคิดในการสร้างการค้นหาข้อมูล

การค้นหาข้อมูลของกระดานข่าวนั้น ต้องการเพิ่มความสะดวกให้กับผู้ใช้งาน ซึ่งผู้ใช้งานสามารถที่จะค้นหาข้อมูลต่างๆ ได้ โดยใส่คำลงไป และเลือกว่าจะค้นหาแบบใด จากนั้นกระดานข่าวก็จะทำการค้นหาจากฐานข้อมูล แล้วแสดงข้อมูลขึ้นมาบนหน้าจอว่าอยู่ที่ใด หัวข้อกระทู้ใด จากนั้นผู้ใช้งานสามารถทำงานต่อไปได้ ซึ่งรูปแบบของการค้นหาจะแบ่งออกว่าจะหาจากห้องใด และภายในระยะเวลาที่วันของการใส่ข้อมูลนั้น เพื่อเพิ่มความหลากหลายแก่การค้นหา

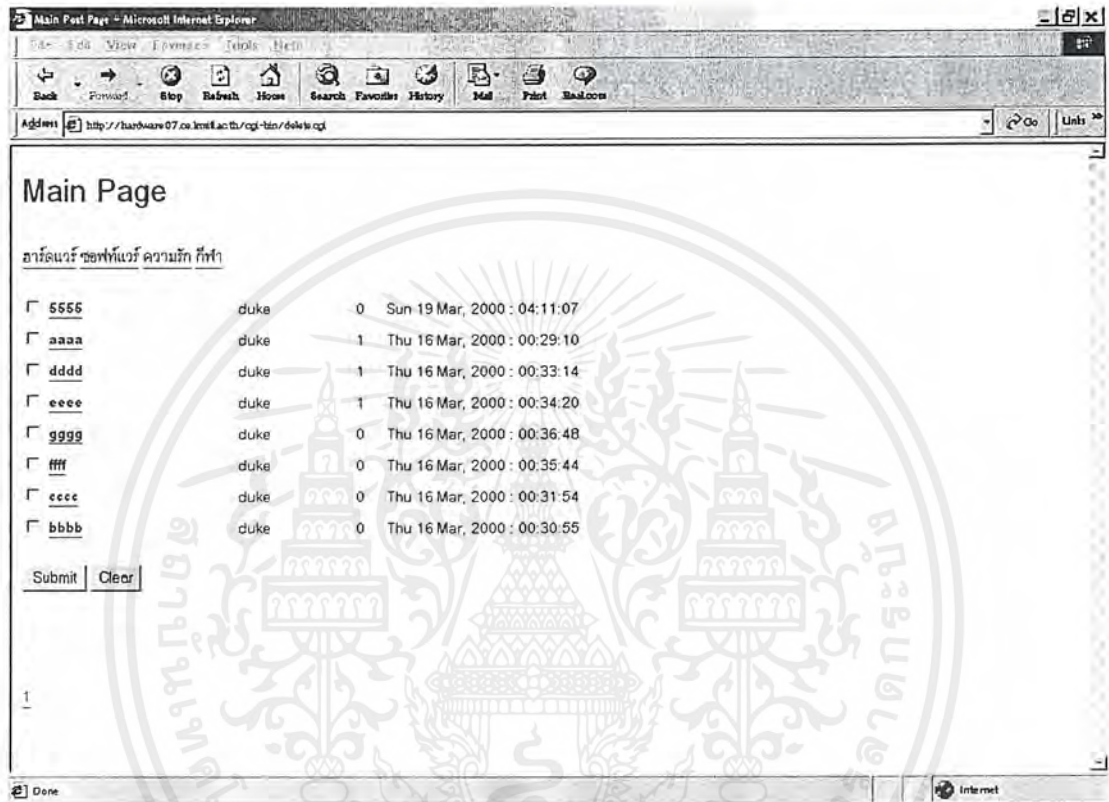
3.4.5.13 การทำงานของผู้ดูแลกระดานข่าว (Administrator)

ในส่วนของสิทธิของผู้ดูแลกระดานข่าวนี้ ในขั้นแรกจะต้องใส่ล็อกอินกับรหัสผ่านก่อน โดยผู้ที่มีสิทธิที่จะแก้ไขส่วนต่างๆ ของกระดานข่าวนั้น จะถูกกำหนดโดยผู้ดูแลอีกทีหนึ่ง (ในตอนสร้างกระดานข่าวเริ่มแรกจะสร้างผู้เข้ามาให้ 1 คน โดยมีสิทธิทุกอย่างในการแก้ไขกระดานข่าว และสามารถเลือกให้ผู้ใช้งานอื่นสามารถที่จะมีสิทธิได้เช่นกัน) จากนั้นเมื่อทำการใส่ล็อกอินและรหัสผ่านเรียบร้อยแล้ว ถ้ารหัสผ่านถูกต้องก็จะสามารถที่จะแก้ไขกระดานข่าวได้ตามคุณสมบัติดังต่อไปนี้



3.4.5.14 การลบกระทู้ (Delete Post or Reply)

เมื่อผู้ดูแลระบบเข้าทำการเลือกที่จะลบกระทู้ โปรแกรมจะเข้าสู่หน้าจอคล้ายๆกับหน้าจอหลัก แต่จะต่างกันตรงที่ ด้านหน้าของแต่ละกระทู้จะมีช่องให้เลือกที่จะลบได้ ในกรณีที่ผู้ดูแลระบบเข้าต้องการที่จะลบกระทู้ แต่ถ้าผู้ดูแลระบบเข้าต้องการที่จะลบการตอบกระทู้ ก็ทำได้โดยเลือกหัวข้อกระทู้นั้นๆ เพื่อไปดูรายละเอียดของการตอบกระทู้ที่จะลบ จากนั้นก็เลือกการตอบกระทู้ที่จะทำการ



รูปที่ 3-22 แสดงหน้าจอการลบกระทู้

ลบ เมื่อผู้ดูแลระบบเข้ากดปุ่มยืนยันทำการลบ โปรแกรมจะให้ผู้ดูแลระบบเข้ายืนยันอีกครั้ง จากนั้นโปรแกรมก็จะทำการลบข้อมูลของกระทู้ที่ได้เลือกออกจากฐานข้อมูล และจัดเรียงลำดับการแสดงผลของกระทู้บนหน้าจอใหม่ แล้วก็จะกลับสู่หน้าจอให้ผู้ดูแลระบบเข้าเลือกกระทู้ที่จะลบอีกครั้ง

3.4.5.15 แนวความคิดในการสร้างการลบกระทู้

ผู้ดูแลระบบสามารถที่จะลบหัวข้อกระทู้ได้ โดยหน้าจอจะแสดงหัวข้อกระทู้ต่างๆเรียงกันขึ้นมา หากต้องการลบก็สามารถทำได้สะดวก เพราะเป็นช่องให้คลิกลงไป และสามารถที่จะลบหลายๆอันพร้อมกันได้ เมื่อกดตกลงแล้วก็จะถามยืนยัน

3.4.5.16 การกำหนดค่าต่างๆของห้องกระทู้ (Room Configuration)

ผู้ดูแลระบบสามารถที่จะแก้ไขค่าต่างๆของห้องกระทู้ได้ดังนี้คือ

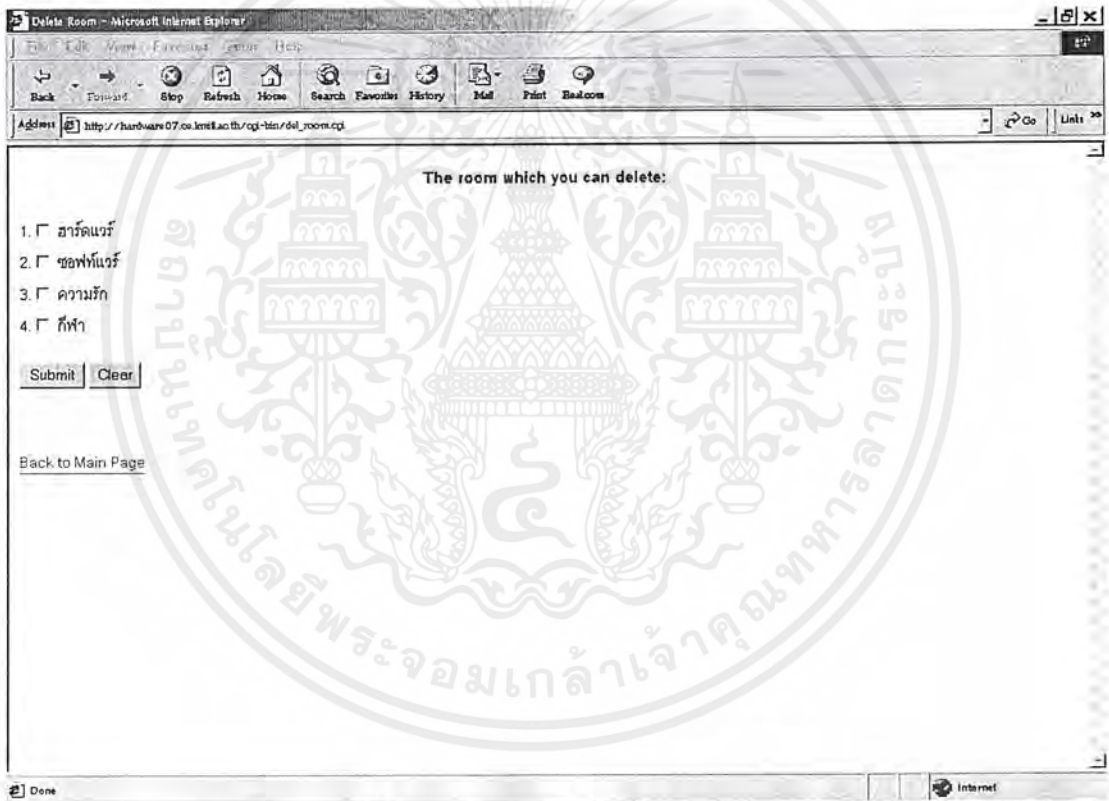
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ลบห้องกระทู้ เมื่อผู้ดูแลกระดานข่าวเลือกที่จะลบห้องกระทู้ โปรแกรมจะเข้าหน้าที่แสดงรายชื่อห้องกระทู้ที่สามารถลบได้ จากนั้นเมื่อผู้ดูแลกระดานข่าวเลือกห้องที่จะลบแล้ว โปรแกรมจะแสดงรายชื่อห้องที่ผู้ดูแลกระดานข่าวได้เลือกที่จะลบไว้แล้วให้ยืนยันอีกทีหนึ่ง เมื่อผู้ดูแลกระดานข่าวยืนยันเรียบร้อยแล้ว โปรแกรมจะทำการลบห้องกระทู้โดยมีหลักการทำงานดังนี้

- ลบกระทู้ที่อยู่ภายในห้องกระทู้นั้นทั้งหมด
- ลบห้องกระทู้นั้นออกจากฐานข้อมูล
- แก้ไขข้อมูลจำนวนห้อง ในฐานข้อมูล

จากนั้นก็เข้าสู่หน้าจอ ให้เลือกการทำงานของผู้ดูแลกระดานข่าวอีกครั้ง

2. เพิ่มห้องกระทู้ เมื่อผู้ดูแลกระดานข่าวเลือกที่จะ เพิ่มกระดานข่าวแล้ว โปรแกรมจะเข้าสู่หน้าจอให้ผู้ดูแลกระดานข่าวป้อนชื่อห้องกระทู้ จากนั้นก็จะมีการทำงานดังนี้



รูปที่ 3-23 แสดงหน้าจอการลบห้องกระทู้

- ตรวจสอบว่าชื่อห้องกระทู้ซ้ำกับที่มีอยู่หรือไม่หากซ้ำก็จะให้ตั้งชื่อใหม่
- ถ้าชื่อไม่ซ้ำก็จะเพิ่มชื่อห้องกระทู้ใหม่เข้าฐานข้อมูล และทำการแก้ไขข้อมูลจำนวนห้องในฐานข้อมูล

จากนั้นก็เข้าสู่หน้าจอ ให้เลือกการทำงานใหม่อีกครั้ง

3. เปลี่ยนชื่อห้องกระทู้ เมื่อผู้ดูแลกระดานข่าวเลือกที่จะเปลี่ยนห้องกระทู้ โปรแกรมจะแสดงรายชื่อห้องกระทู้ทั้งหมด พร้อมทั้งสามารถให้ผู้ดูแลกระดานข่าวสามารถแก้ไขชื่อได้ เมื่อผู้ดูแลกระดานเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข่าวทำการแก้ไขชื่อห้องกระทู้ แล้วกดปุ่มเพื่อยืนยันการแก้ไขแล้ว โปรแกรมก็จะไปทำการเปลี่ยนชื่อห้องในฐานข้อมูล หลังจากนั้นก็จะเข้าหน้าจอให้เลือกการทำงานใหม่



รูปที่ 3-24 แสดงหน้าจอการแก้ไขชื่อห้องกระทู้

3.4.5.17 การกำหนดค่าต่างๆของผู้ใช้ (User Configuration)

ผู้ดูแลระบบสามารถที่จะแก้ไขส่วนต่างๆที่เกี่ยวกับผู้ใช้ได้ดังนี้

1. เพิ่มสิทธิผู้ใช้ให้แก่ใจกระดานข่าวได้ (Add to Administrator Group) เมื่อผู้ดูแลระบบข่าวเลือกการทำงานนี้ โปรแกรมจะแสดงรายชื่อผู้ใช้ในกระดานข่าวทั้งหมดที่สามารถเพิ่มสิทธิได้ จากนั้นเมื่อผู้ดูแลระบบข่าวเลือกผู้ใช้ที่ต้องการแล้วกดปุ่มยืนยันแล้ว โปรแกรมก็จะไปเพิ่มฟิลด์เพื่อแสดงว่าผู้ใช้คนนี้มีสถานะเป็นผู้ดูแลระบบ หลังจากนั้นผู้ใช้คนนี้จะสามารถแก้ไขกระดานข่าวได้ โดยใช้ล็อกอินและรหัสผ่านเดิมของตนเอง

2. ลบสิทธิการแก้ไขกระดานข่าวของผู้ใช้ (Remove from Administrator Group) เมื่อผู้ดูแลระบบข่าวเลือกที่จะลบสิทธิของผู้ใช้ โปรแกรมก็จะแสดงรายชื่อผู้ใช้ที่มีสิทธิทั้งหมดเพื่อให้ผู้กระดานข่าวเลือก หลังจากนั้นก็จะให้ผู้ดูแลระบบข่าวยืนยันอีกครั้ง แล้วโปรแกรมก็จะทำการลบสิทธิ โดยลบฟิลด์ที่แสดงถึงสิทธิออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ลบชื่อผู้ใช้ออกจากสมาชิกของกระดานข่าว ในกรณีที่ผู้ใช้บางคนอาจจะมีพฤติกรรมที่ไม่เหมาะสม ผู้ดูแลกระดานข่าวสามารถที่จะลบชื่อผู้ใช้นั้นออกจากสมาชิกได้ โดยเลือกการทำงานดังกล่าว หลังจากนั้น โปรแกรมก็จะแสดงรายชื่อผู้ใช้ทั้งหมด



รูปที่ 3-25 แสดงหน้าจอการลบผู้ใช้ออกจากสมาชิก

เมื่อผู้ดูแลกระดานข่าวเลือกผู้ใช้ที่จะลบแล้ว โปรแกรมจะให้ยืนยันการลบอีกครั้ง หลังจากนั้น โปรแกรมก็จะทำการลบผู้ใช้นั้นออกจากฐานข้อมูล และแก้ไขข้อมูลจำนวนผู้ใช้ในฐานข้อมูล

3.4.5.18 แนวความคิดในการสร้างการกำหนดค่าต่างๆของผู้ใช้

ผู้ดูแลกระดานข่าวสามารถที่จะกำหนดค่าต่างๆของผู้ใช้ได้ เนื่องจากแนวความคิดว่า สิทธิของผู้ดูแลกระดานข่าวทุกคนเท่าเทียมกันหมด และมีสิทธิเหนือผู้ใช้งานกระดานข่าวทั่วไป ดังนั้นจึงให้ผู้ดูแลกระดานข่าวจัดการ เพิ่มสิทธิ ลดสิทธิ และลบสมาชิกทั่วไปได้ โดยไม่ต้องใส่รหัสอื่นๆอีก ผู้ดูแลกระดานข่าวสามารถที่จะเลือกสมาชิกทั่วไปคนใดก็ได้ ให้เข้ามาเป็นผู้ดูแลกระดานข่าวตามความเหมาะสม และสามารถลดสิทธิของผู้ดูแลกระดานข่าวให้ไปเป็นสมาชิกทั่วไปก็ได้ อีกทั้งยังสามารถถอนสมาชิกออกไปจากกระดานข่าวเสียก็ได้

3.4.5.19 การกำหนดจำนวนกระทู้ที่ให้แสดงต่อหน้า

ผู้ดูแลกระดานข่าวสามารถที่จะกำหนดจำนวนกระทู้ที่ให้แสดงต่อหน้าได้ตามความเหมาะสม โดยเมื่อผู้ดูแลกระดานข่าวเปลี่ยนจำนวนเรียบร้อยแล้ว โปรแกรมก็จะไปแก้ไขค่าในฐานข้อมูลอีกทีหนึ่ง ซึ่งค่านี้ทุกครั้งที่โปรแกรมจะแสดงกระทู้ โปรแกรมจะต้องไปดูเพื่อให้รู้ว่าจะต้องแสดงเป็นจำนวนกี่กระทู้ต่อหน้า

3.4.5.20 แนวความคิดในการสร้างการกำหนดจำนวนกระทู้ที่ให้แสดงต่อหน้า

ผู้ดูแลกระดานข่าวสามารถกำหนดจำนวนกระทู้ที่ให้แสดงต่อหน้าได้เพื่อให้ข้อความต่างๆของกระทู้ที่แสดงขึ้นมาบนหน้าจอเป็นไปอย่างเหมาะสม ไม่มากหรือน้อยจนเกินไป อีกทั้งยังเป็นผลดีต่อความเร็วในการทำงานของระบบด้วย

3.4.5.21 ห้องกระทู้

ห้องกระทู้จะเป็นห้องที่มีไว้เพื่อเก็บกระทู้ให้เป็นหมวดหมู่ เพื่อเวลาที่ผู้ใช้งานที่อาจจะสนใจกระทู้ในหมวดหัวข้อแค่บางหมวด สามารถที่จะเลือกกระทู้ดูได้โดยไม่ต้องเสียเวลา โดยห้องกระทู้ที่ผู้ดูแลกระดานข่าวสามารถที่จะกำหนดค่าต่างๆได้ตามที่ได้กล่าวมาแล้ว ในส่วนล่างของห้องกระทู้ ถ้าห้องนั้นมีจำนวนกระทู้เกินที่จะแสดงได้ต่อหน้าที่ได้กำหนดเอาไว้ ก็จะมีหน้าต่อไปให้เลือกอ่านอีก

ชาวดนเนว	บทโพทเนว	วิทยาตนาศตร	กัฟวา
แฉ	duke	0	Sun 19 Mar, 2000 04:11:24
iii	duke	1	Thu 16 Mar, 2000 00:38:23
mmn	duke	1	Thu 16 Mar, 2000 00:41:36
ooo	duke	0	Thu 16 Mar, 2000 00:43:19
กนก	duke	0	Thu 16 Mar, 2000 00:42:34
lll	duke	0	Thu 16 Mar, 2000 00:40:52
kkk	duke	0	Thu 16 Mar, 2000 00:40:03
jjj	duke	0	Thu 16 Mar, 2000 00:39:11
hhh	duke	0	Thu 16 Mar, 2000 00:37:33
ggg	duke	0	Thu 16 Mar, 2000 00:36:34
fff	duke	0	Thu 16 Mar, 2000 00:35:28
eee	duke	0	Thu 16 Mar, 2000 00:34:00
ddd	duke	0	Thu 16 Mar, 2000 00:32:56
ccc	duke	0	Thu 16 Mar, 2000 00:31:40
bbb	duke	0	Thu 16 Mar, 2000 00:30:38
aaa	duke	0	Thu 16 Mar, 2000 00:28:55

รูปที่ 3-26 แสดงห้องกระทู้และกระทู้ที่อยู่ในห้องกระทู้นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.22 แนวความคิดในการสร้างห้องกระทุ้ง

ห้องกระทุ้งที่สร้างขึ้นมานั้นจะแบ่งประเภทออกเป็นประเภทต่างๆตามห้อง ทำให้สามารถเลือกห้องที่ตัวเองสนใจและเข้าไปดูข้อมูลได้ และแต่ละห้องจะมีหัวข้อกระทุ้งเรียงกันลงมา หากต้องการดูข้อมูลของกระทุ้งใดก็เลือกเข้าไปดูได้ ซึ่งหัวข้อกระทุ้งแต่ละหัวข้อจะมีรายละเอียดบอกเอาไว้ ว่าสมาชิกคนใดเป็นคนตั้ง และตั้งขึ้นมาเมื่อใด



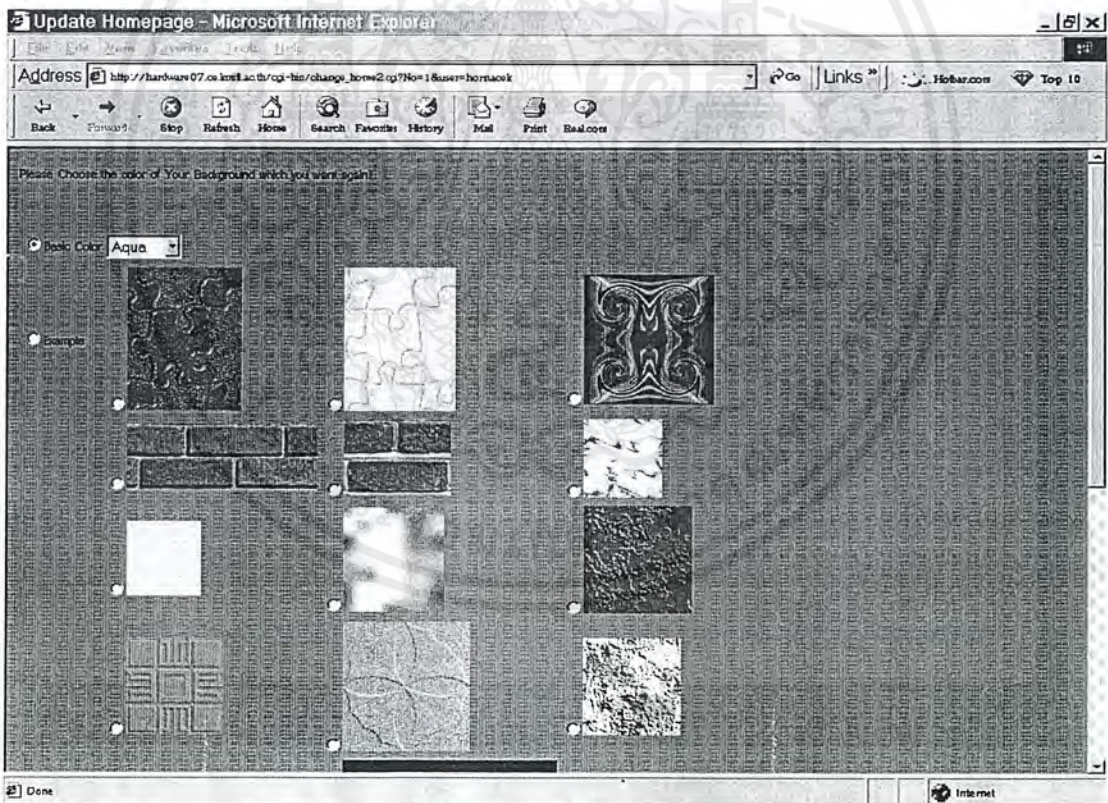
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.23 โฮมเพจส่วนตัว (Personal Homepage)

โฮมเพจส่วนตัวนี้เป็นเสมือนโปรแกรมช่วยสร้างโฮมเพจอย่างง่าย เพื่อให้ผู้ใช้สามารถโฮมเพจของตนเองได้โดยไม่ต้องมีความรู้ในการสร้างโฮมเพจ โดยโฮมเพจนี้จะเป็นโฮมเพจที่สร้างไว้สำหรับโปรแกรมกระดานข่าวนี้โดยเฉพาะ ถ้าผู้ใช้คนใดไม่มีโฮมเพจที่อื่นก็สามารถที่จะสร้างโฮมเพจที่นี้ได้ เพื่อให้คนอื่นเข้ามาดูข้อมูลต่างๆของตนเองได้ สำหรับขั้นตอนการสร้างโฮมเพจมีขั้นตอนดังต่อไปนี้

ในขั้นแรกให้ผู้ใช้ล็อกอินเข้าระบบก่อน หลังจากนั้นถ้าจะเริ่มทำการสร้างโฮมเพจก็ให้คลิกปุ่ม Create Homepage โปรแกรมจะให้เริ่มจากการให้เลือกรูปแบบของพื้นหลังก่อน โดยจะสามารถที่จะสร้างได้จาก 3 รูปแบบคือ

- เลือกจากสีพื้นฐาน เช่น สีแดง สีม่วง สีเหลือง เป็นต้น
- เลือกจากพื้นฐานหลังที่มีมาให้เลือก โปรแกรมจะได้เตรียมรูปแบบพื้นหลังมาให้ผู้ใช้เลือกที่จะใช้ได้ โดยผู้ใช้ก็สามารถที่จะคลิกเลือกตัวอย่างของพื้นหลังนั้นได้ด้วย
- เลือกจากรหัสสี โดยรหัสสีจะเป็นรหัสสีของเลขฐาน 16 จำนวน 6 หลัก โดย 2 หลักแรกเป็นสีแดง 2 หลักต่อมาเป็นสีเขียว และ 2 หลักสุดท้ายเป็นสีน้ำเงิน



รูปที่ 3-27 แสดงหน้าจอการเลือกรูปแบบพื้นหลังโฮมเพจ

หลังจากเลือกรูปแบบพื้นหลังเสร็จแล้วก็จะมียกการให้ผู้ใช้เลือกว่าจะทำการสร้างตัวอักษรหรือว่าจะใส่รูปภาพเข้าไปในโฮมเพจ ถ้าเลือกที่จะสร้างตัวอักษรโปรแกรมก็จะให้ใส่รูปแบบตัวอักษรว่าจะให้เป็นอย่างไร ให้เลือกว่าจะให้ตัวอักษรนั้นเชื่อมโดยไปยังเวปไหนได้หรือไม่ และเลือกว่าหลังจากสร้างตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรแล้วจะให้มีการเว้นบรรทัดหรือไม่ แต่ถ้าผู้ใช้เลือกที่จะใส่รูปภาพโปรแกรมก็จะให้ใส่ชื่อไฟล์รูปภาพที่จะทำการ upload เข้าไปหลังจากนั้นก็ให้กำหนดรูปแบบต่างๆของรูปภาพนั้น ในส่วนนี้จะคล้ายๆกับการสร้างตัวอักษร เมื่อสร้างเสร็จ 1 ครั้ง โปรแกรมก็จะถามว่าจะสร้างต่อหรือจะ Save คือหยุดการสร้างแล้วนั่นเอง ถ้าเลือกจะสร้างต่อโปรแกรมจะกลับเข้าสู่เมนูอีกครั้งว่าจะสร้างตัวอักษรหรือจะใส่รูปภาพ ถ้าเลือก Save ก็เป็นอันเสร็จการสร้างโฮมเพจส่วนตัว

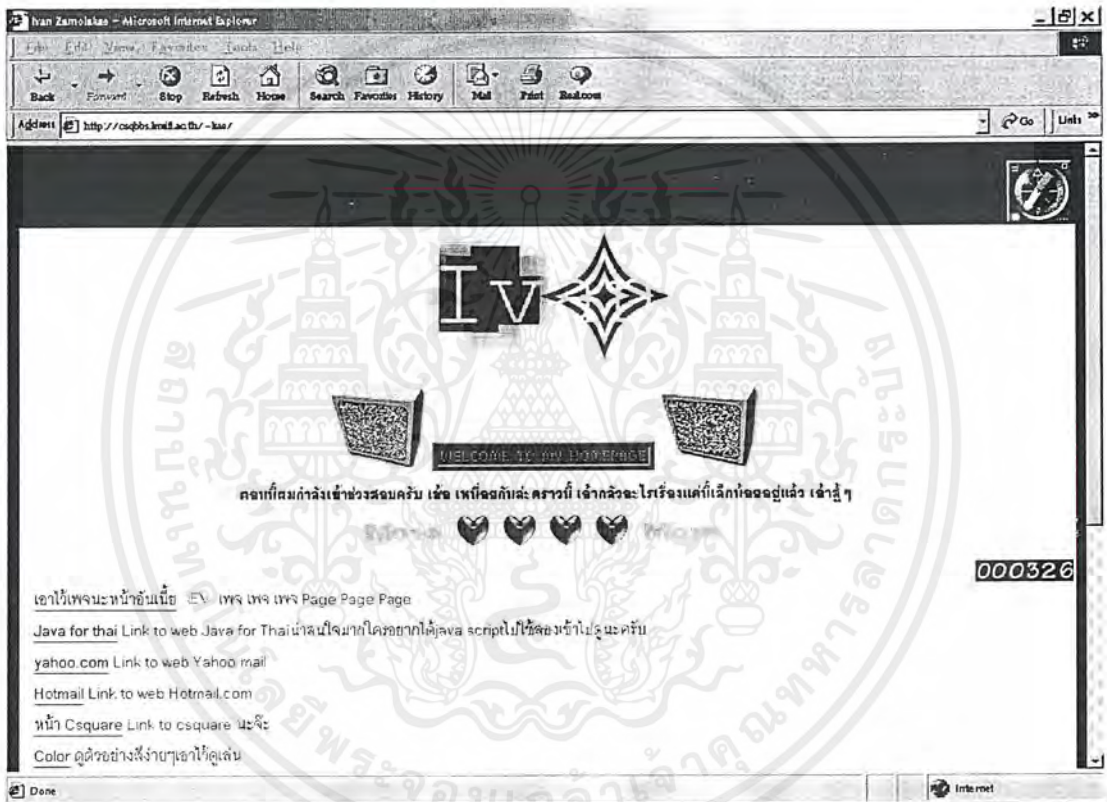
- การเพิ่มตัวอักษรหรือรูปภาพ

การเพิ่มตัวอักษรหรือรูปภาพจะเป็นการทำงานของผู้ใช้ที่ได้เคยสร้างโฮมเพจไว้แล้ว โดยจะสามารถทำงานได้โดยคลิกที่ Create Homepage เหมือนกัน แต่โปรแกรมจะเข้าสู่หน้าจอเมนูให้เลือกว่าจะใส่ตัวอักษรหรือใส่รูปภาพ หลังจากนั้นวิธีใช้ก็จะเหมือนกันกับการสร้างโฮมเพจใหม่ทุกประการ

- การแก้ไขข้อมูลโฮมเพจ



เมื่อผู้ใช้ทำการสร้างโฮมเพจแล้ว แล้วต้องการจะทำการแก้ไขข้อมูลบางส่วนที่เราได้สร้างไปก็สามารถทำได้โดย ให้คลิกที่ชื่อของผู้ใช้ที่ต้องการจะทำการแก้ไขหลังจากนั้น โปรแกรมจะให้ใส่รหัสผ่านของผู้ใช้คนนั้น หลังจากนั้นถ้าผู้ใช้กรอกรหัสถูกต้องโปรแกรมก็จะมีส่วนต่างๆของโฮมเพจที่ได้เคยสร้างไว้มาให้เลือกจะทำการแก้ไขส่วนไหน ไม่ว่าจะเป็นรูปแบบพื้นหลัง ตัวอักษรหรือรูปภาพก็ตาม เมื่อเลือกเสร็จแล้วก็สามารถทำการแก้ไขรูปแบบต่างๆได้ตามที่ต้องการ ซึ่งจะมีให้เลือกอยู่ 2 แบบก็คือ Delete Object หรือ Update Object โดยที่ Delete Object จะเป็นการลบส่วนนั้นออกไป ส่วน Update Object จะเป็นการเปลี่ยนแปลงรูปแบบของส่วนที่เราเลือก จะคล้ายๆกับตอนที่เราสร้างส่วนนั้นในครั้งแรก จากนั้นโปรแกรมก็จะทำการ Save ข้อมูลให้ใหม่โดยอัตโนมัติ



รูปที่ 3-28 แสดงหน้าจอตัวอย่างของโฮมเพจส่วนตัวของผู้ใช้

3.4.5.24 แนวความคิดในการสร้างโฮมเพจส่วนตัว

โฮมเพจส่วนตัวนี้ เพื่อเพิ่มความสะดวกสบายให้กับสมาชิกโดยที่สมาชิกของกระดานข่าวสามารถที่จะสร้างโฮมเพจของตัวเองขึ้นมาได้ โดยไม่จำเป็นจะต้องมีความรู้เกี่ยวกับการสร้างโฮมเพจ หรือการทำงานของโปรแกรมอื่นๆ เพียงแค่เลือกพื้นหลังจากที่สร้างขึ้นไว้ให้ และใส่ข้อความลงไป เลือกรูปแบบต่างๆ และใส่รูปภาพลงไป ก็จะได้โฮมเพจของตัวเองขึ้นมา นอกจากนี้ยังสามารถที่จะแก้ไขได้ด้วย

บทที่ 4

บทวิจารณ์และสรุป

4.1 ข้อดีต่างๆในการทำงานของกระดานข่าว

- กระดานข่าวที่สร้างขึ้นมาจะมีระบบความปลอดภัยให้กับผู้ใช้งาน โดยผู้ที่เข้ามาตั้งกระทู้หรือยุ่งเกี่ยวกับข้อมูลต่าง ๆ นั้นจะต้องสมัครเป็นสมาชิกหรือเป็นผู้ดูแลกระดานข่าวเท่านั้น เมื่อผู้ใช้งานต้องการเข้ามาใช้งานจะต้องทำการ Login ก่อน และเมื่อใช้งานเสร็จก็ให้ Logout ออกไป ผู้ที่ไม่ได้ทำการ Login จะทำได้เพียงแค่อ่านข้อมูลของหัวข้อกระทู้เท่านั้น แต่จะทำการตั้งกระทู้ หรือแก้ไขข้อมูลต่างๆไม่ได้
- ข้อมูลต่างๆบนกระดานข่าวนั้นจะถูกเก็บลงบนฐานข้อมูล ทำให้จัดการกับข้อมูลได้อย่างมีระเบียบ และเป็นระบบ เมื่อต้องการค้นหา, แก้ไข, ลบข้อมูลก็จะทำได้โดยง่าย กว่าที่เก็บข้อมูลลงบนไฟล์แบบทั่วไป และจากการเก็บข้อมูลลงบนฐานข้อมูลนี้เอง ทำให้เราเพิ่มประสิทธิภาพในการทำงานบางอย่างได้ เช่นการค้นหาข้อมูล เราสามารถแบ่งตามวันได้ และแบ่งตามห้องต่างๆได้ หรือ การแสดงกระทู้ทั้งหมดต่อหน้า เราสามารถกำหนดจำนวนได้ และ อื่นๆอีกมากมาย
- สมาชิกของกระดานข่าวสามารถที่จะแก้ไขข้อมูลของตัวเองได้ แม้กระทั่งรหัสลับ เพื่อเป็นการสนับสนุนให้จำรหัสลับได้ง่ายขึ้นและเปลี่ยนแปลงข้อมูลให้มีความทันสมัย
- การทำงานต่างๆของกระดานข่าว จะพยายามให้ผู้ใช้งานผ่านขั้นตอนในการทำงานให้น้อยที่สุด ให้ถึงจุดหมายเร็วที่สุด เพื่อให้เกิดความเข้าใจต่อ โปรแกรมได้ง่ายขึ้น และลดความยุ่งยากลง
- การตั้งกระทู้และการตอบกระทู้นั้น จะไม่รวมกันอยู่ในหน้าเดียวกัน เพื่อแยกให้เป็นระเบียบ หากจะดูการตอบกระทู้ ให้กดเลือกเข้าไปที่หัวข้อกระทู้นั้น แล้วจะเข้าไปหน้าของการตอบกระทู้ ซึ่งจะมีการตอบกระทู้ทั้งหมด และช่องให้กรอกเพิ่ม เพื่อให้ผู้ใช้งานดูข้อมูลได้อย่างเป็นระเบียบ
- ในส่วนของผู้ดูแลระบบก็เช่นเดียวกัน การทำงานต่างๆของผู้ดูแลระบบ จะพยายามที่จะให้สามารถทำงานได้โดยผ่านขั้นตอนให้น้อยที่สุด เพื่อความสะดวก และลดความสับสนของผู้ใช้งาน
- ผู้ดูแลกระดานข่าวจะมีสิทธิเหนือผู้ใช้งานทั่วไป และมีสิทธิเท่ากับผู้ดูแลกระดานข่าวคนอื่น ทำให้สามารถที่จะจัดการกำหนดค่าต่างๆได้เลย ไม่ต้องรอผู้ดูแลระบบทั้งหมดอนุญาต ทำให้เกิดความสะดวกมากขึ้น เช่น การเพิ่มสิทธิผู้ทั่วไปให้เป็นผู้ดูแลกระดานข่าว หรือการลดสิทธิผู้ดูแลกระดานข่าวคนอื่นให้เป็นผู้ใช้งานทั่วไปได้ ตามความเหมาะสม หรือลบชื่อผู้ใช้ออกจากสมาชิกของกระดานข่าว ซึ่งหากผู้ดูแลกระดานทำไปโดยความไม่เหมาะสมเอง ผู้ดูแลระบบทั้งหมดจะเป็นผู้จัดการเอง
- ผู้ใช้งานสามารถที่จะสร้าง โคมเพจได้เองโดยที่ไม่ต้องมีความรู้เกี่ยวกับการสร้างเลข และไม่จำเป็นต้องใช้เครื่องมือสร้าง โคมเพจตัวอื่น เพียงแค่ทำการเลือกรูปพื้นหลังตามที่จัดเอาไว้ให้ และเลือกข้อความและรูปภาพลง ไป แล้วเลือกรูปแบบต่างๆลงไป ก็สามารถสร้าง โคมเพจของตัวเองขึ้นมาได้

4.2 ข้อเสียต่างๆในการทำงานของกระดานข่าว

- การตั้งกระทู้ และการตอบกระทู้ นั้นยังไม่มีระบบตรวจสอบข้อความที่ไม่เหมาะสมที่มีประสิทธิภาพเพียงพอ ทำให้ไม่สามารถที่จะกันกระทู้ที่มีข้อความที่ไม่เหมาะสมได้เพียงพอ
- กราฟฟิกของโปรแกรมยังไม่สวยงามนัก เนื่องจากเกิดปัญหาขึ้นเมื่อนำโฮมเพจที่สร้างขึ้นนำรวมกับตัวโปรแกรมที่ใช้ซีจีไอเขียนขึ้น ทำให้ใช้ได้แค่กราฟฟิกแบบธรรมดา
- โปรแกรมกระดานข่าวยังไม่สามารถที่จะแก้ไขปัญหาในบางส่วนได้อย่างสมบูรณ์ เนื่องจากโครงสร้างของโปรแกรม ทำให้ในบางครั้งอาจจะเกิดปัญหาขึ้นได้



บทที่ 5

ภาคผนวก

5.1 การติดตั้งระบบที่ใช้งานต่างๆ

5.1.1 การติดตั้งระบบปฏิบัติการ (Linux)

ในขั้นตอนแรกเราจะทำการติดตั้งระบบปฏิบัติการ ซึ่งระบบปฏิบัติการที่เราเลือกใช้งานในระบบก็คือ ลินุกซ์ เพราะการทำงานของลินุกซ์อยู่ในรูปแบบของยูนิกซ์ (UNIX) ที่มีประสิทธิภาพในการทำงานค่อนข้างสูง ระบบสนับสนุนการทำงานของเซิร์ฟเวอร์ และค่อนข้างใช้กันอย่างแพร่หลายในปัจจุบัน

- เตรียมเครื่องคอมพิวเตอร์
- จัดหาต้นแบบโปรแกรมลินุกซ์ อาจจะเป็น CD-ROM หรือลงจาก FTP / HTTP Server
- บูทระบบเป็น LINUX เพื่อเริ่มทำการติดตั้ง
- เริ่มทำการติดตั้ง และ ทำสำเนาไฟล์ต่างๆลงบนฮาร์ดดิสก์ของท่าน
- ทำการติดตั้งเพิ่มเติมที่เรียกว่า First-time Setup

5.1.1.1 เตรียมเครื่องคอมพิวเตอร์

ฮาร์ดแวร์ที่สามารถใช้งานบนระบบลินุกซ์ได้จะมีดังนี้

- Intel-486DX, Pentium, Pentium-Pro, Pentium MMX, Pentium II
- RAM 16 MB ขึ้นไป
- ฮาร์ดดิสก์ IDE Hard disk ขนาด 2 GB
- Ethernet Controller
- SCSI Controller Adaptec
- VGA Card
- Serial Mouse และ PS/2 Mouse
- Modem

5.1.1.2 จัดหาต้นแบบโปรแกรม Linux

เราสามารถติดตั้ง Linux ได้หลายวิธีด้วยกัน

- ติดตั้งจาก CD-ROM
- ติดตั้งจาก FTP/HTTP Server
- ติดตั้งจาก Directory ที่มีข้อมูลของ Linux อยู่แล้ว

5.1.1.3 บูทระบบเป็น Linux เพื่อทำการติดตั้ง

ขั้นตอนนี้จะทำการบูทระบบของ Linux ขึ้นมาเพื่อเริ่มทำการติดตั้ง

- Boot จาก CD-ROM
- Boot จาก DOS/Windows
- Boot จากแผ่น Boot/Root

5.1.1.4 เริ่มทำการติดตั้ง และสำเนาไฟล์ต่างๆ ลงบนฮาร์ดดิสก์ของท่าน

- เลือกไครฟ์ที่จะทำการติดตั้ง Linux
- แบ่งพาร์ติชัน (Partition)

ซึ่งระบบ Linux จะมีระบบแบ่งพาร์ติชันโดยอัตโนมัติ เพื่อให้ความสะดวก โดยจะแบ่งเป็น 3 พาร์ติชัน คือ Root partition ขนาด 1000 MB, Swap partition ขนาด 128 MB และ Data partition

5.1.1.5 เลือกวิธีการติดตั้ง

เราเลือกวิธีการติดตั้ง Linux ได้หลายวิธีด้วยกัน

- ติดตั้งจาก CD-ROM
- ติดตั้งจาก FTP/HTTP Server
- ติดตั้งจาก Directory ที่มีข้อมูลของ Linux อยู่แล้ว

5.1.1.6 เลือกส่วนประกอบของ Linux

หลังจากเลือกรูปแบบการติดตั้งเรียบร้อยแล้ว จะเป็นการเลือกซอฟต์แวร์ซึ่งจะได้แก่ base, data, sis, server, source, desktop และ desktop-src

5.1.1.7 เริ่มการทำสำเนาไฟล์

หลังจากเลือกส่วนต่างๆเรียบร้อยแล้ว ก็จะทำสำเนาไฟล์ต่างๆลงในเครื่อง แล้วหลังจากนั้นก็ จะทำการ reboot

5.1.2 การติดตั้งระบบเครือข่าย

เมื่อเครื่องคอมพิวเตอร์ของเรามีการเชื่อมต่อทางด้านฮาร์ดแวร์เรียบร้อยแล้ว เราก็เริ่มทำการติดตั้ง โดยจะต้องติดตั้งค่าต่างๆดังนี้ IP Address, Gateway, DNS Server

5.1.3 การติดตั้ง WWW Server, Apache Server

5.1.3.1 WWW Server

WWW Server ปกติจะทำงานที่ Port 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ค้นหาว่า WWW Server Root อยู่ที่ไหน ซึ่งโดยทั่วไป RedHat จะอยู่ที่ /home/httpd
- นำไฟล์ HTML ที่เราต้องการทำงานไปใส่ไว้ในไดเรกทอรี htdocs
- หากมี CGI-Script ให้นำไปใส่ไว้ที่ <WWW Server Rot>/cgi-bin
- หากต้องแก้ไข Configuration ก็สามารถแก้ไขได้ที่ <WWW Server Root>/etc
- Log File แสดงการทำงานจะอยู่ที่ <WWW Server Root>/logs
- เรียกใช้งาน โปรแกรม httpd โดยใช้ Script apachectl ซึ่งจะอยู่ที่ <WWW Server Root>/bin/
- เมื่อ WWW Server ทำงานแล้ว สามารถใช้ WWW Browser เรียกใช้ <http://<ชื่อเครื่องของท่าน>>

5.1.3.2 การติดตั้ง Apache WWW Server

- ทำการหาไฟล์โปรแกรม Apache จะชื่อ apache_VERSION.tar.gz
- unpack ไฟล์และทำการติดตั้ง
- จะได้ WWW Server ซึ่งจะมี WWW Server Root อยู่ที่ /usr/local/etc/httpd

5.1.4 การติดตั้งฐานข้อมูล (MYSQL)

การติดตั้ง MYSQL จะมีส่วนของ Server และ Client

MYSQL Server

- หาโปรแกรม MYSQL ซึ่งสามารถหาได้ที่ <http://www.tcx.se/download.html> มีทั้ง Source และ binary
- ติดตั้งโปรแกรม

MYSQL Client

- ติดตั้ง API Module ที่ช่วยในการเขียนโปรแกรมติดต่อกับ MYSQL ซึ่งมี 2 Module คือ MYSQL Module ซึ่งเป็น Module ของ MYSQL และ DBI ซึ่งเป็น Module มาตรฐาน

5.1.5 การติดตั้ง MYSQL Grant Table

Grant Tables คือ ตารางที่เก็บข้อมูลในการกำหนดสิทธิการใช้งานฐานข้อมูลของผู้ใช้ เพื่อเหตุผลทางด้านความปลอดภัย ซึ่งต้องรันคำสั่งเพื่อสร้างฐานข้อมูลเหล่านี้

เมื่อคำสั่งทำงานเรียบร้อยแล้วจะได้ฐานข้อมูลที่ชื่อว่า MYSQL ภายในฐานข้อมูลจะประกอบด้วย ตาราง user, db และ host ส่วนค่าเริ่มต้นที่กำหนดมาให้มีดังนี้

- ฐานข้อมูลที่ชื่อว่า test หรือนำหน้าด้วย test_ จะอนุญาตให้ทุกคนสามารถใช้ได้
- root เป็น superuser มีสิทธิในการทำอะไรได้ทุกอย่าง ซึ่งยังไม่ได้กำหนดรหัสผ่าน ต้องกำหนดรหัสผ่านให้กับ root เป็นอันดับแรก
- บอกให้ MYSQL อ่านค่า grant tables ใหม่ เพื่อให้ค่าที่กำหนดใหม่สามารถใช้งานได้

5.2 พื้นฐานการใช้งาน Perl

5.2.1 ที่อยู่ของ Perl

ในระบบ Unix ถ้าคุณใช้คำสั่ง `$ whereis perl` จะได้รับรายงานมาว่า `/usr/bin/perl` ซึ่ง `/usr/bin/perl` นั่นก็คือพาทิไคลเรททอรีของตัวแปลภาษานั้นเอง และถ้าคุณสืบรายงานคุณว่า `no perl` หรือหาไม่พบในพาทิไคลเรททอรี ก็ให้สอบถามไปยังผู้ดูแลระบบ (Administrator) หรือทำการติดตั้งถ้าเป็นระบบของคุณเอง (Linux เรียก setup หรือ unzip)

5.2.2 การรันโปรแกรม Perl ในระบบ Unix

สำหรับระบบยูนิกซ์ ถ้าคุณใช้ยูนิกซ์ คุณสามารถสร้างได้โดยใช้โปรแกรม VI อีดิเตอร์ สร้างไฟล์ใหม่ที่มีนามสกุล `.pl` ขึ้นมา โดยพิมพ์คำสั่งต่อไปนี้ที่ยูนิกซ์เชลล์

```
$ vi easy.pl
```

หากระบบยูนิกซ์ของคุณไม่มี vi อีดิเตอร์ คุณก็สามารถใช้อีดิเตอร์ตัวอื่นหรือใช้คำสั่ง `cat >easy.pl` ก็ได้ จากนั้นให้คุณพิมพ์โปรแกรมด้วยภาษา Perl ลงไป สิ่งหนึ่งที่คุณต้องจำเมื่อคุณจะเขียนโปรแกรม Perl บนยูนิกซ์ก็คือ บรรทัดบนสุดของไฟล์คุณจะต้องเขียนดังนี้

```
#!/usr/bin/perl
```

```
print "Hello Perl\n";
```

บรรทัดบนสุดนี้เป็นการบอกถึงพาทิไคลเรททอรีของตัวแปลภาษาของ Perl ซึ่งส่วนมากแล้ว ตัวแปลภาษาของ Perl จะถูกเก็บอยู่ในไคลเรททอรีในข้างต้น แต่ถ้าคุณไม่แน่ใจ ก็ให้คุณพิมพ์คำสั่ง `whereis perl` เพื่อตรวจสอบอีกครั้ง เมื่อคุณพิมพ์โปรแกรมเสร็จแล้ว ให้ทำการเซฟโดยกดปุ่ม `ESC + ":" + wq` และ `Enter` เพื่อทำการเซฟและออกจาก vi จากนั้นทำการรัน โปรแกรม โดยพิมพ์ดังนี้

```
$ perl easy.pl
```

ถ้าโปรแกรมมีข้อผิดพลาด ตัวแปลภาษาจะแสดงข้อความบอกถึงจุดที่ผิดพลาดมาให้เพื่อแก้ไข หรือคุณอาจจะใช้วิธีการรัน โดยการแปลงไฟล์ให้อยู่ในโหมด Execute เสียก่อน โดยพิมพ์คำสั่งต่อไปนี้

```
$ chmod +x easy.pl
```

เมื่อคุณทำการรัน โปรแกรม คุณก็พิมพ์แค่ `easy` เฉย ๆ โปรแกรมก็จะรันโดยอัตโนมัติ

5.2.3 การเรียกใช้ perl จาก Command line ของ DOS

เนื่องจาก Perl ก็คล้ายโปรแกรมบน DOS โปรแกรมหนึ่ง เวลาที่ทำงานจะทำงานอยู่เบื้องหลัง ถ้าไม่มีการเรียกใช้ ก็จะไม่ทำงาน เราสามารถเรียกใช้ใน Perl ด้วย Command line แบบคำสั่ง DOS ได้ ถ้าลองใส่ข้อมูลไปดังนี้ ให้คุณเปิด Notepad แล้วพิมพ์ print "hello my perl"; จากนั้น save ไฟล์ โดยใช้ชื่อ test.pl ไว้ที่ไหนก็ได้ยกเว้น desktop แล้วเข้าไปที่ DOS prompt ใช้คำสั่ง cd ของ DOS ย้ายไป directory ที่คุณเก็บไฟล์ test.pl ไว้ พิมพ์คำว่า perl test.pl ที่ DOS prompt จากนั้นกด Enter 1 ครั้ง คุณก็จะพบกับคำว่า hello my perl ที่ได้พิมพ์ไป

5.2.4 สร้างโปรแกรม CGI อย่างง่าย ให้แสดงผลที่หน้าจอบราวเซอร์

เติม code เหล่านี้ลงไปให้ได้ผลทั้งหมดออกมาดังนี้นะครับ

```
print "Content-type:text/html\n";
print "<html><head><title></title></head>\n";
print "<body>\n";
print "hello my perl\n";
print "</body></html>\n";
```

หมายเหตุ ส่วนที่เป็น \n คือคำสั่งการขึ้นบรรทัดใหม่ ถ้าไม่ใส่ก็ไม่เป็นไรแต่ผลของโปรแกรมที่ส่งมาให้บราวเซอร์ เมื่อ view source ดูเห็นติดกันหมดเป็นบรรทัดเดียว

เมื่อเสร็จแล้วให้ save as เป็นไฟล์ test2.pl โดยไฟล์นี้ให้เก็บไว้ใน directory ที่คุณกำหนดไว้ให้เป็น cgi-bin หรือ directory อื่นที่สามารถ execute ไฟล์ได้ เรียกในบราวเซอร์ดังนี้ <http://you-server-name/cgi-bin/test2.pl>

เมื่อคุณกดปุ่ม Enter บราวเซอร์จะรับที่อยู่ URL ของ test2.pl ไปส่งให้ web server จากนั้น web server จะเรียก perl.exe ซึ่งเป็นตัวแปลภาษา perl ให้ทำการแปลคำสั่งในไฟล์ test2.pl เมื่อ perl.exe แปลเสร็จแล้วจะส่งผลออกมาให้ web server และทำการปิดตัวเองลง เมื่อ web server ได้รับผลมาแล้ว จะส่งผลนั้นกลับไปให้บราวเซอร์ที่เรียกในรูปแบบของ html ไฟล์ เนื่องจากใน test2.pl มีการกำหนดคำสั่งด้วย print "Content-type:text/html\n"; เพราะฉะนั้นการที่จะเขียน cgi ด้วย perl ให้แสดงผลที่หน้าจอบราวเซอร์จะต้องใส่คำสั่ง print "Content-type:text/html\n"; ลงไปด้วย

5.2.5 การลดปริมาณการใช้คำสั่ง print

```
print "Content-type:text/html\n";
print <<myHTML;
<html><head><title>เขียน CGI ด้วย Perl</title></head>
```

```
<body bgcolor=\"black\" text=\"white\">
<center>
ข้อความที่ต้องการแสดงบนหน้าจอบราวเซอร์ ถ้ามีเครื่องหมาย \@ หรือเครื่องหมาย \"<br>
คุณจะต้องใช้เครื่องหมาย \ ใส่วางหน้าเครื่องหมายดังกล่าวด้วย ถ้าไม่ใส่จะเกิด error ขึ้น<br>
ในการใช้ภาษา Perl ถ้าคุณต้องการ ใส่ comment ให้ใช้เครื่องหมาย # ใส่วางหน้าบรรทัดที่<br>
ต้องการ comment ดูตัวอย่างที่กล่าวมาแล้วทั้งหมดได้ใน source code ของไฟล์นี้ (test4.pl)<br>
</center>
</body></html>
myHTML
```

5.2.6 ชนิดของตัวแปร

มีการกำหนดตัวแปรเพียงไม่กี่ชนิดในภาษา Perl ซึ่งต่างจากภาษาอื่น ๆ Perl ได้กำหนดชนิดของตัวแปรเอาไว้ 2 ประเภทใหญ่ ๆ คือ

สเกลล่า (Scalar)

อะเรย์ (Array)

ตัวแปรแบบ Scalar ตัวแปรแบบ Scalar เป็นตัวแปรที่ใช้สำหรับเก็บข้อมูลเดี่ยว ๆ ที่เป็นตัวเลข (Integer) หรือข้อความ (String) วิธีการกำหนดตัวแปรแบบ Scalar นั้น ไม่จำเป็นต้องประกาศตัวแปรก่อน เหมือนกับภาษาซีหรือภาษาปาสคาล เมื่อคุณต้องการใช้ คุณก็สามารถสร้างและกำหนดค่าให้กับมันได้เลย โดยตัวแปรแบบ Scalar นั้นจะต้องใช้เครื่องหมายดอลลาร์ (\$) นำหน้า เช่น

```
$name = "kwangd";
$age = 36;
$address = "Nokomphom 73210";
```

จากข้างต้น เป็นตัวอย่างการประกาศตัวแปรแบบ Scalar ที่ชื่อ \$name , \$age และ \$address ตัวแปรภาษา Perl จะรู้ว่าตัวแปรนั้นเป็นแบบ Integer หรือแบบ String ก็ต่อเมื่อคุณกำหนดค่าให้กับมัน ดังในตัวอย่างข้างต้น และในการกำหนดตัวแปรของ Perl คุณจะกำหนดไว้ที่ส่วนของโปรแกรมก็ได้ ไม่จำเป็นว่าจะต้องอยู่บนสุดของโปรแกรมเท่านั้น ถึงแม้จะเป็นบรรทัดสุดท้าย คุณก็สามารถกำหนดตัวแปรได้เช่นกัน

ในการใช้เครื่องหมายคำพูดที่เป็น "...." และ '.....' นั้น มีความหมายเดียวกัน เช่น

```
$name = "kwangd";
```

มีความหมายเช่นเดียวกับ

```
$name = 'kwangd';
```

และถ้าคุณใช้ตัวแปรตัวเดียวกัน เก็บข้อมูลซ้ำสอง โดยที่เป็นข้อมูลคนละประเภท เช่น

```
$name = "kwangd";
```

```
$name = 36;
```

ตัวแปร \$name จะเก็บข้อมูลครั้งล่าสุดเอาไว้ นั่นคือ 36

ตัวแปรแบบ Array ตัวแปรแบบอะเรย์ เป็นการเก็บข้อมูลเป็นชุดหรือเป็นแถวเรียงต่อกัน.. การกำหนดตัวแปรแบบอะเรย์ในภาษา Perl คุณจะต้องใช้เครื่องหมาย @ นำหน้าชื่อของตัวแปรนั้น

```
@array = ("black", "white", "yellow", "green");
```

จากข้างต้น อะเรย์นี้มีชื่อว่า array มีสมาชิกในอะเรย์ 4 ตัวด้วยกัน ซึ่งเป็นข้อมูลแบบ String นั่นคือ black , white , yellow และ green สมาชิกแต่ละตัวที่อยู่ในอะเรย์นั้น มีคุณสมบัติเป็นตัวแปรแบบ Scalar เพราะฉะนั้น การอ้างถึงสมาชิกแต่ละตัวที่อยู่ในอะเรย์ เราจะต้องใช้เครื่องหมาย \$ และบอกตำแหน่งของสมาชิก ซึ่งจะเริ่มตั้งแต่ 0 , 1 , 2 และ 3 ไปเรื่อย ๆ (เช่นเดียวกับภาษาซี)

เมื่อคุณต้องการแสดงข้อมูลที่อยู่ในอะเรย์ คุณสามารถทำได้โดยตรง โดยใช้คำสั่ง print

ตัวแปรแบบ Associative Array

Associative Array เป็นอะเรย์อีกประเภทหนึ่ง ลักษณะของอะเรย์แบบนี้ จะเป็น "อะเรย์แบบจับคู่" นั่นคือ สมาชิกตัวที่ 0 จะคู่กับสมาชิกตัวที่ 1 , สมาชิกตัวที่ 2 จะคู่กับสมาชิกตัวที่ 3 การประกาศตัวแปรอะเรย์แบบนี้ เราจะใช้เครื่องหมาย % นำหน้าตัวแปรนั้น เช่น

```
%age = ("kwangd", 27, "kai", 30, "Bee", 34);
```

จากการประกาศตัวแปรในข้างต้น เมื่อเราอ้างถึงสมาชิกตัวแรกของอะเรย์ เราก็จะได้สมาชิกตัวที่ 1

```
%age = ("kwangd", 27, "kai", 30, "bee", 34);
```

```
print " $age{'kwangd'}\n";
```

```
print " $age{'kai'}\n";
```

```
print " $age{'bee'}\n";
```

ผลการรันโปรแกรม

```
27
```

```
30
```

```
34
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่า การอ้างถึงสมาชิกแต่ละตัวจะใช้เครื่องหมาย \$ นำหน้าสมาชิกแต่ละตัว ตัวแปรแบบ Associative Array นี้จะใช้ในการรับค่าและจัดการข้อมูลจากฟอร์มในการเขียนโปรแกรมด้วย CGI

5.2.7 หลักการกำหนดชื่อให้กับตัวแปร

การกำหนดชื่อให้กับตัวแปรนั้น มีกฎเพียงไม่กี่ข้อ ซึ่งถ้าคุณเคยผ่านการเขียนโปรแกรมมาก็สามารถทำความเข้าใจได้ง่าย ดังนี้.. 1. ถ้าเป็นตัวแปรแบบ Scalar ให้ใช้เครื่องหมาย \$ และถ้าเป็นตัวแปรแบบอะเรย์จะต้องใช้เครื่องหมาย @ 2. อักษรที่ต่อจากเครื่องหมาย \$ และ @ จะต้องเป็นตัวอักษรเท่านั้น 3. ไม่สามารถใช้สัญลักษณ์ใด ๆ เป็นชื่อของตัวแปรได้ ตัวอย่างการกำหนดชื่อให้กับตัวแปรต่อไปนี้ ให้คุณลองคิดว่าถูกต้องหรือไม่... (ถ้าคุณไม่แน่ใจ ให้คุณลองพิมพ์โปรแกรมและทดลองรันดู)

5.2.8 การรับข้อมูลจากผู้ใช้ (Standard Input)

ในการรับข้อมูลทางคีย์บอร์ดจากผู้ใช้ในภาษา Perl ไม่มีฟังก์ชัน scanf() หรือ readln() เหมือนกับภาษาอื่น แต่จะใช้ <STDIN> ซึ่งเป็น Standard Input สำหรับรับข้อมูลจากคีย์บอร์ด ให้คุณลองพิมพ์โปรแกรม stdin1.pl ต่อไปนี้ และทดลองรันเพื่อดูผลการทำงาน

```
<STDIN>
```

ผลการรันโปรแกรม

```
C:\> perl stdin1.pl
```

```
hello
```

```
C:\>
```

เมื่อคุณรันโปรแกรม คุณจะพบว่าเคอร์เซอร์จะกระพริบเพื่อรอรับการกด ให้คุณพิมพ์อะไรก็ได้ลงไป จากนั้นโปรแกรมก็จะสิ้นสุดการทำงาน <STDIN> เป็นการรับข้อมูลจากคีย์บอร์ด ถ้าคุณต้องการให้สิ่งที่พิมพ์ลงไปเก็บไว้ที่ตัวแปร คุณจะต้องเขียนโค้ดโปรแกรมดังนี้

```
$a = <STDIN>;
```

เมื่อคุณต้องการแสดงค่าที่อยู่ในตัวแปร \$a ออกทางจอภาพ ให้คุณใช้คำสั่ง print

```
print "value is $a\n";
```

โปรแกรมต่อไปนี้เพิ่มเติมจากโปรแกรม stdin1 เป็นการรับชื่อจากผู้ใช้ เมื่อคุณพิมพ์ชื่อลงไป

โปรแกรมจะบอกชื่อคุณกลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.9 Perl และการคำนวณ

การคำนวณในภาษา Perl นั้น ได้กำหนดโอเปอเรเตอร์ทางคณิตศาสตร์เอาไว้ 4 ตัวด้วยกัน นั่นคือ บวก (+), ลบ (-), คูณ (*) และหาร (/) สำหรับการใช้งาน

5.2.9.1 ลำดับของการคำนวณ

ลำดับการคำนวณของ Perl ก็เช่นเดียวกับการคำนวณโดยทั่วไป ซึ่งในนิพจน์จะกระทำเครื่องหมายที่มีน้ำหนักมากกว่า เป็นอันดับแรก Perl มีโอเปอเรเตอร์อยู่ 4 ตัว เพราะฉะนั้นในหนึ่งนิพจน์ การคำนวณจะเริ่มจากเครื่องหมาย คูณ , หาร ก่อนเครื่องหมาย บวกและลบ ดังเช่น โปรแกรม think2.pl ต่อไปนี้

5.2.9.2 เงื่อนไขและการวนลูป

เงื่อนไขคืออะไร

เมื่อคุณจำเป็นต้องตัดสินใจเลือกสิ่งใดสิ่งหนึ่ง หรือทางใดทางหนึ่ง นี้จะเรียกว่า "เงื่อนไข" หรือ กอนดิชัน (Condition) ผลลัพธ์จากเงื่อนไขนั้นจะมีเพียงสิ่งเดียวหรือทางเดียวเท่านั้น ในภาษา Perl มีรูปแบบของเงื่อนไข 5 ประเภทด้วยกัน นั่นคือ

if

if ... Else

while

เราจะเริ่มศึกษากันที่เงื่อนไขประเภทแรก

เงื่อนไขแบบ if

if เป็นรูปแบบเงื่อนไขที่ง่ายที่สุดในการเขียนโปรแกรม ไม่ว่าจะเขียนภาษาซีหรือภาษาปาสคาล หากพูดถึงเรื่องของเงื่อนไข จะต้องขึ้น if ก่อน นั่นก็เพราะว่า รูปแบบของเงื่อนไข if นั้นสามารถเห็นได้ชัดเจน รูปแบบของเงื่อนไข if มีลักษณะดังนี้

```
if (เงื่อนไข) { คำสั่งหรือสเตตเมนต์ }
```

การเขียนเงื่อนไขแบบ if นี้ คุณจะต้องเขียนให้อยู่ในรูปแบบข้างต้น ซึ่งจะต้องประกอบด้วยเครื่องหมายปีกกาเปิดและปิด {...} ทุกครั้ง ตัวอย่าง โปรแกรม if1.pl ต่อไปนี้ แสดงการใช้เงื่อนไขแบบ if

```
โปรแกรม if1.pl
```

```
$num = 20;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( $num == 20 ) {

print "number is 20";

}

```

ผลการรันโปรแกรม

number is 20

จากโปรแกรมในข้างต้น กำหนดให้ \$num มีค่า 20 จากนั้นเงื่อนไข if จะทำการเช็คดูว่าตัวแปร \$num มีค่า 20 จริงหรือเปล่า ถ้าเป็นจริงดังว่า ก็จะทำการสั่งที่อยู่ในเครื่องหมายปีกกาทันที จากนั้นก็จบโปรแกรม คราวนี้ เราสามารถสรุปได้แล้วว่า if นั้น จะทำการสั่งในบรรทัดต่อไปก็ต่อเมื่อเงื่อนไขเป็นจริง (TRUE) ดังเช่นในตัวอย่าง if1.pl ค่า \$num เก็บค่า 20 เพราะฉะนั้นจึงทำให้เงื่อนไขเป็นจริง ประโยคที่อยู่ถัดไปจึงถูกกระทำ

ในภาษาซี ถ้าประโยคที่อยู่ถัดจาก เงื่อนไข if มีเพียงบรรทัดเดียว คุณสามารถเขียน ในลักษณะนี้ได้

```
if( num == 20 ) printf("number is 20");
```

แต่ในภาษา Perl คุณจะต้องใส่เครื่องหมายปีกกาครอบทุกครั้ง ถึงแม้ว่าจะมีเพียงประโยคเดียวก็ตาม

จะเห็นได้ว่า การเปรียบเทียบตัวเลขและข้อความนั้นจะใช้เครื่องหมายต่างกัน นั่นคือ ถ้าเป็นตัวเลขเรา จะใช้เครื่องหมาย == แต่ถ้าเป็นข้อความเราจะใช้ eq ซึ่งย่อมาจาก equal (แปลว่า "เท่ากับ") ซึ่งนอกจากเครื่องหมาย == และ eq แล้ว ยังมีการเปรียบเทียบอีกหลายแบบ ดังตารางที่ 3.1 ต่อไปนี้

	Numbers	Strings
\$x เท่ากับ \$y	\$x == \$y	\$x eq \$y
\$x ไม่เท่ากับ \$y	\$x != \$y	\$x ne \$y
\$x มากกว่า \$y	\$x > \$y	\$x gt \$y
\$x มากกว่าหรือเท่ากับ \$y	\$x >= \$y	\$x ge \$y
\$x น้อยกว่า \$y	\$x < \$y	\$x lt \$y
\$x น้อยกว่าหรือเท่ากับ \$y	\$x <= \$y	\$x le \$y

ตารางที่ 3-1 แสดงเครื่องหมายเปรียบเทียบตัวเลขและข้อความ

ตัวอย่างโปรแกรม chop1.pl ต่อไปนี้ คัดแปลงจากโปรแกรม if2.pl เป็นตัวอย่างการรับชื่อ จากผู้ใช้ทางคีย์บอร์ด ถ้าผู้ใช้พิมพ์คำว่า "yoyo" โปรแกรมจะแสดงข้อความตอบกลับมาว่า hello , yoyo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม chop1.pl

```
print "Enter your name : ";
$name = <STDIN>;
chop($name);
if ($name eq "yoyo"){

print "\nhello , $name";

}
```

ผลการรันโปรแกรม

```
Enter your name : yoyo
hello , yoyo
```

จากโปรแกรม chop1.pl ในข้างต้น เมื่อคุณพิมพ์ชื่อลงไปว่า yoyo โปรแกรมจะทำการตัดอักขระตัวสุดท้ายด้วยฟังก์ชัน chop() จากนั้นจึงเข้าสู่เงื่อนไข if เพื่อทำการตรวจสอบต่อไป คุณทราบหรือไม่ว่าทำไมต้องทำการตัดอักขระตัวสุดท้ายออก.... นั่นก็เพราะว่า พอคุณพิมพ์ชื่อลงไป คุณจะต้องเกาะ Enter เมื่อคุณพิมพ์เสร็จ รหัสของปุ่ม Enter หรือ newline character ก็จะติดไปกับข้อความที่คุณพิมพ์ลงไปด้วย นั่นคือ ถ้าคุณพิมพ์คำว่า "yoyo" และกดปุ่ม Enter ตัวแปร \$name จะเก็บอักขระทั้งหมด 5 ตัวนั่นคือ y , o , y , o และ \n หรือ newline character นั่นเอง ถ้าคุณนำตัวแปร \$name ไปเปรียบเทียบกับในเงื่อนไข If เลย เงื่อนไขจะไม่เป็นจริง เพราะข้อมูลที่อยู่ในตัวแปร \$name นั้นมีเกินมา 1 ตัว เพราะฉะนั้น เราจึงต้องเรียกฟังก์ชัน chop() ซึ่งทำหน้าที่ตัดตัวอักขระที่อยู่ท้ายสุดของข้อความออกไป...

แต่สำหรับการรับข้อมูลที่เป็นตัวเลขนั้น ไม่จำเป็นต้องตัดรหัส \n ออกเหมือนกับการรับข้อความ
ทางเลือกที่มากกว่า 1 เงื่อนไข

จากตัวอย่างการใช้ if ที่ผ่านมา เราได้เปรียบเทียบเพียงแค่ 1 เงื่อนไขเท่านั้น ถ้าในกรณีที่เรามีมากกว่า 1 เงื่อนไขที่จะต้องเปรียบเทียบพร้อมกัน เราจะใช้เครื่องหมาย || (OR) และ && (AND) ดังนี้

- การ OR ก็คือ "หรือ" หมายถึง ถ้าเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง if จะกระทำ เราจะใช้เครื่องหมาย ||
- การ AND ก็คือ "และ" หมายถึง ทุก ๆ เงื่อนไขจะต้องเป็นจริง if จึงจะกระทำ เราจะใช้เครื่องหมาย &&

ลองพิจารณาตัวอย่าง โปรแกรม if3.pl

โปรแกรม if3.pl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$a = 10;
if ( ($a==20) || ($a==10) ) {

print "that is true";

}

```

จากโปรแกรมในข้างต้น กำหนดให้ตัวแปร \$a มีค่าเท่ากับ 10 จากนั้นเข้าสู่เงื่อนไข if โดย if จะตรวจสอบว่า "\$a เท่ากับ 20 หรือ \$a เท่ากับ 10 หรือไม่" เมื่อ \$a มีค่า 10 จึงทำให้เงื่อนไขที่สองเป็นจริง (\$a==10) เมื่อคุณรัน โปรแกรมนี้ จึงแสดงข้อความ "that is true" บนจอภาพ

คราวนี้ ลองมาพิจารณาโปรแกรม if4.pl ต่อไปนี้

```

โปรแกรม if4.pl

$a=50;
if ( ($a>=50) && ($a==50)){

print "that is true";

}

```

จากโปรแกรมในข้างต้น คราวนี้เราให้ตัวแปร \$a เก็บค่า 50 เอาไว้ จากนั้นเงื่อนไข if จะทำการตรวจสอบว่า "\$a มากกว่าหรือเท่ากับ 50 และ \$a เท่ากับ 50 หรือไม่" เราได้กำหนดให้ \$a มีค่า 50 เพราะฉะนั้น เงื่อนไขทั้ง 2 นี้เป็นจริงพร้อมกัน จึงทำให้ข้อความ that is true แสดงบนจอภาพ ถ้าเรากำหนดให้ \$a มีค่าเป็น 60 ละ จะเกิดอะไรขึ้น....

ถ้า \$a เป็น 60 เงื่อนไขแรกเท่านั้นที่เป็นจริง แต่เงื่อนไขที่สองเป็นเท็จ เงื่อนไขเป็นจริงเพียงเงื่อนไขเดียว เพราะฉะนั้น เงื่อนไข if จึงไม่แสดงข้อความ "that is true"...

if และ Else

การใช้ else จะใช้ในกรณีเงื่อนไขที่ไม่ตรงกับที่กำหนดใน if คือ นอกเหนือจาก if แล้ว จะให้ทำอะไร เช่น โปรแกรมต่อไปนี้

```

โปรแกรม ifelse1.pl

$name = "yoyo";
if ( $name eq "yoyo" ) {

print "hello , yoyo";

}else{

```

```
print "good bye";
```

```
}
```

จากโปรแกรม จะแสดงข้อความว่า "hello , yoyo" แต่ถ้าเราเปลี่ยนให้ตัวแปร \$name เป็นคำอื่น โปรแกรมก็จะแสดงข้อความว่า "Good bye" ดังโปรแกรมต่อไปนี้

โปรแกรม ifelse2.pl

```
$name = "yooooo";
```

```
if ( $name eq "yoyo" ) {
```

```
print "hello , yoyo";
```

```
}else{
```

```
print "good bye";
```

```
}
```

while

หากคุณเคยเขียนโปรแกรมภาษา C มาก่อน คุณจะสามารถใช้เงื่อนไขนี้ได้ง่ายมาก ลักษณะการใช้ while คล้ายกับการใช้ if แต่ while จะใช้ในการวนลูปเสียเป็นส่วนใหญ่ ดัง โปรแกรม while1.pl ต่อไปนี้

โปรแกรม while1.pl

```
$count = 0;
```

```
while ($count != 10){
```

```
print "$count\n";
```

```
$count++;
```

```
}
```

โปรแกรมในข้างต้น จะพิมพ์ค่าตัวแปร \$count โดยจะตรวจดูว่า ตัวแปร \$count เท่ากับ 10 หรือไม่ ถ้าไม่เท่ากับ 10 ก็จะบวกค่าให้กับตัวแปรทีละ 1 ไปเรื่อย ๆ

ลูปในภาษา Perl

ในภาษา Perl มีคำสั่งวนลูปอยู่ไม่กี่คำสั่ง คำสั่ง while ก็เป็นอันหนึ่ง แต่ผมขอยกให้เป็นเงื่อนไขไป เพราะมันเกี่ยวกับเงื่อนไขจริง ๆ นี้ครับ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งวนลูปที่เราจะมาศึกษากัน มีดังนี้

for

foreach

until

ลูปแบบ for

ลูปที่ง่ายและเบสิกที่สุดเห็นจะเป็นลูปของ for เหมือนกับการเขียน โปรแกรมด้วยภาษาซีเขียนเลขครับ หากเคยศึกษาซีมาก่อน ลักษณะของลูปมีดังนี้

for(ตัวแปร=ค่าเริ่มต้น ; เงื่อนไข ; เปลี่ยนค่า) { คำสั่งหรือประโยคที่ต้องการให้ทำ }

ตัวอย่างการใช้ลูปแบบ for คุณจะต้องระวังให้ตัวอย่างหนึ่งในเรื่องของตัวแปร จะพบมาก เมื่อคุณเขียน ลูปแบบนี้ โดยไม่ได้เครื่องหมาย \$ ให้กับตัวแปรที่กำหนด เพราะคุณเคยชินกับภาษาซีมาก่อน

โปรแกรม for1.pl

```
Sname= "kwangd DevMan";
for ($count=0;$count<=10;$count++){
print "$name\n";
}
```

เมื่อคุณรัน โปรแกรม for1.pl คุณจะพบกับคำว่า kwangd DevMan ทั้งหมด 10 ครั้ง เงื่อนไขในการใช้ลูป for อีกอย่างหนึ่งก็คือ คุณจะต้องใส่วงเล็บปีกกา {} ทุกครั้ง ถึงแม้จะมีเพียงคำสั่ง เดียวก็ตาม.....

ลูปแบบ foreach

ลูปแบบ foreach นี้จะใช้สำหรับจัดการสมาชิกใน Array ลักษณะของลูปแบบ foreach นั้น จะอ่านอะ เรย์มาเก็บไว้ในตัวแปรแบบสเกลาร์ทีละตัว ๆ จนกระทั่งครบทุกตัว เช่น

โปรแกรม foreach1.pl

```
@data = ("hello1","hello2","hello3");
foreach $a (@data){
print "$a\n";
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคุณรัน โปรแกรม `foreach1.pl` คุณจะพบกับผลดังนี้

hello1

hello2

hello3

จะเห็นได้จากโปรแกรมว่า ตัวแปรอะเรย์ `@data` จะเก็บสมาชิก `hello1` , `hello2` และ `hello3` เอาไว้ เมื่อเข้าสู่รูปแบบ `foreach` แล้ว

5.2.10 CGI สคริปต์

การเขียน CGI ด้วยภาษา Perl เราสามารถทำได้ โดยการเพิ่มรายละเอียดลงไปโปรแกรม เพื่อให้โปรแกรมสามารถทำงานเป็น CGI ได้ สิ่งแรกที่คุณจะต้องใส่ไว้ในบรรทัดที่ 2 ของการเขียนโปรแกรม CGI คือ การบอกรูปแบบของเอกสาร ดังนี้

```
#!/usr/bin/perl
print "Content-type:text/html\n\n";
```

ในข้างต้น จะเห็นว่า เราได้ใส่บรรทัด `"Content-type:text/html\n\n"` เป็นการกำหนด MIME Header เพื่อบอกให้เบราว์เซอร์ทราบว่า ไฟล์นี้เป็นไฟล์แบบใด

และในการเขียน CGI ด้วยภาษา C บนยูนิกซ์ คุณก็สามารถทำได้โดยใช้ลักษณะเดียวกัน แต่คุณไม่ต้องเขียน `#!/usr/bin/perl` ไว้ในบรรทัดแรก เพราะภาษา C จะต้องถูกคอมไพล์เพื่อนำไปใช้ ไม่เหมือนกับภาษา Perl ที่สามารถนำไปใช้ได้ทันที ตัวอย่างนี้ เป็น CGI ที่เขียนด้วยภาษา C บนยูนิกซ์

```
#include<stdio.h>

void main(){
printf("Content-type:text/html\n\n");
}
```

การรัน CGI ใน account

ในที่นี้ต้องขอยกตัวอย่าง account ที่เป็นแบบ shell หรือยูนิกซ์ เพราะ server ที่ `comsc2` นี้เป็นยูนิกซ์ เมื่อคุณล็อกอินมาที่ account ของคุณแล้ว (เช่นการ telnet) ให้คุณตรวจสอบดูก่อนว่ามีตัวแปลภาษา Perl อยู่ในระบบหรือเปล่า โดยพิมพ์คำสั่ง `whereis perl` ที่เชลล์

```
$ whereis perl [ENTER]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากในระบบมีตัวแปลภาษา Perl ระบบจะแสดงพาธออกมาให้ (ถ้าไม่มีคุณควรถามไปยังผู้ดูแลระบบ) จากนั้น ให้คุณถามไปยังผู้ดูแลระบบของคุณก่อนว่า คุณจะสามารถใช้ CGI บน Server นี้ได้อย่างไร นามสกุลของไฟล์จะต้องเป็นอะไร และจะเก็บไว้ในไดเรกทอรีใด

ที่ Thaitop ที่ผมเก็บ โหมดเพจนี้อยู่ เป็น NT4 และสามารถ CGI ที่เขียนด้วยภาษา Perl ได้ โดยจะต้อง กำหนดนามสกุลของไฟล์ให้เป็น .plx และเก็บไว้ใน /cgi-bin ซึ่งถ้าคุณใช้ UNIX พวก RedHat คุณอาจ จะต้องกำหนดนามสกุลของไฟล์ให้เป็น .cgi พร้อมกับเปลี่ยนโหมดไฟล์ให้เป็นไฟล์ในโหมด UNIX เพราะถ้าคุณเขียนมันขึ้นมาด้วย Notepad และ Upload ขึ้นไปบน Server ที่เป็น UNIX คุณจะรัน โปรแกรม CGI นั้นไม่ได้ เพราะเป็นไฟล์ที่สร้างจากคนละระบบกัน คุณจะต้องใช้โปรแกรม Convert แปลงจากไฟล์ในระบบ Windows ให้เป็นระบบ Unix เสียก่อน

การรันและทดสอบ โปรแกรมในบทนี้ คุณจะต้องปฏิบัติให้ถูกต้องตามข้อกำหนดการเขียน CGI ใน Server ของคุณ คือ

ถ้าใช้ UNIX คุณจะต้องใช้โปรแกรม vi เขียน หรือใช้ Notepad เขียนและ Convert ให้เป็นไฟล์ในระบบ UNIX จากนั้นจึง Upload ขึ้นไป และเปลี่ยนโหมดให้เป็น 755 จึงจะใช้งานได้

แต่ถ้าคุณใช้ NT คุณใช้ Notepad เขียนและ Upload ขึ้นไปได้เลย โดยใช้นามสกุลของไฟล์ตามที่ Server ของคุณกำหนดไว้เช่น .pl , plx หรืออื่น ๆ

โปรแกรม easy.plx (หรือ easy.cgi ตามที่ Server กำหนด)

```
#!/usr/bin/perl
print "Content-type:text/html\n\n";
print "<h1>CGI is easy for you</h1>";
```

จากนั้น ถ้าคุณใช้ระบบ UNIX ให้คุณเปลี่ยนโหมดของไฟล์นี้ ให้เป็น 755 โดยใช้คำสั่งนี้

```
Schmod 755 easy.plx
```

จากนั้นให้คุณทดลองใช้บราวเซอร์ จีไปที่ URL ของโฮมเพจของคุณ เช่น

```
http://domain.com/~account/easy.plx เป็นต้น
```

ฟอร์มและการ Submit

เมื่อคุณสร้างฟอร์มขึ้นมา 1 ฟอร์ม การ Submit ฟอร์มก็คือการส่งข้อมูลที่อยู่ภายในฟอร์มนั้น ไปประมวลผลโดยโปรแกรมบน Server ด้วยโปรแกรม CGI ซึ่งคุณจะต้องกำหนดชื่อโปรแกรม CGI ลงใน action และกำหนด method ให้เป็น POST เพราะเป็นการส่งข้อมูลไปให้กับโปรแกรม CGI ดังนี้

```
<form method=POST action="/cgi-bin/easy.plx">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการ Submit ข้อมูลที่อยู่ภายในฟอร์มจะ ถูกนำไปเก็บไว้ในตัวแปรที่ชื่อ CONTENT_LENGTH ซึ่งเป็นตัวแปรตัวหนึ่งบน Server ดังตัวอย่างของโปรแกรม CGI ต่อไปนี้

โปรแกรม easy1.plx

```
#!/usr/bin/perl
print "Content-type : text/html\n\n";
print "<body bgcolor='white'>\n";
print "<h4>read data from CONTENT_LENGTH</h4><p>";

read( STDIN , $buffer , $ENV{'CONTENT_LENGTH'} );
print "<h4>data is $buffer</h4><p>";
```

จากโปรแกรมข้างต้น บรรทัดที่ 5 เป็นการอ่านข้อมูลจากตัวแปร CONTENT_LENGTH มาเก็บไว้ในตัวแปร \$buffer โดยใช้คำสั่ง read โดยจะอ่านจาก Standard Input (STDIN) จากตัวแปรระบบ CONTENT_LENGTH เราจะเขียนคำสั่ง read ดังนี้

```
read( STDIN , ตัวแปรที่จะเก็บค่าเอาไว้ , $ENV{ ตัวแปรระบบ } );
```

ให้คุณเข้าใจว่า บรรทัดนี้จะอ่านข้อมูลที่ได้จากการ Submit มาเก็บไว้ในตัวแปรที่ต้องการ และบรรทัดที่ 6 เป็นการแสดงข้อมูลภายในตัวแปร CONTENT_LENGTH ฟอร์มต่อไปนี้เป็นทดสอบ easy1.plx

```
<form method=POST action="/cgi-bin/easy1.plx">
```

```
Enter text : <input type=text name="text">
```

Enter text :

```
<input type=submit><input type=reset>
```

</form>

จากฟอร์มในข้างต้น ให้คุณลองใส่ข้อความใด ๆ ลงในช่องและกดปุ่ม Submit ดูสิครับ.....

เมื่อคุณกดปุ่ม Submit ข้อความที่คุณพิมพ์ลงไปนั้น จะถูกนำไปเก็บไว้ในตัวแปร CONTENT_LENGTH ที่อยู่ใน Server โดยข้อความที่คุณพิมพ์ลงไปนั้น โปรแกรม CGI จะแสดงออกมาในรูปแบบ

ชื่อของอุปกรณ์ในฟอร์ม=ข้อความที่พิมพ์เข้าไป

จากในตัวอย่าง จะเห็นว่าชื่อของช่องรับข้อความ (name) ก็คือ "text" เพราะฉะนั้น เมื่อคุณกดปุ่ม Submit ข้อความภายในตัวแปร CONTENT_LENGTH ก็จะถูกเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
text=this+is+a+test
```

จะเห็นได้ว่า ข้อความที่อยู่ในตัวแปร CONTENT_LENGTH จะแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนที่เป็นชื่อและส่วนที่เป็นข้อมูล โดยส่วนที่เป็นเครื่องหมายบวก (+) ก็คือ การเว้นวรรคนั่นเอง ถ้ามีอุปกรณ์ภายในฟอร์มมากกว่า 1 ตัวเช่น มีช่องรับข้อความ 2 ช่อง ข้อมูลภายในตัวแปร CONTENT_LENGTH จะถูกแบ่งด้วยเครื่องหมาย & เช่น

```
<form method=POST action="/cgi-bin/easy2.plx">
```

```
Enter text1 : <input type=text name="text1">
```

```
Enter text1 : 
```

```
Enter text2 : <input type=text name="text2">
```

```
Enter text2 : 
```

```
<input type=submit><input type=reset>
```

```
  </form>
```

ให้คุณทำความเข้าใจ โดยการทดลองพิมพ์ข้อความลงในช่องรับข้อความและ Submit ดู หากคุณมีอุปกรณ์ในฟอร์มมาก มันก็จะเรียงต่อ ๆ กันไปด้วยเครื่องหมาย & และตรงไหนที่เป็นช่องว่าง มันก็จะแทนด้วยเครื่องหมาย + นั่นเอง

ถ้าคุณใส่ช่องแรกว่า Hello darling และใส่ช่องที่ 2 ว่า Good bye เมื่อ Submit จะได้ผลดังนี้

```
data is text1=Hello+darling&text2=good+bye
```

เห็นมั้ยว่า ช่องแรกคือ text1 จะเก็บ Hello+darling และช่องที่ 2 เก็บ good+bye โดยมี & ถัดกัน

การผ่านข้อมูลที่ได้จากฟอร์มลงในตัวแปร

เราสามารถเก็บข้อมูลที่ได้จากการ Submit ฟอร์ม มาเก็บไว้ในตัวแปรแบบ Associative Array เพื่อให้ง่ายต่อการใช้งาน เราจะใช้โค้ดโปรแกรมดังนี้

```
sub pasteform {
```

```
read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
```

```
@pairs = split(/&/,$buffer);
```

```
foreach $pair (@pairs){
```

```
($name,$value) = split(/=/,$pair);
```

```
$value =~ tr/+// ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/eg;
$value =~ s/~!/~!/g; $FORM{$name}=$value;

}

}

```

เราจะเขียนโค้ดโปรแกรมนี้ ไว้ที่ส่วนท้ายของโปรแกรม CGI เป็น procedure หนึ่ง ซึ่งจะต้องเรียกเพื่ออ่านข้อมูลจาก CONTENT_LENGTH มาเก็บไว้ในตัวแปร @FORM

โปรแกรม form.plx ต่อไปนี้ ใช้สำหรับแสดงผลข้อมูลภายในฟอร์ม

โปรแกรม form.plx

```

#!/usr/bin/perl
print "Content-type:text/html\n\n";
&pasteform;
print "name is $FORM{'name'}<p>";
print "email is $FORM{'email'}<p>";

sub pasteform {
read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
@pairs = split(/&/,$buffer);

foreach $pair (@pairs){

($name,$value) = split(/=/,$pair);
$value =~ tr/+//;
$value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/eg;
$value =~ s/~!/~!/g; $FORM{$name}=$value;

}

}

```

ดังตัวอย่างต่อไปนี้

```

<form method=POST action="/cgi-bin/form.plx">
Name : <input type=text name="name">
Name : |
Email : <input type=text name="email">

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Email :

<input type=submit><input type=reset>

Submit Query Reset </form>

เมื่อคุณคลิกปุ่ม Submit คุณก็จะได้ข้อความที่อยู่ในช่อง name และ email เก็บอยู่ในตัวแปร \$FORM {name} และ \$FORM{email} ตามลำดับ คุณสามารถนำเอาวิธีการนี้ไปใช้งานในโปรแกรม CGI ของคุณได้ โดยให้คุณก๊อปปี้ sub pasteform นี้ลงไปโปรแกรม CGI ของคุณ จากนั้น คุณก็เรียก &pasteform ก่อน ข้อมูลก็จะถูกนำมาเก็บไว้ในตัวแปร \$FORM{"ชื่อของอุปกรณ์"} โดยอัตโนมัติ ดังเช่น โปรแกรม form.plx นั้นเอง

สรุปง่าย ๆ คือ เราจะใช้ pasteform นี้ก็ต่อเมื่อเราต้องการอ่านค่าจาก Form มาใช้ เช่น เราต้องการเก็บค่าอีเมลล์จากผู้ใช้ เราก็สร้างฟอร์มมาอันหนึ่ง จากนั้นสร้าง CGI ขึ้นมารองรับ โดยภายใน CGI นั้นใส่โค้ดของ pasteform เอาไว้ท้ายสุด ดังนี้

```
.....เริ่มต้น โปรแกรม CGI .....
.....
.....
.....
sub pasteform {
read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
@pairs = split(/&/,$buffer);

foreach $pair (@pairs){

($name,$value) = split(/=/,$pair);
$value =~ tr/+ / /;
$value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/eg;
$value =~ s/~!/~!/g;

$FORM{$name}=$value;
}
}
```

รูปที่ 5-1 โปรแกรมซีจีไอในรูปแบบ sub pasteform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจ้า sub pasteform นี้จะทำหน้าที่เหมือนกับคำสั่ง ๆ หนึ่ง ซึ่งเราจะเรียกเมื่อเราเริ่มโปรแกรม CGI เรา จะเรียกใช้โดยเขียนเอาไว้ต้นไฟล์ ดังนี้

```
#!/usr/bin/perl
print "Content-type:text/html\n\n";
&pasteform; <--
```

เมื่อเราเรียกแล้ว ค่าที่เราต้องการก็จะถูกเก็บอยู่ในตัวแปรอะไรชื่อ %FORM โดยเราสามารถเรียกใช้ ได้ โดยใช้

```
$FORM{ ชื่ออุปกรณ์ };
```

เช่น ถ้าเราสร้างฟอร์มโดยมีเท็กบ็อกที่กำหนด name ว่า email เช่น

```
<input type=text name="email">
```

เมื่อเรา pasteform แล้ว CGI จะเอาค่าที่ Post มาจากเท็กบ็อกอันนี้เก็บไว้ในตัวแปร \$FORM{email} โดยอัตโนมัติ จากนั้นเราก็สามารถนำเอาค่าจากฟอร์ม email นี้ไปใช้ได้ ดังในตัวอย่างข้างต้นนั่นเอง...

5.2.11 การจัดการไฟล์

การจัดการเกี่ยวกับไฟล์ในภาษา Perl เราจะใช้ฟังก์ชันคล้าย ๆ กับภาษา C นั่นคือ เราจะต้องมีการเปิดไฟล์ก่อนเป็นอันดับแรก และจะต้องระบุด้วยว่าเราเปิดไฟล์เพื่ออะไร จากนั้นเมื่อใช้งานไฟล์เสร็จแล้ว จึงทำการปิดไฟล์

การเปิดและปิดไฟล์

การเปิดไฟล์ เราจะใช้ฟังก์ชัน open ดังนี้

```
open( ตัวแปรไฟล์ , "พารามิเตอร์ของชื่อไฟล์" );
```

ตัวแปรไฟล์เป็นตัวแปรที่ใช้อ้างถึงในการจัดการไฟล์ คุณต้องกำหนดตัวแปรนี้ขึ้นมาเอง (อะไรก็ได้) และการระบุวิธีการเปิดไฟล์ ในภาษา Perl จะระบุได้โดยใช้เครื่องหมายดังนี้

" " ไม่ระบุเครื่องหมายคือการเปิดไฟล์เพื่ออ่าน

">" หมายถึงการเขียนไฟล์ใหม่

">>" หมายถึงการเปิดไฟล์เพื่อเพิ่มข้อมูลลงไปinfile

ดังตัวอย่างต่อไปนี้ เป็นการเปิดไฟล์ hello.dat ทั้ง 3 แบบ

```
open( FILE , "hello.dat" );
```

เป็นการเปิดไฟล์ hello.dat แบบอ่านข้อมูลอย่างเดียว

```
open( FILE , ">hello.dat" );
```

เปิดไฟล์ hello.dat แบบเขียน ถ้ามีไฟล์เก่าอยู่แล้ว ข้อมูลเก่าก็จะหายไป

```
open( FILE , ">>hello.dat" );
```

เปิดไฟล์ hello.dat แบบเพิ่มข้อมูลใหม่เข้าไป ข้อมูลเก่าจะยังอยู่ครบหมดไม่สูญหาย

เมื่อคุณเปิดไฟล์แล้ว คุณก็ต้องปิดไฟล์ด้วยเมื่อใช้เสร็จ ถ้าไม่ปิดคุณทราบหรือไม่ว่าจะเกิดอะไรขึ้น มันเหมือนกับเปิดประตูแล้วไม่ยอมปิดนั้นล่ะครับ (อ้าว... คนละเรื่องเลย..) เพราะถ้าคุณไม่ปิดไฟล์ ข้อมูลก็จะไม่ถูกเซฟลงดิสก์ และสูญหายได้

การปิดไฟล์ใช้ฟังก์ชันง่าย ๆ คือฟังก์ชัน close ดังนี้

```
close( ตัวแปรไฟล์ );
```

ตัวอย่างเช่น

```
open( FILE , "hello.dat" );
```

```
.....ใช้งานไฟล์.....
```

```
close( FILE );
```

การเขียนข้อมูลลงไฟล์

เมื่อคุณต้องการเขียนข้อมูลลงไฟล์ เราจะใช้ฟังก์ชัน print ครับ โดยเราจะต้องระบุตัวแปรไฟล์ลงไปด้วย โดยใช้รูปแบบดังนี้ครับ

```
print ตัวแปรไฟล์ "ข้อมูลที่ต้องการเขียนลงไฟล์";
```

ตัวอย่างเช่น ถ้าคุณต้องการเขียนคำว่า "Perl is easy for you" ก็ต้องเขียนดังนี้

```
open( FILE , "hello.dat" );
```

```
print FILE "Perl is easy for you";
```

```
close( FILE );
```

จากข้างต้น ไฟล์ hello.dat จะถูกสร้างขึ้นใหม่ และเขียนข้อมูลลงไปในไฟล์ว่า "perl is easy for you" ถ้าคุณต้องการให้เขียนข้อมูลลงไฟล์ โดยข้อมูลนี้มาจากตัวแปรล่ะก็ คุณก็สามารถทำได้เช่นกัน ดังโปรแกรมต่อไปนี้

```
$name = "Richard";
```

```
$age = "30";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
open( FILE , ">data.dat" );
print FILE "name is $name , age is $age\n";
close( FILE );
```

จากโปรแกรมในข้างต้น กำหนดตัวแปร \$name และ \$age จากนั้นเขียนลงไฟล์ data.dat โดยจะเห็นได้ว่าเป็นการเขียนแบบเพิ่มข้อมูลลงไฟล์ (>) ข้อมูลจะเพิ่มเข้าไปในไฟล์ ดังนี้

```
name is Richard , age is 30 <-- ตามด้วยขึ้นบรรทัดใหม่ "\n"
```

การอ่านข้อมูลจากไฟล์

ในการอ่านข้อมูลจากไฟล์ จะง่ายกว่าการเขียนข้อมูลลงไฟล์ เพียง 3 บรรทัด ข้อมูลก็อ่านจากไฟล์มาเก็บลงในตัวแปรได้แล้ว ดังนี้

```
open( FILE , "data.dat" );
@buffer = <FILE>;
close( FILE);
```

เห็นมั้ยครับ ข้อมูลจากไฟล์ data.dat จะถูกอ่านมาเก็บไว้ในตัวแปรอะเรย์ชื่อ @buffer โดยเขียนโค้ดโปรแกรมเพียง 3 บรรทัดจริงๆ

คราวนี้เราจะมาทำความเข้าใจกับข้อมูลภายในไฟล์กันสักเล็กน้อย โดยสมมติว่าในไฟล์ data.dat มีข้อมูลต่อไปนี้เก็บอยู่

```
this is a test 1
this is a test 2
this is a test 3
this is a test 4
this is a test 5
```

เมื่อเราอ่านไฟล์โดยใช้โค้ด โปรแกรมดังที่ได้เสนอก่อนไปในข้างต้นนั้น ข้อมูลทั้ง 5 บรรทัดจะถูกอ่านและเก็บลงตัวแปรอะเรย์ @buffer โดยจะแยกเป็นทีละบรรทัด ๆ เลย นั่นคือ ช่องที่ 0 ของอะเรย์ buffer จะเก็บบรรทัด 1 ช่องที่ 1 ก็จะเก็บบรรทัด 2 ... ไปเรื่อย ๆ ดังนี้

```
$buffer[0] เก็บ this is a test 1
$buffer[1] เก็บ this is a test 2
$buffer[2] เก็บ this is a test 3
$buffer[3] เก็บ this is a test 4
$buffer[4] เก็บ this is a test 5
```

5.2.12 การส่งข้อมูลจากฟอร์มไปยังที่อยู่ E-mail ของผู้รับ

เราสามารถส่งข้อมูลจากแบบฟอร์มไปยังที่อยู่ e-mail ของผู้รับจากภายใน Tag <FORM> ได้โดยตรง โดยการกำหนดให้ ACTION เป็น "mailto: e-mail" และกำหนด option enctype ให้เป็น "test/plain" ดังรูปแบบนี้

```
<FORM ACTION = mailto:ngamnij@hotmail.com enctype="text/plain">
```

บราวเซอร์ส่วนมากจะสามารถเข้าใจความหมาย และจัดการส่งข้อมูลให้ได้อย่างถูกต้อง แต่ข้อเสียของการส่งด้วยวิธีนี้คือ เราไม่สามารถควบคุมรูปแบบของการส่งข้อมูลที่จะถูกส่งไปได้ และเราก็ไม่สามารถส่งข้อความโต้ตอบกลับไปยังผู้ส่งเมลนั้นได้ นอกจากนี้บราวเซอร์บางตัวอาจจะไม่สามารถเข้าใจรูปแบบของ Tag <FORM> แบบนี้ได้

อาจจะยกตัวอย่างส่วนของเมลฟอร์มในส่วนของ HTML โค้ด

```
<html>
<head><title>CGI Send Mail Program</title></head>
<body>
<h4>CGI Mail Gateway</h4>
<form action="SendMail.cgi" method="POST">
<p>Full Name<input type="text" size=25 maxlength=256 name="Fname"><br>
E-mail <input type="text" size=20 maxlength=256 name="Email"><br>
To <input type="text" size=25 maxlength=256 name="Mailto"><br>
Subject <input type="text" size=25 maxlength=256 name="Subject"></p>
<p><textarea name="MailBody" rows=5 cols=49></textarea></p>
<p><input type="submit" name="Submit" value="Send Mail Now">
<input type="reset" name="Clear" value="Clear"></p>
</form>
</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTML ที่แสดงนี้ สำหรับส่งเมล โดยมีรายละเอียดตัวแปร ที่ส่งค่าผ่านฟอร์ม ไปยัง CGI โปรแกรม ซึ่ง กำหนดชื่อโปรแกรมในฟอร์มแท็ก (Tag) คือ “SendMail.cgi” รายละเอียดตัวแปรคือ Fname สำหรับ ค่าชื่อ-นามสกุลของผู้ส่ง, Email สำหรับรายละเอียด, E-mail Address หรือที่อยู่ของผู้ส่งเมื่อใช้อ้างอิง ถึงผู้ส่ง, Mailto สำหรับรายละเอียดค่า E-mail Address หรือที่อยู่ของผู้รับปลายทาง, Subject คือส่งนที่ ระบุถึงชื่อเรื่องสำหรับเมล และ Mailbody ใช้สำหรับอ้างอิงถึงค่าของเนื้อความในเมล

ส่วนของโปรแกรม SendMail.cgi จะได้ดังนี้

```
#!/usr/bin/perl
```

```
&parse_form;
```

```
print “content-type: text/html \n\n”;
```

```
$F_name = $contents{‘Fname’};
```

```
$E_mail = $contents{‘Email’};
```

```
$Mail_to = $contents{‘Mailto’};
```

```
$Mail_subj = $contents{‘Subjext’};
```

```
$Mail_body = $contents{‘Mailbody’};
```

ส่วนแรกคือการถอดค่าที่ผ่านมากับฟอร์มในโปรแกรมย่อยที่ชื่อ parse_form จากนั้นสั่งพิมพ์ผล ลัพท์ที่ส่งกลับในรูปแบบของ HTML และนำค่าตัวแปรหรือคีย์ที่ผ่านค่ามา กำหนดลงตัวแปร ได้แก่ ตัวแปร \$F_name คือค่าของชื่อ-สกุล ของผู้ส่ง, \$E_mail คือค่าของ Email Address ของผู้ส่ง, \$Mail_to คือค่าของ Email Address ของปลายทาง, \$Mail_subj คือค่าของหัวข้อเรื่องเมล และ \$Mail_body คือค่าของเนื้อความเมล

```
if ($Mail_to = ~/([\w\-\+]+)@([\w\-\+\.]+)/){
```

```
    open (SENDMAIL, “[usr/bin/mail $Mail_to]”);
```

```
{else {
```

```
    print “Address is invalid.”;
```

```
    exit(0);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะทำการตรวจสอบ Email Address หรือที่อยู่ของผู้รับปลายทาง ว่ามีการใส่ค่า หรือกำหนดค่ามาถูกต้องหรือไม่ หากกำหนดไม่ถูกต้องจะทำการพิมพ์ "Address is invalid" และจบการทำงาน แต่ถ้ากำหนดถูกต้องจะทำการส่งเมลต่อไป โดยเรียกใช้คำสั่ง mail ใน PATH ที่อยู่หรือที่เก็บคำสั่งบนเซิร์ฟเวอร์คือ "/usr/bin" ซึ่งการเรียกโปรแกรมในภาษา Perl สามารถทำได้หลายวิธีเช่น การใช้คำสั่ง Exec หรือสั่งรัน ในลักษณะเหมือน Shell Script แต่ในโปรแกรมนี้ใช้คำสั่ง Open เหมือนการใช้งานไฟล์ (File Processing) โดยกำหนดชื่อที่ต้องการอ้างอิงสำหรับ โปรแกรมที่เรียกใช้งานเป็น "SENDMAIL"

```
print SENDMAIL <<Mail_Header;
```

```
From; $E_mail <$F_name>
```

```
To: $Mail_to
```

```
Reply_To: $E_mail
```

```
Subject: $Mail_subj
```

```
X-Mailer: CGI Mail Demo System
```

```
Mail_Header
```

ทำการพิมพ์ส่วนหัวของไฟล์ (Mail Header) โดยสั่งให้พิมพ์รายละเอียดที่ตามมาจกบรรทัด print SENDMAIL<<Mail_Heacer จนถึงบรรทัดสุดท้ายที่กำหนดไว้คือ Mail_Header จากนั้น โปรแกรมจะทำการติดต่อกำสั่ง mail บน UNIX และส่งรายละเอียดของ Mail Header

```
print SENDMAIL "\n",$Mail_body,"\n";
```

```
close (SENDMAIL);
```

ต่อไปคือสั่งพิมพ์เนื้อความหรือใจความของเมล และปิดการใช้งานคำสั่ง mail

```
print <<End_Thanks;
```

```
<HTML>
```

```
<BODY>
```

```
<H1> Thank you </H1>
```

```
<P>for using the CGI Mail Demo. </P>
```

```
</BODY>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</HTML>
```

```
End_Thanks
```

ต่อไปคือนำส่วนของโปรแกรมย่อยที่ชื่อ `parse_form` โปรแกรมย่อยหรือฟังก์ชัน `parse_form ()` จะทำหน้าที่ถอดรหัส หรือนำข้อมูล ที่ผ่านค่ามาจาก บราวเซอร์โปรแกรม โดยนำค่าต่างๆ นั้นได้กับคีย์หรือตัวแปรที่ระบุไว้ใน HTML

```
sub parse_form {
```

```
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
```

```
    if (length($buffer)<5) {
```

```
        $buffer = $ENV{QUERY_STRING};
```

```
    }
```

ตัวแปร `$buffer` จะรับค่าจากตัวแปรประเภท Environment Variable ซึ่งค่าที่มีการโพสหรือ ส่งผ่านมา จะถูกเก็บในตัวแปร Environment ชื่อ “QUERY_STRING” โปรแกรมจะนำค่าที่มีการโพสหรือส่งผ่านบราวเซอร์โปรแกรม นั้นมาใช้งานโดยใช้คำสั่ง `$ENV{QUERY_STRING}` ซึ่งรูปแบบที่ได้จะเป็น “KEY”=“VALUE”&“KEY”=“VALUE”&.....=“VALUE”

```
@pairs = split(/&/, $buffer);
```

```
foreach $pair (@pairs) {
```

```
    ($name, $value) = split(/=/, $pair);
```

```
    $value =~ tr/+//;
```

```
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
```

```
    $contents{$name} = $value;
```

```
}
```

```
}
```

จากนั้นโปรแกรมจะมาจัดแยกข้อมูลของคีย์ หรือ ตัวแปร โดยแยกตัวแปรต่างๆ และค่าที่ส่งมาด้วยนั้นให้อยู่ในรูปแบบ `$contents{'KEY'} = “VALUE”` และเมื่อใช้งานก็จะใช้คำสั่งรูปแบบดังนี้ `$contents{'[KEY NAME]'`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 พื้นฐานการใช้งาน MySQL

- การสร้างตารางและการเพิ่มข้อมูล (Creating and Insertion)

ถ้าตารางหรือ Table ของเรามีรายละเอียดดังนี้

ฐานข้อมูลชื่อ BOOK, ตารางชื่อ PhoneBook และมีรายละเอียดฟิลด์ ดังนี้

ID = หมายเลขหนังสือ, FNAME = ชื่อ, LNAME = นามสกุล, และ Phone = เบอร์โทรศัพท์ และแต่ละฟิลด์มีรายละเอียดต่างๆ

Database = BOOK

Table = PhoneBook

จะได้การเขียน SQL สำหรับการสร้างตารางดังต่อไปนี้

```
CREATE TABLE PhoneBook
```

```
(ID char(8),
```

```
FName char(30),
```

```
LName char(30),
```

```
Phone char(30))
```

ถ้าต้องการเพิ่มข้อมูลจะได้ว่า

```
INSERT INTO PhoneBook
```

```
(ID,FName,LName,Phone)
```

```
VALUES('38626072','Chalongchai','Jongpasertporn','222-1919 01-2223539')
```

- การสอบถามข้อมูล (Query) หรืออ่านข้อมูล (Accessing data)

ถ้าเราต้องการรู้ข้อมูลของตารางสามารถใส่คำสั่งได้ดังนี้

```
SELECT ID,FName,Phone
```

```
FROM PhoneBook
```

```
WHERE ID = '38626061'
```

- การปรับปรุงข้อมูล (Update) และการลบข้อมูล (Delete)

ถ้าเราต้องการปรับปรุงข้อมูล ให้ใส่ข้อมูลดังนี้

```
UPDATE PhoneBook
```

```
SET Phone = '5116844'
```

```
LName = 'Jaidee'
```

```
WHERE ID = '38626073'
```

5.4 Java กับ Graphics

ถ้าเราจะพูดถึงถึงเรื่องของกราฟิก ถ้าเทียบกับการศึกษาการเขียน โปรแกรมภาษาอื่น ๆ เราก็คงจะมาถึงการเขียน โปรแกรมในระดับที่สูงกันพอสมควรแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.1 การ compile และรันโปรแกรม

การ compile นั้น ให้ท่าน download ตัว JDK (Java Development Kit) มาติดตั้งที่เครื่องก่อน จากนั้นเราก็จะสามารถ compile และรัน โปรแกรม Java ได้

เมื่อท่านเขียน โค้ด โปรแกรม ให้ท่านใช้ Edit หรือ Notepad จากนั้น save ในชื่อไฟล์ ตามด้วยนามสกุล .java

เมื่อท่านต้องการ compile ให้ใช้

```
c:\> javac filename.java
```

เราก็จะได้ .class มา จากนั้นใช้คำสั่ง java หรือ jview ในการรัน

```
c:\> java filename
```

ไม่ต้องใส่ .class ลงไปนะครับ หรือ

```
c:\> jview filename
```

หากกำหนดชื่อไฟล์ให้เป็น Test.java เมื่อ compile จะได้ Test.class และเมื่อรัน เราจะพิมพ์แค่

```
c:\> jview Test
```

5.4.2 การวาดภาพในวินโดวส์

คงจะแปลกไม่น้อยถ้าจะมีโปรแกรมที่ใช้ในการวาดภาพเป็นภาษาจาวา ในความเป็นจริงนั้น คงจะเห็นได้ยาก แต่ถ้าจะถามว่าในการเขียนแอปพลิเคชันหนึ่ง ๆ เนี่ย ในไดอะล็อกหนึ่ง นอกจากการสร้างปุ่มกด , การสร้างคอมโบบ็อก หรือการสร้างลาเบลแล้ว เราสามารถสร้างเส้นตรง , วาดวงกลมขึ้นมาในไดอะล็อกโปรแกรมของเราได้หรือไม่?? คำตอบก็คือได้ เราสามารถวาดภาพต่าง ๆ ลงในวินโดวส์ได้ พร้อมกับการสร้างปุ่มกด หรือสร้างอุปกรณ์ต่าง ๆ

การวาดภาพในภาษาจาวา เราจะใช้คลาส Graphics ซึ่งเป็นคลาสที่อยู่ในฟังก์ชัน paint เราคงจะจำกันได้ ในการเขียนโปรแกรมจาวาที่ผ่าน ๆ มา จะมีฟังก์ชันที่ชื่อ paint() อยู่ภายในคลาสของโปรแกรมด้วย ดังโค้ดโปรแกรมต่อไปนี้

```
public void paint(Graphics g){
```

```
    resize(200,200);
```

}

จากฟังก์ชัน `paint()` ในข้างต้น อีโอปเจ็ทต์ `g` ของคลาส `Graphics` ตัวนี้ล่ะครับ เราจะใช้ในการวาดสิ่งต่าง ๆ ลงในวินโดวส์ ซึ่งเราจะมาทดลองวาดกันไปที่ละอย่าง ๆ ดังต่อไปนี้

5.4.3 การสร้างเส้นตรง

การสร้างเส้นตรงนี้ง่ายมาก ๆ กฎในการสร้างเส้นตรงนั้นมีอยู่ 2 อย่างครับ คือ เราจะต้องกำหนดจุดเริ่มต้น และจุดสิ้นสุดของเส้น เราจะใช้ฟังก์ชัน `drawLine()` ในการวาด ดังต่อไปนี้

โปรแกรม `G.java`

```
import java.awt.*;

public class G extends Frame{

public G(){

setTitle("Test Window");

}

public void paint(Graphics g){

resize(400,400);

g.drawLine(50,50,200,200);

}

public static void main(String args[]){

G g=new G();

g.show();

}
```



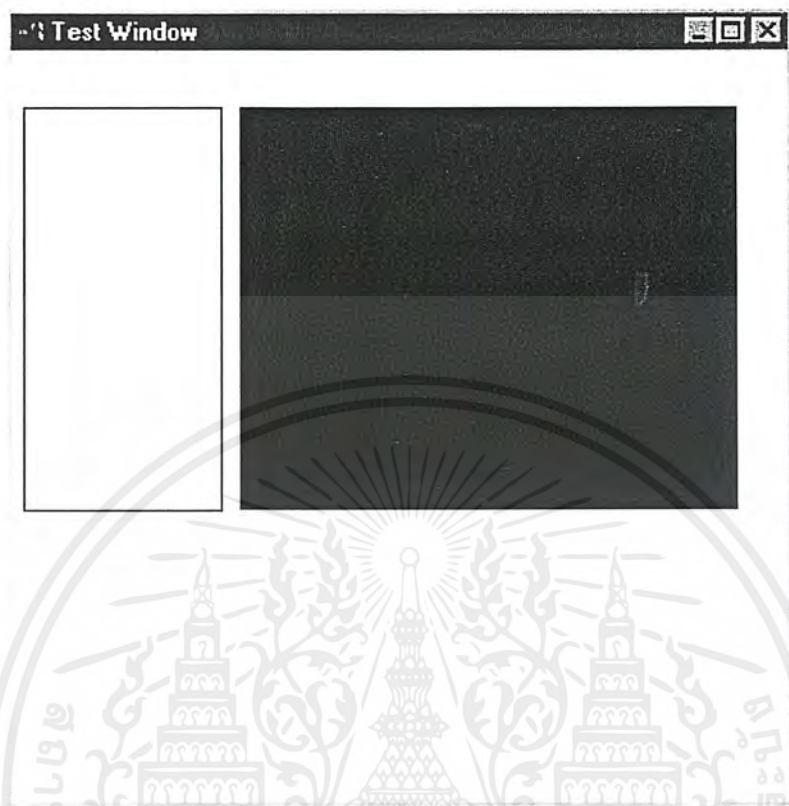
รูปที่ 5-2 รูปแสดงผลการรันโปรแกรม

5.4.4 การสร้างสี่เหลี่ยม

อันดับต่อไปก็คือการสร้างรูปสี่เหลี่ยม เราจะแบ่งรูปสี่เหลี่ยมออกเป็น 3 แบบคือ

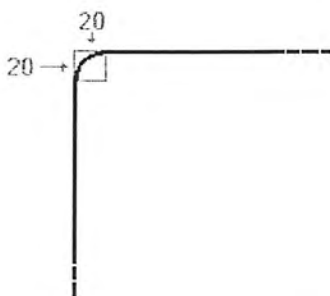
1. สี่เหลี่ยมมุมฉากธรรมดา ๆ เราจะใช้ฟังก์ชัน `drawRect()` สำหรับการสร้างสี่เหลี่ยมโปรง และ `fillRect()` สำหรับสร้างสี่เหลี่ยมแบบทึบ ดังโค้ดโปรแกรมต่อไปนี้ เป็นการสร้างรูปสี่เหลี่ยมมุมฉากแบบโปรง และแบบทึบ ดังโค้ดโปรแกรมต่อไปนี้

```
public void paint(Graphics g){
    resize(400,400);
    g.drawRect(10,50,100,200);
    g.fillRect(120,50,250,200);
}
```



รูปที่ 5-3 รูปแสดงผลการทำงานของโค้ดโปรแกรมในข้างต้น

2. สีเหลี่ยมมุมมน ก็เช่นเดียวกับการสร้างสี่เหลี่ยม แต่เพียงเปลี่ยนชื่อฟังก์ชันเท่านั้น คือฟังก์ชัน `drawRoundRect()` ใช้สำหรับสร้างสี่เหลี่ยมมุมมนแบบโปร่ง และ `fillRoundRect()` สำหรับสร้างสี่เหลี่ยมมุมมนแบบทึบ แต่การใส่พารามิเตอร์ให้กับ 2 ฟังก์ชันนี้ เราจะต้องกำหนดเพิ่มลงไปอีก 2 ตัวคือ ความกว้างและความสูงของมุมมน โดยเปรียบเทียบกับรูปสี่เหลี่ยมเล็ก ๆ

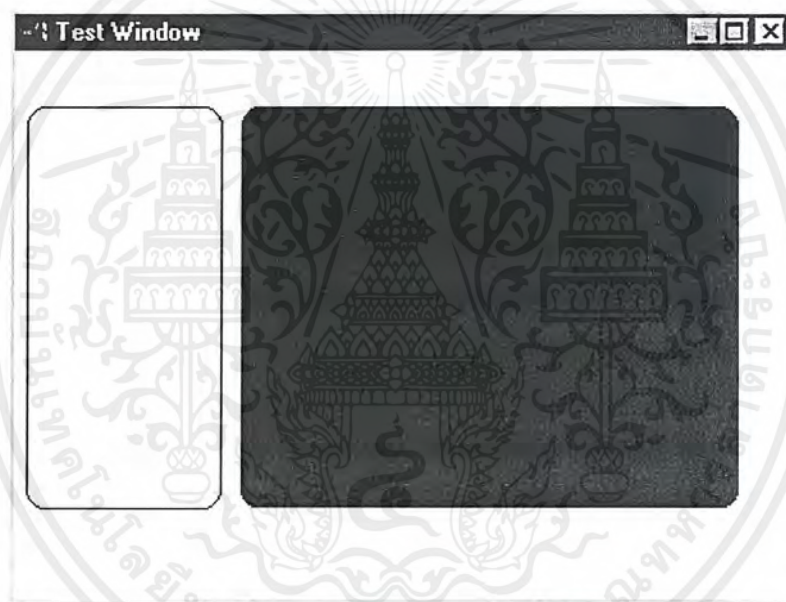


รูปที่ 5-4 ภาพแสดงความกว้างและความสูงของมุมมน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดโปรแกรมต่อไปนี้ เป็นการสร้างสี่เหลี่ยมมุมมน ให้สังเกตดูว่า ค่าพารามิเตอร์ 2 ตัวหลังนั้น ถ้ายิ่งมาก สี่เหลี่ยมก็จะยิ่งมนมากขึ้น ดังรูปที่ 4 เป็นผลการรัน โปรแกรม

```
public void paint(Graphics g){
    resize(400,300);
    g.drawRoundRect(10,50,100,200,20,20);
    g.fillRoundRect(120,50,250,200,20,20);
}
```



รูปที่ 5-5 แสดงผลจากการรันโปรแกรม

5.4.5 สี่เหลี่ยมมุมฉาก 3D

รูปสี่เหลี่ยมแบบ 3 มิติก็เช่นเดียวกัน เราจะใช้ฟังก์ชัน `draw3DRect()` และ `fill3DRect()` ในการสร้างรูป เช่นเดียวกับการใช้ฟังก์ชันสร้างสี่เหลี่ยมธรรมดา แต่เราจะต้องกำหนดค่า `true` หรือ `false` ลงไปในฟังก์ชันด้วย เพราะมันจะเป็นตัวกำหนดรูปแบบของความเป็น 3 มิติ คือ ถ้าเป็น `true` จะเป็นสี่เหลี่ยมแบบนูนขึ้น แต่ถ้าเป็น `false` ก็จะเป็นสี่เหลี่ยม 3 มิติแบบยุบลง ดัง โค้ด โปรแกรมต่อไปนี้

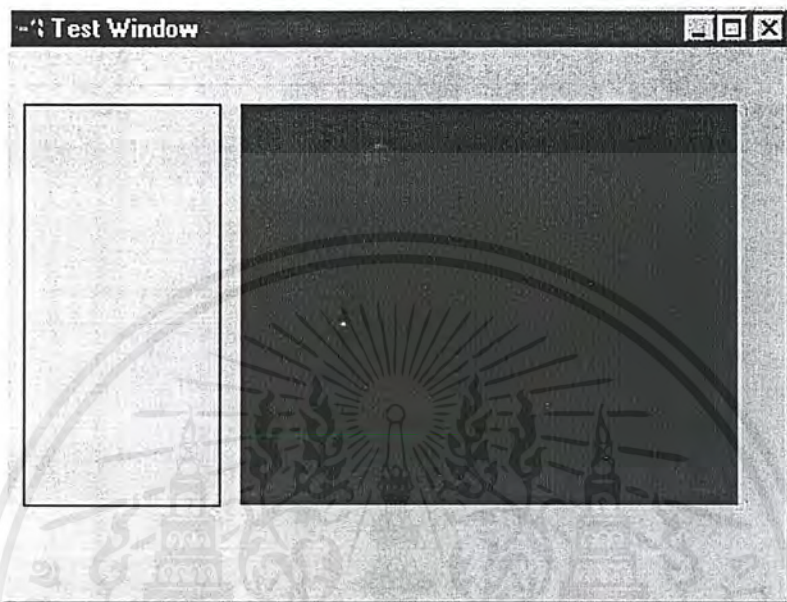
```
public void paint(Graphics g){
    resize(400,300);
    g.draw3DRect(10,50,100,200,true);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

g.setColor(Color.red);
g.fill3DRect(120,50,250,200,false);
}

```



รูปที่ 5-6 รูปแสดงผลการรันสร้างรูปสี่เหลี่ยมแบบ 3D

5.4.6 การสร้างวงกลมและส่วนโค้ง

การสร้างวงกลมและวงรี เราจะใช้ฟังก์ชันเดียวกัน แต่ต่างกันที่การกำหนดความกว้างและความยาว หากคุณเคยเขียน โปรแกรมสร้างวงกลมมาก่อน คุณจะรู้ว่าคุณต้องกำหนด พารามิเตอร์ให้กับมันเพียง 3 ตัวคือ พิกัด x และ y กับค่ารัศมี แต่ฟังก์ชันที่ใช้วาดวงกลมในภาษาจาวา เราจะต้องกำหนดพารามิเตอร์ 4 ตัว เหมือนกับการสร้างรูปสี่เหลี่ยม การที่มันจะเป็นวงกลมหรือวงรีได้นั้นขึ้นอยู่กับความเหมาะสมของความกว้างและความยาวของสี่เหลี่ยมนั่นเอง

1. การสร้างวงกลม

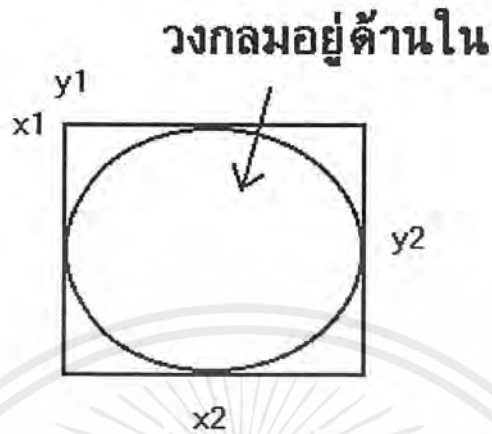
ฟังก์ชัน `drawOval()` จะใช้สำหรับสร้างวงกลมหรือวงรีแบบโปร่ง และ `fillOval()` ใช้สำหรับสร้างวงกลมหรือวงรีแบบทึบสี รูปแบบการใช้งานฟังก์ชันมีดังนี้

```
drawOval( x1 , y1 , x2 , y2 )
```

```
fillOval( x1 , y1 , x2 , y2 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

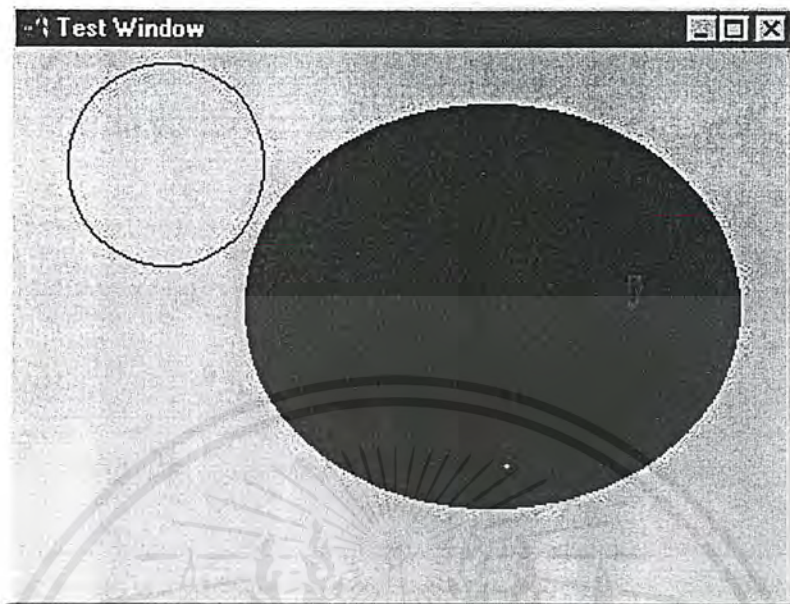
โดยที่ $x1$ และ $y1$ เป็นจุดเริ่มต้นของสี่เหลี่ยมซึ่งเป็นรูปที่ใช้สำหรับเทียบวงกลม และ $x2$ กับ $y2$ เป็นความกว้างและความสูงของมัน ให้เราทำความเข้าใจหลักการสร้างวงกลมในรูปที่ 6



รูปที่ 5-7 ผลของการรันโปรแกรมสร้างวงกลม

การสร้างวงกลมมีหลักอยู่ง่าย ๆ คือ เราจะต้องให้ $x2 - x1$ มีค่าเท่ากับ $y2 - y1$ นั่นคือ ความกว้างและความยาวของวงกลมจะเท่ากัน ส่วนหลักการสร้างวงรีมีอยู่ง่าย ๆ คือ ถ้า $x2 - x1$ ไม่เท่ากับ $y2 - y1$ นั่นล่ะครับ จะเป็นวงรีแล้ว เราลองมาดูโค้ดโปรแกรมต่อไปนี้ เป็นตัวอย่างการสร้างวงกลมและวงรี

```
public void paint(Graphics g){
    resize(400,300);
    g.drawOval(30,30,100,100);
    g.fillOval(120,50,250,200);
}
```



รูปที่ 5-8 ผลการรันโปรแกรมแสดง

2. การวาดเส้นโค้ง

การสร้างเส้นโค้งในภาษาจาวานั้น จะใช้หลักคล้าย ๆ กับภาษาชั้นสูง คือ เราจะต้องนำเอาเรื่องของมุมและวงกลม 1 หน่วยมาใช้ให้เกิดประโยชน์ ฟังก์ชันที่ใช้ในการสร้างเส้นโค้งคือ drawArc() และ fillArc() โดยเราจะต้องกำหนดพารามิเตอร์ 6 ตัวด้วยกันคือ

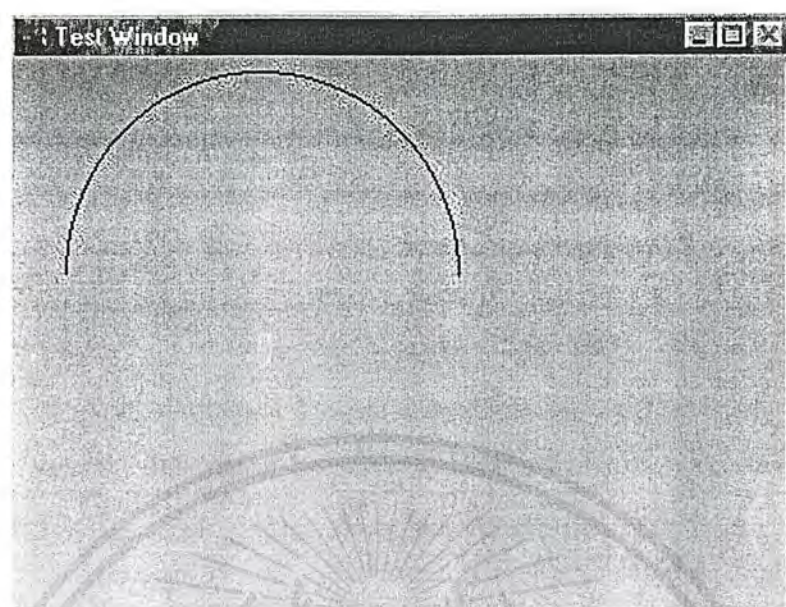
drawArc(x1 , y1 , x2 , y2 , องศาเริ่มต้น , องศาสิ้นสุด)

ตัวแปร x1 , y1 กับ x2 , y2 นั้นมีหน้าที่เช่นเดียวกับการสร้างวงกลมในหัวข้อที่แล้ว แต่องศาเริ่มต้นและสิ้นสุดนี้จะเป็นตัวที่ระบุความยาวของเส้นโค้ง หากเราต้องการสร้างส่วนโค้งครึ่งวงกลม เราจะเขียนโค้ดโปรแกรมดังนี้

```
public void paint(Graphics g){
    resize(400,300);
    g.drawArc(30,30,200,200,0,180);
}
```

จากโค้ดโปรแกรมในข้างต้น เป็นการสร้างวงกลมโดยกำหนดสีเหลี่ยมที่จุด 30,30 และยาว 200 จากนั้นวาดวงกลมตั้งแต่จุด 0 ถึง 180 ดังรูปที่ 8 แสดงผลการรันโปรแกรมนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-9 จากการ์ตูนโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Gundavaram Shishir, CGI programming on the World Wide Web, O'Reilly & Associates, c1996
2. Eckel George, Building a Linux Internet server, Chris Hare Imprint Indianapolis, c1995
3. Randy Jay, MySQL and MSQL, O'Reilly, c1999
4. ฉลองชัย จงประเสริฐพร, CGI Web Programming, SE-EDUCATION PUBLIC COMPANY LIMITED, c1995
5. วศิน เพิ่มทรัพย์, Linux, บริษัท โปรวิชั่น จำกัด, c1994
6. งามนิจ อาจอินทร์, Web Programming, บริษัท ดวงกลมสมัย จำกัด, c1997



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้