

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การรับส่งข้อมูลผ่านทางอินเทอร์เน็ต
INTERNET INFORMATION TRANSFER



โดย

นาย เกรียง ไกร พงษ์ศิริ

นาย นฤพนธ์ วัชรอุจิวงศ์

อาจารย์ที่ปรึกษา

อ.สุรเดช ศรีไตรลักษณ์

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน..... 34061
วัน, เดือน, ปี..... 1 ต.ค. 2542

ขอสงวนลิขสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หรือเพื่อวัตถุประสงค์อื่นใดที่ขัดแย้งกับวัตถุประสงค์ของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตรเรื่อง

การรับส่งข้อมูลผ่านทางอินเทอร์เน็ต

Internet Information Transfer

จัดทำโดย

นาย เกรียงไกร พงษ์ศิริ เลขประจำตัว 39013149

นาย นฤพนธ์ วรรตรุจิวงศ์ เลขประจำตัว 39013160

อาจารย์ที่ปรึกษา

อ.สุรเดช ตรีไตรลักษณ์

รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ อาจารย์ที่ปรึกษา

(อ.สุรเดช ตรีไตรลักษณ์)

วันที่/...../.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูลผ่านทางอินเทอร์เน็ต

นาย เกรียงไกร พงษ์ศิริ รหัส 39013149

นาย นฤพนธ์ วรรค์จิววงศ์ รหัส 39013160

อ. สุรเดช ตรีไตรลักษณะ (อาจารย์ที่ปรึกษา)

ภาคการศึกษาที่ 2 ปีการศึกษา 2541

บทคัดย่อ

รายงานฉบับนี้เป็นการศึกษาเกี่ยวกับสื่อการเรียนการสอนผ่านระบบอินเทอร์เน็ต ประกอบไปด้วยสองส่วน ส่วนแรกคือส่วนที่นำเสนอข้อมูล ในขณะที่ส่วนที่สองประกอบไปด้วยส่วนที่รับและประเมินผลข้อมูล โดยรับข้อมูลจากผู้เข้ามาชมเว็บไซต์ โดยที่จะแตกต่างจากเว็บไซต์อื่น โดยที่เว็บไซต์ที่สร้างจากโปรแกรมนี้สามารถที่จะตอบสนองได้ทันทีระหว่างผู้เข้ามาชมเว็บไซต์ และ เว็บไซต์ที่เป็นฐานข้อมูล ดังนั้นจึงสามารถนำมาประยุกต์ใช้งาน เช่น การสอนทางไกล และการสอบทางไกล

Internet Information Transfer

Mr.Kriangkrai Phongsiri

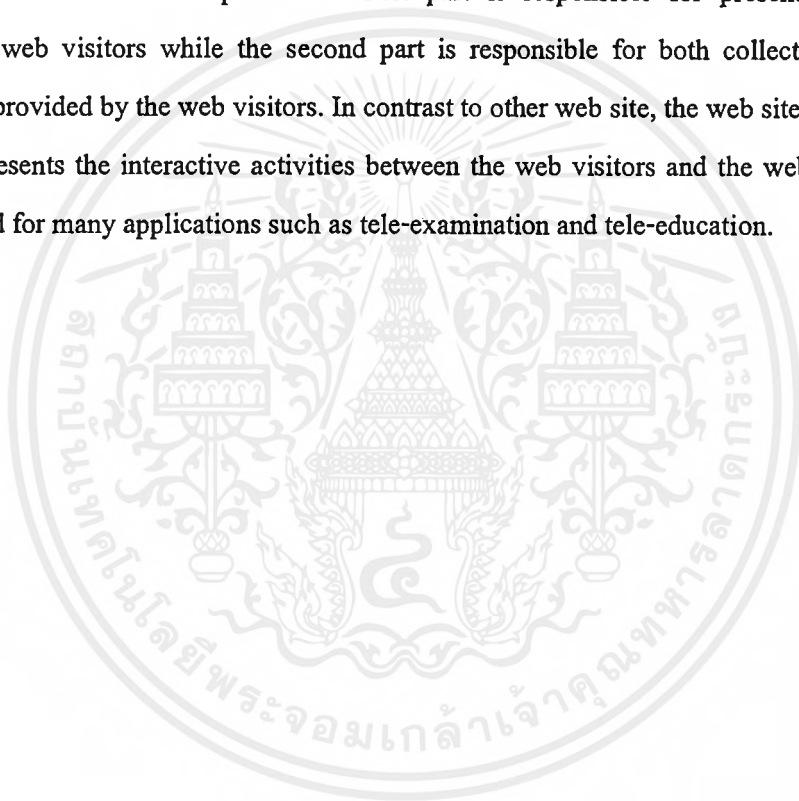
Mr.Narupon Waratrujiwong

Mr.Suradej Tretriluxana (Advisor)

2nd Semester , education Year 1998

Abstract

This project presents the study of the possibility in the education through the internet mean. The project consists of two parts. The first part is responsible for presenting the information to the web visitors while the second part is responsible for both collecting and evaluating the data provided by the web visitors. In contrast to other web site, the web site created from our project presents the interactive activities between the web visitors and the web server and thus can be used for many applications such as tele-examination and tele-education.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโครงการชิ้นนี้สำเร็จลุล่วงไปด้วยดีต้องขอขอบพระคุณ อ. สรุเดช ตรีไตรลักษณ์ อาจารย์ที่ปรึกษา ที่ให้คำแนะนำในการทำโครงการ ตลอดจนการแก้ปัญหาต่างๆที่เกิดขึ้น และขอขอบคุณเพื่อนๆที่เป็นกำลังใจและคำแนะนำต่างๆในการทำโครงการนี้ และขอบคุณเจ้าหน้าที่ห้องคอมพิวเตอร์ภาคอิเล็กทรอนิกส์ที่ให้ยืมคอมพิวเตอร์ในการทดลอง

สุดท้ายขอขอบคุณบิดา มารดา ที่ได้ให้การสนับสนุนการศึกษามาด้วยดีโดยตลอดมา

เสีโส

(นาย เกรียงไกร พงษ์ศิริ)

นศ.

(นาย นฤพนธ์ วรรตรุจิวงศ์)

ผู้จัดทำ

สารบัญ

เรื่อง	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
บทที่ 1 บทนำ	1
บทที่ 2 อินเทอร์เน็ต	2
2.1 ชื่อเครื่องในอินเทอร์เน็ต	3
2.2 การเชื่อมต่อเข้าอินเทอร์เน็ต	4
2.2.1 การต่อแบบเทอร์มินัล	4
2.2.2 การเชื่อมต่อด้วยโปรโตคอล	5
2.2.3 การเชื่อมต่อเข้าเครือข่ายเข้าอินเทอร์เน็ต	7
2.3 World Wide Web	7
2.4 ระบบเครือข่าย	9
2.5 การเชื่อมต่อระหว่าง UNIX กับ Microsoft Windows	9
2.6 Home Page	11
บทที่ 3 ระบบยูนิกซ์	12
3.1 ส่วนประกอบของระบบยูนิกซ์	12
3.2 คุณสมบัติของระบบยูนิกซ์	15
3.3 โครงสร้างของระบบยูนิกซ์	16
3.3.1 Hard Ware	16
3.3.2 UNIX Kernel	16
3.3.3 Shell	16
3.3.4 User Program	16
3.4 โครงสร้างไคเร็คทอรีใน UNIX	16
บทที่ 4 HTML	17
4.1 โครงสร้างพื้นฐานของ HTML	17
4.2 URLs	18

4.3 การสร้างแบบสอบถาม	19
บทที่ 5 Perl และ ASP	21
5.1 ตัวแปรภาษาของ Perl	21
5.2 มองหา Perl สำหรับระบบ	22
5.3 ติดตั้ง Perl สำหรับ UNIX	22
5.4 การสร้างและรันโปรแกรม	23
5.5 หลักการทำงานASP	28
5.6 ขั้นตอนการสร้าง ASP	28
5.7 การแทรกไฟล์	31
5.8 สาเหตุที่ทำให้เกิดปัญหา	32
5.9 การตรวจสอบ Browser ว่าสามารถใช้ ASP ได้หรือไม่	33
5.10 การเรียกใช้และตรวจสอบ	33
5.11 ไฟล์ Global.asa	36
5.12 การสร้างไฟล์ Global.asa	36
5.13 การเรียกใช้ตัวแปรและ Application ที่กำหนดไว้ใน Global.asa	38
บทที่ 6 CGI PROGRAM ON THE WORLD WIDE WEB	40
บทที่ 7 การเลือกใช้ Server และภาษาที่จะใช้เขียน CGI	53
บทที่ 8 ผลการทดลอง	56
บทที่ 9 สรุปและวิจารณ์	60
บรรณานุกรม	

บทที่ 1

บทนำ

อินเทอร์เน็ตนับว่าเป็นการพัฒนาทางการสื่อสารที่สามารถติดต่อระยะทางไกลๆ ได้และมีความไวในการติดต่อสื่อสารได้รวดเร็วพอสมควร และมีค่าใช้จ่ายที่ถูกลง สำหรับนักเรียนนักศึกษา เพราะสามารถติดต่อได้ฟรีโดยต่อเข้าทางสถาบันที่เราศึกษาอยู่

ในระบบอินเทอร์เน็ตนั้นมีความรู้ที่เราสามารถที่จะค้นหาได้มากมาย แต่ในการทดสอบเพื่อวัดความรู้ว่าได้อะไรไปบ้างจากการที่เข้ามาศึกษาในเว็บไซต์ของระบบอินเทอร์เน็ตนั้นๆ มีน้อยมาก เพราะ Web page ส่วนใหญ่จะเน้นในเรื่องการโฆษณาขายสินค้ามากกว่าการที่จะให้ความรู้แก่ผู้ที่เข้ามาเยี่ยมชม Web page นั้น ใน Project นี้จึงได้การแบบทดสอบความรู้ขึ้น เมื่อเราเข้ามาเยี่ยมชมใน Web page แล้วเราจะได้อะไรกลับไปบ้าง จึงได้สร้างแบบทดสอบขึ้น และยังสามารถประยุกต์ใช้งานเป็นการสอบทางไกลได้อีกด้วย โดยที่ผู้สอบไม่ต้องเดินทางมาสอบถึงสถานที่ เพียงแค่เปิดระบบ Internet ก็จะสามารถสอบได้เลย นอกจากนี้ยังสามารถที่จะสามารถบอกคะแนนของผู้ที่สอบได้เลยไม่ต้องรอให้เสียเวลา เพื่อรอการตรวจสอบผลสอบ ดังนั้นในอนาคต Internet จะต้องมีความจำเป็นต่อชีวิตประจำวันมากขึ้น ดังนั้นเราจะทำให้การศึกษาผ่านระบบอินเทอร์เน็ตนั้นมีความน่าสนใจมากยิ่งขึ้น และสามารถทำการเรียนรู้วิชาต่างๆ โดยผ่านอินเทอร์เน็ต และทำการทดสอบความรู้ได้ในขณะนั้น หลังจากทำการทดสอบเสร็จสิ้น ซึ่งมีความไวในการตอบโต้ได้ในขณะนั้น เราจึงจำเป็นต้องทำการศึกษาในเรื่อง Common Gate way Interface ซึ่งเป็นระบบติดต่อแบบสองทาง โดยที่ผู้รับและผู้ส่งสามารถติดต่อกันได้ในปัจจุบันทันด่วน

โดยวิทยานิพนธ์เล่มนี้ ได้ทำการรวบรวมเนื้อหาของโปรแกรมและระบบที่ใช้ในการสร้างโฮมเพจและระบบติดต่อแบบสองทาง ซึ่งเนื้อหาในรายงานเล่มนี้ประกอบไปด้วยเนื้อหาต่างๆ โดยที่จะแยกออกเป็นบทๆ ดังนี้

บทที่ 2 อินเทอร์เน็ต

บทที่ 3 ยูนิคซ์

บทที่ 4 HTML

บทที่ 5 Perl และ ASP

บทที่ 6 CGI Program on the World Wide Web

บทที่ 7 การเลือกใช้ Server และภาษาที่จะใช้เขียน CGI

บทที่ 8 ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

อินเทอร์เน็ต

ในปัจจุบันนี้ Internet ได้เข้ามามีความสำคัญมากในระบบการสื่อสารโดยมีวิวัฒนาการจากปี ค.ศ 1970 เป็นต้นมา โครงข่ายระบบคอมพิวเตอร์ (Computer network) เริ่มเติบโตขึ้นมาผู้ผลิตคอมพิวเตอร์หลายรายได้เริ่มหันมาผลิตมินิคอมพิวเตอร์ (Minicomputer) ที่มีสมรรถนะภาพเพียงพอสำหรับการใช้งานทางด้านการค้าขาย เพื่อรองรับผู้ใช้ที่มีมากขึ้นเนื่องจากคอมพิวเตอร์ขนาดนี้ราคาไม่แพงมากนัก ทำให้แผนกต่างๆในองค์กรขนาดใหญ่ สามารถหาซื้อมาใช้งานได้และเริ่มเชื่อมต่อคอมพิวเตอร์เหล่านี้เข้าด้วยกัน เพื่อให้มีการส่งข้อมูลระหว่างคอมพิวเตอร์เข้ากันได้อย่างรวดเร็ว จึงเริ่มมีการติดตั้งโครงข่ายเชื่อมโยงกันในแผนกต่างๆ ซึ่งโครงข่ายนี้มีชื่อเรียกว่า (Local Area Network : LAN) ต่อมา LAN ก็เริ่มมีผู้ใช้มากขึ้นเนื่องจากติดตั้งง่าย ซึ่งทำให้แต่ละแผนกในองค์กรสามารถหาซื้อมาติดตั้งได้ และสามารถดูแลระบบกันเองได้ โดยไม่ได้วางแผนให้มี LAN เป็นระบบเดียวกันทั้งองค์กรและทำให้เกิดปัญหาอย่างหนึ่งคือ จะไม่สามารถเชื่อมต่อเข้ากันได้ นอกจากนี้เทคโนโลยีของ LAN ของแต่ละระบบมีข้อจำกัดที่แตกต่างกันออกไป เช่นมีการจำกัดในเรื่องระยะทางที่ใช้เชื่อมต่อ (อย่างน้อยความยาวของสายจะต้องไม่เกิน 500 เมตร เป็นต้น)

นอกจากนี้ในช่วงปี ค.ศ 1960-1970 นักวิทยาศาสตร์และวิศวกรได้เริ่มพัฒนาระบบโครงข่ายที่มีการเชื่อมต่อระบบคอมพิวเตอร์ระยะทางไกลๆขึ้นมาซึ่งเราเรียกระบบโครงข่ายนี้ว่า (Wide Area Network หรือ Long-haul network : WAN) การเชื่อมต่อคอมพิวเตอร์ในลักษณะนี้จะใช้ Modem เพื่อส่งสัญญาณผ่านสายไปในระยะทางไกลๆ แต่ในระบบ WAN ที่พัฒนาขึ้นในขณะนั้น จะเชื่อมต่อคอมพิวเตอร์ได้เฉพาะระบบเท่านั้น คือไม่สามารถพ่วงระบบ LAN เข้าไปได้ทำให้ระบบโครงข่ายของคอมพิวเตอร์มีประสิทธิภาพไม่เต็มที่หลังจากปี ค.ศ 1970 เป็นต้นมา ARPA (Advanced Research Project Agency) ทำการค้นคว้าและวิจัยในเรื่องระบบโครงข่ายคอมพิวเตอร์โดยใช้เทคโนโลยี ของโครงข่ายต่างๆที่มีใช้อยู่ในขณะนั้น และ ARPA ได้นำเทคโนโลยีไปใช้ในทางทหารโครงการต่างๆที่ ARPA ได้ทำอยู่คือระบบ WAN ที่มีชื่อเรียกว่า “ARPANET” ซึ่งเป็นโครงข่ายที่ใช้ในการสื่อสารผ่านดาวเทียมและคลื่นวิทยุแต่ ARPA ก็มีปัญหาเหมือนกับองค์กรอื่นๆ คือระบบโครงข่ายที่แตกต่างกันไม่สามารถเข้ากันได้ ดังนั้น ARPA จึงจะพยายามที่จะแก้ไขปัญหานี้ โดยการระดมความคิดจากนักวิจัยต่างๆ การแก้ไขของ ARPA จึงประสบความสำเร็จคือสามารถเชื่อมต่อระบบ LAN และ WAN เข้าด้วยกันได้และระบบโครงข่ายนี้มีชื่อเรียกว่า “Internetwork” ซึ่งเราเรียกกันสั้นๆว่า “Internet”

แต่แนวความคิดทางอุดมคติของระบบโครงข่าย Internet ในตอนริเริ่มคือ ต้องการให้ระบบโครงข่ายทั้งหมดสามารถเชื่อมต่อถึงกันได้หมดทั้งระบบ ซึ่งก็คือ Internet ต้องเป็นระบบเปิด (Open

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเชิงงานเพื่อการศึกษเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System) และระบบเปิดก็มีแนวโน้มเป็นไปได้สูง เนื่องจากองค์กรขนาดใหญ่มีปัญหาทางด้านระบบเปิด คือองค์กรใหญ่ๆต้องมีคอมพิวเตอร์หลายๆเครื่อง จากหลายบริษัทเพื่อให้ได้ใช้งานตามวัตถุประสงค์ต่างๆกัน ดังนั้นระบบเปิดเท่านั้นที่สามารถเชื่อมต่อระบบโครงข่ายที่ต่างกันได้ สิ่งที่สนับสนุนระบบเปิดอีกอย่างก็คือ การทำเอกสารเผยแพร่สิ่งที่พัฒนาและคิดค้นขึ้นใหม่เกี่ยวกับ TCP/IP โดยที่ทีมงานวิจัยของ ARPA ได้จัดทำเอกสารทางเทคนิคเกี่ยวกับ TCP/IP ทั้งหมดในรูปแบบของไฟล์ข้อมูลคอมพิวเตอร์ และเก็บไว้ในระบบโครงข่ายของ ARPANET

อินเทอร์เน็ตประกอบไปด้วยเครือข่ายจำนวนมากต่อเชื่อมเข้าด้วยกันจนกลายเป็นเครือข่ายขนาดมหึมา เครือข่ายย่อยในอินเทอร์เน็ตมักเป็นเครือข่ายเฉพาะบริเวณที่อาจใช้เทคโนโลยีทางฮาร์ดแวร์ในเครือข่ายแตกต่างกันไป แต่ซอฟต์แวร์ในเครือข่ายจะทำงานภายใต้หลักสากลทำให้ทุกเครือข่ายสามารถแลกเปลี่ยนและส่งผ่านข้อมูลระหว่างกันได้ ในการส่งข้อมูลระหว่างเครื่องต้องมีการกำหนดชื่อผู้รับผู้ส่งในทำนองเดียวกับการส่งจดหมายทางไปรษณีย์ คอมพิวเตอร์ทุกเครื่องในอินเทอร์เน็ตต้องมีหมายเลขประจำตัว ดังนั้นผู้ใช้อินเทอร์เน็ตต้องทราบถึงวิธีการเรียกชื่อเครื่อง

2.1 ชื่อเครื่องในอินเทอร์เน็ต

ชื่อเครื่องหรือเรียกว่า ชื่อโฮสต์ (host name) ในอินเทอร์เน็ตมีวิธีเขียนเป็นมาตรฐาน เช่น kmitl.ac.th เป็นเครื่องของลาดกระบัง หรือ nuclues.nectec.or.th เป็นเครื่อง nontri ที่มหาวิทยาลัยเกษตร ชื่อจะบอกเป็นส่วน ๆ และบอกถึงองค์กรที่สังกัดโดยใช้เครื่องหมายจุดเป็นตัวแบ่ง หากนำชื่อเครื่องมาประกอบกับรหัสประจำตัวของผู้ใช้ซึ่งเรียกว่า ชื่อบัญชี (account name) ก็จะกลายเป็นที่อยู่ประจำตัวของผู้ใช้ในอินเทอร์เน็ตซึ่งใช้สำหรับการส่งจดหมายอิเล็กทรอนิกส์ได้ ที่อยู่ประจำของผู้ใช้อินเทอร์เน็ตจะใช้ชื่อบัญชีคั่นด้วยเครื่องหมาย "@" และต่อท้ายด้วยชื่อคอมพิวเตอร์ เช่นผู้ใช้ซึ่งมีบัญชีชื่อ s9013160 บนเครื่อง kmitl.ac.th จะมีที่อยู่ในอินเทอร์เน็ตดังนี้คือ

s9013160@kmitl.ac.th

ขอให้สังเกตว่าที่อยู่จะเขียนแบบต่อเนื่องกันไปโดยไม่มีช่องว่างแทรก ที่อยู่ข้างต้นจะเป็นที่อยู่ประจำตัวของ s9013160 เราสามารถส่งจดหมายอิเล็กทรอนิกส์ไปยัง s9013160 ได้โดยใช้คำสั่ง

mail s9013160@kmitl.ac.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนหลังของสัญลักษณ์ @ ซึ่งเป็นชื่อเครื่องนั้นเรานิยมเรียกโดยทั่วไปว่า โดเมน (Domain) ที่อยู่ทางอิเล็กทรอนิกส์ประจำตัวผู้ใช้ทุกคนจะมีรูปแบบดังนี้

ชื่อบัญชีผู้ใช้@โดเมน

2.2 การเชื่อมต่อเข้าอินเทอร์เน็ต

ในยุคของเครือข่ายคอมพิวเตอร์ ไมโครคอมพิวเตอร์หรือพีซีประจำบ้านสามารถใช้เชื่อมโยงเข้าสู่อินเทอร์เน็ตได้ เครื่องที่เชื่อมต่อเข้าอินเทอร์เน็ตมีอยู่อย่างหลากหลายชนิดโดยไม่จำกัดประเภทของฮาร์ดแวร์และซอฟต์แวร์ ในศูนย์บริการใหญ่ ๆ เรามักพบเครื่องระดับสถานีวิศวะกรรมใช้ซีพียูสมรรถนะสูง มีจอภาพขนาดใหญ่ความละเอียดสูง ทำงานด้วยระบบปฏิบัติการยูนิกซ์เป็นระบบเครือข่ายเฉพาะบริเวณ (LAN) และต่อทั้งเครือข่ายเข้าอินเทอร์เน็ต แต่ภายในบ้านเราสามารถใช้อุปกรณ์ใดเครื่องหนึ่งที่ทำงานด้วยวินโดวส์หรือคอสเพื่อต่อเข้ากับระบบอินเทอร์เน็ตได้เช่นกันการเชื่อมคอมพิวเตอร์เข้าอินเทอร์เน็ตมีวิธีการหลัก 3 วิธีดังนี้

1. การต่อเป็นเทอร์มินัล เป็นการต่อโดยใช้โมเด็มผ่านสายโทรศัพท์ โดยใช้ซอฟต์แวร์จำลองพีซีเป็นเทอร์มินัลของโฮสต์
2. การต่อด้วยโปรโตคอล SLIP/PPP ใช้การต่อเชื่อมด้วยโมเด็มผ่านสายโทรศัพท์หรือสายสื่อสาร และใช้โปรโตคอล SLIP หรือ PPP
3. การต่อเข้าเครือข่าย ต่อพีซีเข้ากับเครือข่ายซึ่งเชื่อมอยู่กับอินเทอร์เน็ตอยู่ก่อนแล้ว

ทั้งสามแบบช่วยให้ใช้งานอินเทอร์เน็ตแบบออนไลน์ได้เช่นเดียวกัน แต่การต่อแบบแรกจะมีข้อจำกัดในการใช้บริการบางประเภทโดยเฉพาะด้านกราฟฟิก ส่วน 2 แบบถัดมาเป็นการต่อเชื่อมที่ทำให้พีซีเป็นส่วนหนึ่งของระบบอินเทอร์เน็ตโดยสมบูรณ์ และสามารถใช้งานเครือข่ายได้เต็มอย่างพิถีพิถัน

2.2.1 การต่อแบบเทอร์มินัล

การต่อเชื่อมต่อแบบนี้ พีซีจะทำหน้าที่เป็นเพียงเทอร์มินัลซึ่งหมายถึงอุปกรณ์อินพุตสำหรับป้อนคำสั่ง และเป็นอุปกรณ์ Output สำหรับแสดงผล การทำงานตามคำสั่งจะเกิดขึ้นที่คอมพิวเตอร์ที่ศูนย์บริการ การเชื่อมต่อแบบนี้ใช้อุปกรณ์จำนวนน้อยและไม่สลับซับซ้อน อุปกรณ์ที่จำเป็นต้องใช้เพิ่มเติมได้แก่ โมเด็มสำหรับติดเข้ากับพีซีและซอฟต์แวร์ที่เรียกว่า ตัวจำลองเทอร์มินัลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(TerminalEmulator) ทำหน้าที่สื่อสารกับคอมพิวเตอร์ตัวจำลองเทอร์มินัลจะแปลงพีซีให้เป็นเสมือนเทอร์มินัลตัวหนึ่งของคอมพิวเตอร์ที่ศูนย์บริการ การเชื่อมต่อแบบนี้จะมีลักษณะดังรูปที่ 2.2.1 การเชื่อมต่อแบบเทอร์มินัลมีข้อจำกัดการใช้งานอยู่มาก ทั้งนี้เพราะพีซีทำหน้าที่เป็นเพียงช่องทางขอใช้โฮสต์ที่ศูนย์บริการโดยไม่ถือว่าพีซีนั้นเป็นส่วนหนึ่งของอินเทอร์เน็ต ข้อสำคัญในการเชื่อมต่อ



รูปที่ 2.2.1 การต่อคอมพิวเตอร์ให้เป็นเทอร์มินัลอาศัย โมเด็มเพื่อรับส่งข้อมูลด้วยสายโทรศัพท์

คอมพิวเตอร์เป็นเทอร์มินัลคือ คำสั่งที่ป้อนจากพีซีจะถูกส่งไปปฏิบัติงานที่โฮสต์ข้อมูลจะไม่ส่งตรงมายังคอมพิวเตอร์ที่บ้าน ตัวอย่างเช่นหากว่าฝากโอนย้ายแฟ้มด้วย ftp จะเป็นการโอนย้ายข้อมูลกับโฮสต์ ข้อมูลจะยังไม่ส่งตรงมายังคอมพิวเตอร์ที่บ้าน ถ้าหากการโอนย้ายมายังพีซีจะต้องใช้โปรแกรมโอนย้ายอีกชั้นหนึ่งเช่นโปรแกรม sz ที่มีในโฮสต์ หรือใช้คำสั่งโอนย้ายจากโปรแกรมจำลองเทอร์มินัลเอง การใช้งานแบบนี้ยังถูกจำกัดอยู่เฉพาะโปรแกรมประยุกต์ที่เป็นแบบตัวอักษร ถึงแม้ว่าจะใช้วินโดวส์ก็ไม่สามารถใช้โปรแกรมประยุกต์บนโฮสต์ที่ทำงานแบบกราฟฟิกอย่างเช่นบริการ www ได้ นอกจากนี้ยังจำกัดการใช้งานหนึ่งช่องหน้าต่างเท่านั้น การเชื่อมต่อพีซีเป็นเทอร์มินัลเหมาะกับการใช้งานประจำบ้านซึ่งคิดคั้งและดูแลได้ง่ายโดยใช้อุปกรณ์ฮาร์ดแวร์น้อยชิ้นและซอฟต์แวร์ที่ใช้งานก็ไม่สลับซับซ้อน

2.2.2 การเชื่อมต่อด้วยโปรโตคอล

หากต้องการให้คอมพิวเตอร์เป็นส่วนหนึ่งของอินเทอร์เน็ตและสามารถใช้บริการรูปแบบต่างๆ ในอินเทอร์เน็ตได้อย่างสมบูรณ์จำเป็นต้องเชื่อมต่อด้วยโปรโตคอลที่ซีพี/ไอพีไป การต่อแบบนี้ยังอาจเรียกว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องอาศัยโมเด็มหรือใช้สายสื่อสารอนุกรม จึงจำเป็นต้องมีโปรโตคอลเพิ่มเติมซึ่งได้แก่ SLIP หรือ PPP

SLIP (Serial Line Interface protocol) เป็นโปรโตคอลที่ออกแบบสำหรับสายสื่อสารอนุกรม ต่อมาภายหลังได้มีการพัฒนาโปรโตคอลขึ้นใหม่คือ PPP (Point to Point Protocol) เพื่อให้สามารถใช้งานร่วมกับโปรโตคอลอื่นได้ SLIP และ PPP เป็นโปรโตคอลต่างแบบกันที่ไม่สามารถทำงานร่วมกันได้ คอมพิวเตอร์ปลายทางทั้งสองด้านจึงต้องใช้ซอฟต์แวร์โปรโตคอลตัวใดตัวหนึ่งที่ตรงกัน โดยในที่นี้จะกล่าวถึงเฉพาะโปรโตคอล SLIP เท่านั้น

รูปแบบการเชื่อมต่อแบบ SLIP มีลักษณะดังรูปที่ 2.2.2 หากมองโดยผิวเผินแล้วจะไม่มี ความแตกต่างทางด้านการจัดวางอุปกรณ์ที่บ้านแต่อย่างใด เพราะยังคงใช้คอมพิวเตอร์โมเด็ม สาย โทรศัพท์อยู่เช่นเดิม ส่วนแตกต่างได้แก่การติดตั้งซอฟต์แวร์และการใช้ซอฟต์แวร์ ตลอดจนการเชื่อมต่อทางด้านปลายทางที่ศูนย์บริการซึ่งต้องเตรียมอุปกรณ์ฮาร์ดแวร์เพื่อการเชื่อมต่อไว้



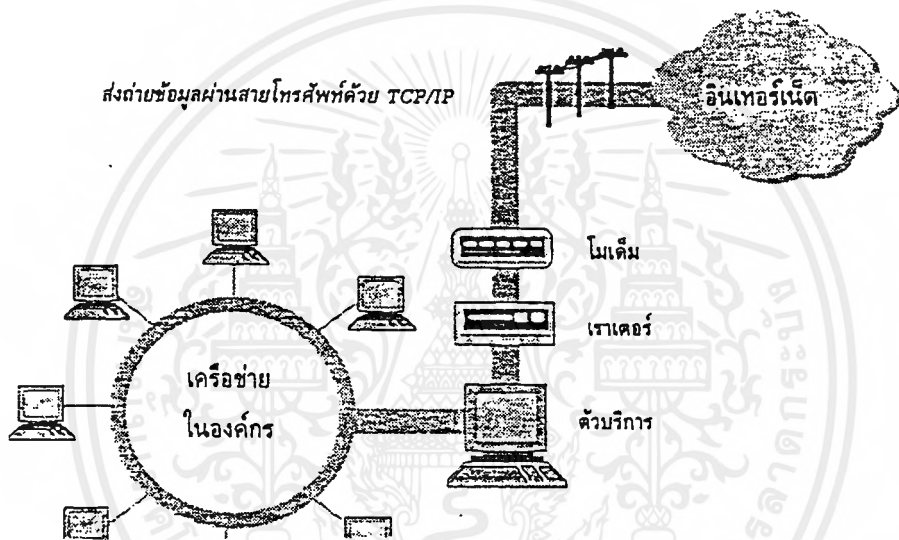
รูปที่ 2.2.2 การต่อเชื่อมแบบใช้โปรโตคอล

การเชื่อมต่อด้วย SLIP โดยหมุนโทรศัพท์เป็นการเชื่อมต่อแบบชั่วคราวเพราะเครื่องจะเป็น โฮสต์เมื่อหมุนโทรศัพท์เขาศูนย์บริการและหมดสภาพเป็นโฮสต์เมื่อวางสายโทรศัพท์ แต่ถ้าสายที่ใช้ งานเป็นสายเช่าหรือวางสายเองก็สามารถใช้ SLIP แบบถาวรได้

พีซีที่ใช้โปรโตคอล SLIP หรือ PPP จะมีหมายเลขไอพีประจำตัวและทำหน้าที่เป็นโฮสต์ตัว หนึ่งในอินเทอร์เน็ต ซึ่งต่างกับแบบเทอร์มินัลที่ทำหน้าที่เป็นเพียงอุปกรณ์ปลายทางโดยไม่มีหมายเลขไอพีประจำ หากพีซีทำงานด้วยยูนิคซ์หรือระบบปฏิบัติการชนิดมัลติยูสเซอร์ก็สามารถมีผู้ใช้ได้ หลายคนพร้อมกัน หรือใช้งานได้หลายงานได้ในเวลาเดียวกันนั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การเชื่อมเครือข่ายเข้าอินเทอร์เน็ต

หากมีเครือข่ายคอมพิวเตอร์อยู่แล้ว เราสามารถต่อเครือข่ายเข้ากับศูนย์บริการและติดตั้งซอฟต์แวร์ที่ซีพี/ไอพี เครือข่ายจะเป็นส่วนหนึ่งของอินเทอร์เน็ตทันที คอมพิวเตอร์ภายในเครือข่ายจะทำงานโดยใช้โปรโตคอลที่ซีพี/ไอพีอย่างเต็มตัว โดยมีหมายเลขไอพีเป็นของตัวเองและถือเป็นโฮสต์ในอินเทอร์เน็ตด้วย การต่อเครือข่ายเข้ากับอินเทอร์เน็ตมักต้องการสายเชื่อมโยงแบบถาวรซึ่งอาจเป็นสายเช่าจากองค์การโทรศัพท์ หรือการเดินสายเชื่อมต่อเองเพื่อให้ต่อเชื่อมได้ตลอดเวลา การเชื่อมต่อจะมีลักษณะดังรูปที่ 2.2.3



รูปที่ 2.2.3 การเชื่อมเครือข่ายเข้าอินเทอร์เน็ต

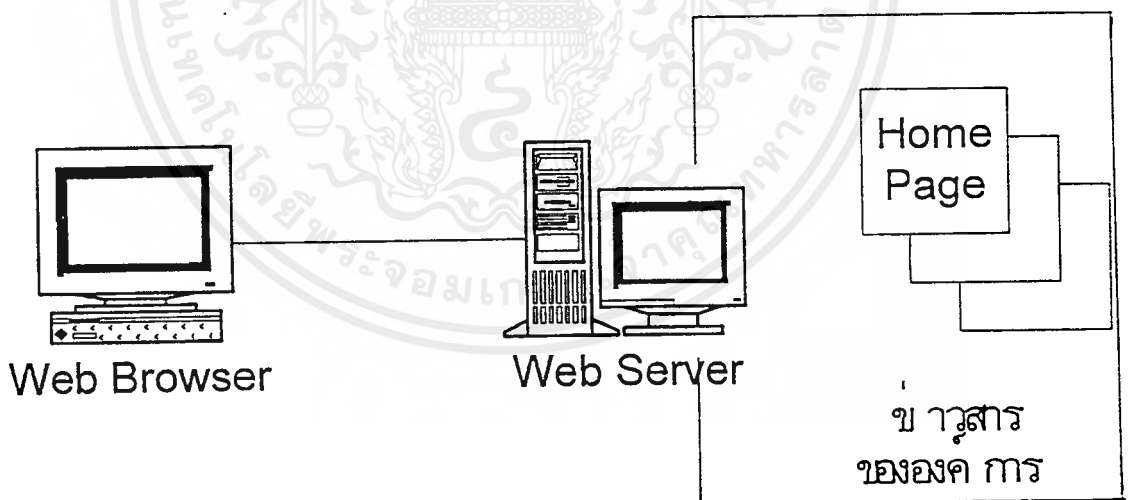
2.3 World Wide Web

World Wide Web หรือ WWW หรือจะเรียกสั้น ๆ ว่า เว็บบ (Web) เป็นรูปแบบหนึ่งของระบบการเชื่อมโยงเครือข่ายข่าวสาร ใช้ในการค้นหาข้อมูลข่าวสารบนอินเทอร์เน็ตจากแหล่งข้อมูลหนึ่งไปยังแหล่งข้อมูลที่อยู่ห่างไกลออกไปให้มีความง่ายต่อการใช้งานมากที่สุด WWW จะแสดงผลอยู่ในรูปแบบของเอกสารที่เรียกว่า ไฮเปอร์เท็กซ์ (Hyper Text) ซึ่งเป็นฐานข้อมูลชนิดหนึ่งทำหน้าที่รวบรวมข่าวสารข้อมูลที่อยู่กระจัดกระจายไปในที่ต่าง ๆ ทั่วโลกให้สามารถนำมาใช้งานได้เสมือนอยู่ในที่เดียวกันคล้ายกับเส้นใยแมงมุมที่ถักทอเส้นสายเชื่อมโยงกันไปมา แม้ว่าจะมีเส้นใยจำนวนมาก แต่ละเส้นจะถูกจัดวางทับกันมีจุดเชื่อมต่อที่ทำให้ตัวแมงมุมสามารถที่จะเดินทางไปยังจุดใด ๆ บนเส้นใยเหล่านี้ได้ การให้บริการของอินเทอร์เน็ตแบบ WWW เป็นระบบงานที่ทรงพลังมากในยุคปัจจุบันทำให้ระบบเครือข่ายอินเทอร์เน็ตเป็นที่นิยมใช้กันอย่างแพร่หลายทั่วโลก เพราะสามารถให้บริการข้อมูลได้ทั้งในแบบข้อความ เสียง ภาพนิ่ง และภาพเคลื่อนไหว ทำให้ข้อมูลที่นำเอกสารนี้เป็นเอกสารที่ส่งวนไวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาแสดงมีความน่าสนใจ การค้นหาข้อมูลจะมีการเชื่อมโยงข้อมูลตามเส้นทางที่กำหนดไว้เรียกว่า LINKS โดยในแต่ละจุดเชื่อมโยงจะมีเส้นทางเดิน ไปยังจุดต่อไปติดตั้งอยู่เป็นระยะในรูปแบบของข้อความ หรือรูปภาพ ทำให้ผู้ใช้สามารถเลือกที่จะเดินทางต่อไปยังจุดมุ่งหมายใดก็ได้ และที่นั่นผู้ใช้จะพบว่ายังมีทางเดินไปถึงจุดหมายอื่นได้อีก ทำให้สามารถค้นหาข้อมูลที่มีรายละเอียดมากยิ่งขึ้น

ในช่วงแรก การใช้งานระบบอินเทอร์เน็ตจะมีแค่การส่งแฟ้มข้อมูล และการส่งจดหมายที่เป็นข้อความธรรมดาเท่านั้น ทำให้มีการใช้งานอยู่ในจำนวนจำกัด เมื่อเทคโนโลยีคอมพิวเตอร์ โดยเฉพาะเทคโนโลยีการแสดงผลทำได้ดีขึ้น Software ต่างๆ จึงเปลี่ยนการแสดงผลไปทางภาพ ซึ่งสามารถสื่อสารกับผู้ใช้ได้ง่ายขึ้นเริ่มมีการใช้ระบบเชื่อมโยงข้อมูลในเครือข่าย ที่เรียกว่า WWW โดยใช้โปรแกรมค้นหาและแสดงข้อมูล (Browser) เช่น Netscape และ Internet Explorer ซึ่งสามารถแสดงข้อมูลเป็นภาพที่สวยงาม แสดงภาพเคลื่อนไหว หรือมีเสียงประกอบด้วยก็ได้

นอกจากจะส่งเอกสารที่มีทั้งข้อความ ภาพ และ เสียงไปพร้อมกันได้แล้วระบบ WWW ยังใช้งานได้ง่ายโดยไม่เสียค่าใช้จ่ายมาก จึงมีผู้นำไปใช้ในวงการธุรกิจ โดยการตั้งหน่วยบริการข้อมูลที่เรียกว่า Web Site ขึ้นมาเมื่อถูกคำติดต่อผ่านระบบอินเทอร์เน็ตเข้ามา สิ่งแรกที่ลูกค้าเห็นก็คือ Home Page ซึ่งจะมีภาพและข้อความต้อนรับพร้อมกับบอกให้รู้ว่า มีข้อมูลอะไรที่ลูกค้าสามารถดูได้ดังรูป



รูปที่ 2.3.1 รูปแสดงการทำงานของ Web Site

จากรูปลูกค้าสามารถจะชี้แล้วกดปุ่มขอข้อมูลที่ต้องการได้ มีผู้ประมาณว่า ในปัจจุบันมีข้อมูลใน Web อยู่ในอินเทอร์เน็ตมากกว่า 20 ล้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบเครือข่าย

อินเทอร์เน็ต คือ ระบบที่ช่วยในการติดต่อสื่อสารระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ ซึ่งอาจมีโครงสร้างการทำงานที่ต่างกัน ดังนั้นโครงสร้างของระบบจึงแบ่งออกเป็น 2 ส่วน ส่วนแรก เป็นส่วนของการติดต่อสื่อสารและรับส่งข้อมูล ซึ่งทำให้การเชื่อมโยงของระบบเครือข่ายเป็นไปได้ โดยมีข้อกำหนดต่างๆ ในการเชื่อมโยงซึ่งรวมเรียกว่า TCP/IP ส่วนที่สอง เป็นส่วนการใช้งานซึ่งเป็นการใช้ประโยชน์จากระบบเครือข่าย

แต่เดิมอินเทอร์เน็ตมีใช้อยู่ในระบบ UNIX เท่านั้นแต่ในปัจจุบัน ไม่ว่าจะเป็นเครื่องคอมพิวเตอร์ที่ใช้ระบบ UNIX, Microsoft หรือ Macintosh ต่างก็มีเครื่องมือสำหรับระบบอินเทอร์เน็ตพร้อมอยู่ในตัวหรือเพิ่มเข้าไปได้อย่างง่ายดาย มีทั้งระบบเครือข่ายและ Software ต่างๆ ที่ใช้กับอินเทอร์เน็ต แต่เครื่องมือเหล่านี้ก็มีได้นำมาใช้งาน เนื่องจากคนส่วนใหญ่ไม่เข้าใจว่าถึงแม้ระบบจะแตกต่างกันแต่ระบบเครือข่าย TCP/IP ก็สามารถติดต่อถึงกันได้ ถึงแม้ว่าจะใช้ Software ตัวเดียวกัน แต่ระบบข่าวสารภายในองค์กร มีความเหนือกว่าระบบ WWW ตรงที่การเชื่อมโยงเครือข่ายภายในองค์กร เป็นการเชื่อมโยงระบบ LAN ที่มีความเร็วข้อมูลสูงถึง 10Mbps ซึ่งเร็วกว่าการติดต่อผ่าน Modem โดยทั่วไปมีความเร็วต่ำกว่า 100 Kbps ทำให้ไม่มีปัญหาในเรื่องเวลาที่ใช้ในการส่งข้อมูลภาพ นอกจากนี้ยังสามารถติดต่อกันโดยใช้เสียงและภาพเคลื่อนไหวได้อีกด้วย

2.5 การเชื่อมต่อระหว่าง UNIX กับ Microsoft Windows

เมื่อระบบเครือข่ายของเรามีเครื่องแม่ข่ายที่ใช้ระบบต่างกัน มีทั้งใช้ Windows NT และที่ใช้ UNIX ปัญหาที่เราจะรู้แน่ๆ ก็คือระบบ Microsoft และระบบ UNIX ต่างก็ใช้ข้อจำกัดการสื่อสารที่ต่างกัน ในปัจจุบันมีการติดตั้งใช้งาน Windows ในระบบเครือข่าย UNIX มากขึ้น การที่เครื่องในระบบหนึ่งต้องการใช้แฟ้มข้อมูลที่อยู่บนเครื่องที่อยู่อีกระบบหนึ่งจึงเป็นสิ่งที่เป็นที่พบเห็นอยู่เสมอ ดังนั้นวิธีที่จะทำให้ทั้งสองระบบร่วมกันได้ ก็คือ จัดการให้ทั้งสองฝ่ายใช้ระบบแฟ้มข้อมูลแบบเดียวกัน ซึ่งก็มาถึงปัญหาที่ต้องเลือกว่า จะให้ฝ่ายไหนเป็นฝ่ายปรับตัวไปเข้ากับอีกฝ่ายหนึ่ง

ถ้าเราเลือกที่จะปรับเปลี่ยนทางด้านระบบ UNIX ก็จะทำให้ได้สองวิธี

1.คือหา Program ที่สามารถติดต่อกับข่ายของ Microsoft ได้โดยตรงมาใช้งานบนเครื่องที่เป็นลูกข่าย UNIX ทำให้มันสามารถติดต่อกับเครื่องแม่ข่ายที่เป็น Windows NT ได้โดยไม่ต้องเพิ่มบริการอะไรในตัวแม่ข่ายนั้นแต่ปัญหาไม่ได้อยู่ที่การค้นหา Program ดังกล่าวมาใช้งาน เพราะ จะมีคุณภาพที่ยังมีมาตรฐานที่แตกต่างกันค่อนข้างมาก

2.เป็นการเพิ่มบริการเครือข่ายที่ไม่ใช่บริการของ UNIX บนเครื่องแม่ข่าย UNIX เช่นบริการเครือข่ายของ NetWare (IPX/SPX) หรือบริการเครือข่ายของ Microsoft และถ้าเครื่องลูกข่ายระบบ Windows มีบริการดังกล่าวอยู่ มันก็สามารถจะติดต่อกับตัวแม่ข่าย UNIX ได้โดยตรง ข้อเสียของวิธีนี้

ก็คือมันจะทำให้เครื่องแม่ข่ายมีความซับซ้อนเพิ่มขึ้น อาจเป็นปัญหาต่อการจัดการและบริหารตัวแม่ข่ายได้

ถ้าเราเลือกปรับเปลี่ยนด้าน Windows

ถ้าเราเลือกที่จะไม่ไปยุ่งกับตัว UNIXเลยเราต้องมาดูว่าจะนำเอาระบบอะไรมาเพิ่มเติมให้กับระบบ Microsoft คำตอบก็คือใช้ระบบเพิ่มข้อมูลเครือข่าย NFS (Network File System) ซึ่งเป็นระบบที่นับได้ว่ามีประสิทธิภาพ และจัดการได้ง่าย ในการใช้เพิ่มข้อมูลและเครื่องพิมพ์ร่วมกันในเครือข่าย NFS เป็นระบบที่ถูกพัฒนาขึ้นมากกว่าสิบปีแล้ว โดยบริษัท Sun Microsoft เพื่อใช้กับระบบ UNIX ที่มาจากค่ายต่างๆกัน สามารถใช้เพิ่มและข้อมูลร่วมกันได้ ผู้ใช้จากเครื่องหนึ่งสามารถที่จะเรียกใช้เพิ่มจากอีกเครื่องหนึ่งได้ โดยไม่ต้องรู้เลยว่าเพิ่มนั้นอยู่ที่เครื่องไหน

NFS ก็ได้กลายเป็นมาตรฐานอุตสาหกรรมของระบบเพิ่มข้อมูลแบบกระจาย และในปัจจุบันบริษัทต่างๆได้ซื้อสิทธิในการซื้อ NFS เพื่อนำไปใช้ในระบบของตน ดังนั้นการ Software ใดๆเกี่ยวกับ NFS มาใช้ในระบบ Microsoft จึงเป็นเรื่องที่ง่ายกว่าการเปลี่ยนแปลงระบบ UNIX

การติดตั้ง NFS สำหรับเครื่องลูกข่าย

ในการนำ NFS มาใช้ในระบบ Microsoft เพื่อให้มันติดต่อกับระบบ UNIX นั้นวิธีแรกที่จะพูดถึงก็คือใช้ Software สำหรับลูกข่าย NFS ในเครื่องลูกข่าย Windows ที่ต้องการติดต่อโดยตรงกับระบบแม่ข่าย UNIX ที่เปิดให้บริการ NFS โดยบริการนี้มีอยู่แล้วในระบบ UNIX ส่วนใหญ่ ขึ้นอยู่กับว่าจะเปิดใช้หรือไม่เท่านั้น

เนื่องจาก NFS เป็นระบบมาตรฐานที่เปิดกว้าง ลูกข่าย NFS สามารถติดต่อกับเครื่องใดๆ ที่ติดตั้งระบบ NFS เหมือนกันได้ ข้อดีของวิธีนี้ก็คือ ทำให้ระบบสามารถติดต่อกับระบบแม่ข่าย UNIX ได้อย่างสมบูรณ์ โดยที่ตัวของแม่ข่าย UNIX ไม่ต้องมี Software อะไรเพิ่มเติมเลย

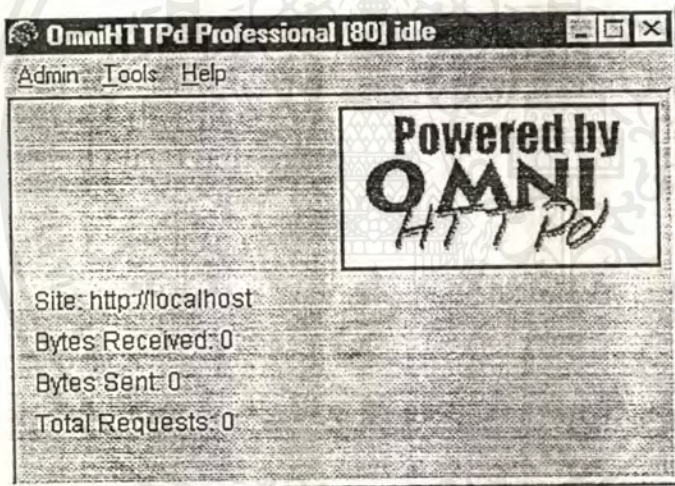
การติดตั้ง NFS สำหรับเครื่องแม่ข่าย

ในเครือข่ายที่มีสมาชิกในระบบทั้งสองแบบอยู่ค่อนข้างมาก การที่จะปรับปรุงเครื่องลูกข่ายที่มีจำนวนมาก คงเป็นเรื่องที่ไม่สะดวกนัก วิธีที่น่าได้ผลก็คือ การนำระบบ NFS ไปใส่ใน Windows NT ของเครื่องแม่ข่าย โดยเอาเครื่องแม่ข่ายของ Windows NT เป็น Gateway เพื่อเปิดให้ลูกข่ายของระบบเครือข่าย Microsoft มองเห็นและใช้ทรัพยากรต่างๆ ของระบบ UNIX ได้ เมื่อลูกข่ายเหล่านี้ติดต่อกับที่ระบบแม่ข่าย Windows NT ตามปรกติ ก็จะมองเห็นทรัพยากรดังกล่าวได้ด้วยวิธีนี้ จึงไม่ต้องเพิ่ม Software ใดๆ ให้กับเครื่องลูกข่ายเลย

ในทางกลับกัน ก็จะบริการ NFS บน Windows NT เพื่อเปิดให้ระบบลูกข่ายของระบบ UNIX มองเห็นและใช้ทรัพยากรต่างๆ ของ Windows NT ได้ โดยลูกข่าย UNIX จะติดต่อใช้ทรัพยากรเหล่านั้นผ่านทางระบบ NFS ของตนเองที่มีอยู่แล้ว ข้อดีของการเลือกใช้วิธีนี้ นอกจากจะไม่ต้องหาตัวไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Software มาเพิ่มเติมให้กับเครื่องลูกข่ายทั้งในระบบ Microsoft และ UNIX แล้ว Software และ Hardware ของเครื่องแม่ข่าย Windows NT ก็มักมีราคาถูกลงกว่าของเครื่องแม่ข่าย UNIX นอกจากนี้ การจัดการกับระบบ Windows NT ยังทำได้ง่ายกว่ากันมากสำหรับข้อเสียที่มีอยู่ก็คือ มันทำให้ตัวแม่ข่าย Windows NT มีความซับซ้อนขึ้นและอาจต้องมีการเปลี่ยนแปลงไปใช้ในระบบที่ดีกว่าเดิม เพื่อให้ได้ประสิทธิภาพตามต้องการ

ดังนั้นความสามารถในการจัดเก็บระบบเพิ่มข้อมูลของ NFS ในระบบ UNIX บวกกับความสามารถในการบริหารของระบบ Windows NT ทำให้การผสมกันระหว่าง ระบบเครือข่ายทั้งสองระบบเป็นไปได้ด้วยดี และเนื่องจาก NFS เป็นตัวกลางที่ดีมากในการเชื่อมโยงระบบที่แตกต่างกัน มันจึงทำให้เป็นวิธีที่มีประสิทธิภาพอย่างยิ่งอีกวิธีหนึ่ง ในการรวมระบบเพิ่มข้อมูลและบริการเครื่องพิมพ์ ระหว่างเครือข่าย Microsoft และ UNIX



รูปที่ 2.5.1 เป็นรูปของ Web Server ที่ใช้ติดต่อกับระบบ UNIX ได้

2.6 Home Page

โฮมเพจเป็นผลผลิตของไฮเปอร์เท็กซ์ที่ผู้เข้าไปใช้บริการบนอินเทอร์เน็ตพบเห็นกันได้มากที่สุด และเป็นจุดเด่นให้ผู้ให้บริการเกิดความสนใจที่จะใช้อินเทอร์เน็ตมากขึ้นนอกจากบริการพื้นฐานทั่วไป ผู้ให้บริการอินเทอร์เน็ตและศูนย์บริการเว็บทุกแห่งต่างมีโฮมเพจเป็นของตนเองเพื่อเป็นข้อมูลเบื้องต้นที่แสดงให้ผู้ใช้บริการทราบว่าในสถานี่ นั้น ๆ ให้บริการสิ่งใดบ้างและเมื่อเข้าสู่โฮมเพจนั้น ๆ แล้ว เราจะเดินทางไปยังแห่งใดได้ ทำหน้าที่เป็นจุดรวมของการเดินทางเข้าไปสู่ดินแดนแห่งใหม่ และเราเรียกข้อมูลอื่น ๆ ซึ่งอาจเชื่อมโยงต่อจากโฮมเพจว่า เว็บเพจ (Web Page)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบยูนิคซ์

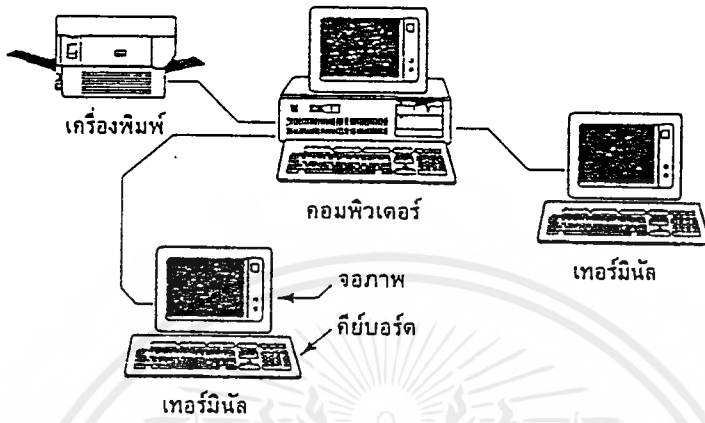
ระบบยูนิคซ์ ระบบปฏิบัติการที่เป็นที่รู้จักกันอย่างกว้างขวาง เริ่มต้นจากภายในรั้วมหาวิทยาลัย แพร่ขยายออกไปสู่วงการธุรกิจต่างๆ ระบบยูนิคซ์เป็นระบบปฏิบัติการที่มีใช้อยู่ในเครื่องคอมพิวเตอร์หลาย ๆ ระดับตั้งแต่เครื่องพีซีไปจนถึงระดับมินิคอมพิวเตอร์ หรือระดับเครื่องเมนเฟรม และในยุคของระบบเปิด เช่น ทุกวันนี้ ยูนิคซ์ถูกยกให้เป็นระบบปฏิบัติการสำหรับระบบเปิด ในแง่ของความสามารถในการทำงานนั้น ระบบยูนิคซ์เป็นระบบปฏิบัติการที่สามารถดึงความสามารถของเครื่องคอมพิวเตอร์ออกมาได้อย่างเต็มประสิทธิภาพ ยิ่งทุกวันนี้ความสามารถของเครื่องคอมพิวเตอร์ถูกพัฒนาไปอย่างมากมาย จนระบบปฏิบัติการบางตัวพัฒนาตามไม่ทัน ยูนิคซ์ก็就会被เลือกขึ้นมาในกรณีนี้ ระบบยูนิคซ์ไม่ใช่เป็นระบบปฏิบัติการที่สามารถทำงานให้กับผู้ใช้เพียงคนเดียวเท่านั้น แต่ยูนิคซ์ยังสามารถทำงานให้กับผู้ใช้หลาย ๆ คน พร้อม ๆ กันบนเครื่องคอมพิวเตอร์เครื่องเดียว เป็นการใช้ทรัพยากรที่มีอยู่อย่างเต็มประสิทธิภาพ

3.1 ส่วนประกอบของระบบยูนิคซ์

ระบบยูนิคซ์พื้นฐานจะประกอบด้วย เครื่องคอมพิวเตอร์หลักหนึ่งเครื่องและอุปกรณ์ต่อพ่วงต่าง ๆ ต่ออยู่ด้วยกัน ระบบยูนิคซ์ที่เรากำลังใช้กันอยู่อาจจะประกอบด้วยเครื่องคอมพิวเตอร์หลักหนึ่งเครื่องซึ่งติดตั้งระบบยูนิคซ์อยู่ เครื่องพิมพ์หนึ่งเครื่องเป็นเครื่องพิมพ์ของระบบและเทอร์มินัลจำนวนหนึ่งซึ่งเราอาจจะใช้เครื่องไมโครคอมพิวเตอร์มาทำเป็นเทอร์มินัลก็ได้กลุ่มของส่วนประกอบเหล่านี้เรียกว่าส่วนของฮาร์ดแวร์ระบบยูนิคซ์พื้นฐานจะมีลักษณะการต่ออุปกรณ์ต่าง ๆ เหมือนดังรูปที่ 3.1

เทอร์มินัลจะเป็นส่วนประกอบของระบบยูนิคซ์ซึ่งเราจะเกี่ยวข้องและเป็นส่วนที่คุ้นเคยมากที่สุด โดยที่จะใช้เรียกไปยังระบบยูนิคซ์เมื่อต้องการจะใช้งาน โปรแกรมต่าง ๆ เราจะทำทุกอย่างป้อนข้อมูล ดูผลลัพธ์การทำงานที่เทอร์มินัล เทอร์มินัลจะประกอบด้วยแป้นพิมพ์สำหรับป้อนข้อความหรือคำสั่งต่าง ๆ และจอภาพ ซึ่งมีลักษณะเหมือนกับจอภาพของเครื่องรับโทรทัศน์ เมื่อเราพิมพ์ข้อความทางแป้นพิมพ์ตัวอักษรที่พิมพ์จะแสดงออกทางจอภาพ ถ้าหากเป็นการพิมพ์คำสั่งให้เครื่องคอมพิวเตอร์หลักของระบบยูนิคซ์ทำงาน ผลลัพธ์จากการทำงานก็จะแสดงให้เราทราบทางจอภาพ หรือบางครั้งเราอาจจะกำหนดให้พิมพ์ผลลัพธ์ทางเครื่องพิมพ์ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ส่วนประกอบของระบบยูนิคซ์อย่างง่าย

ในเครื่องคอมพิวเตอร์หลักจะมีชุดคำสั่งอยู่ส่วนหนึ่งทำงานอยู่ตลอดเวลา เพื่อคอยรับคำสั่งที่เราส่งจากเทอร์มินัลและนำไปใช้งานตามคำสั่งนั้น ชุดของคำสั่งเหล่านี้จะเป็นส่วนซอฟต์แวร์ ซึ่งถูกพัฒนาขึ้นโดยนักเขียนโปรแกรมเพื่อให้คอมพิวเตอร์ทำงานต่าง ๆ ในทุกระบบคอมพิวเตอร์ ซอฟต์แวร์จะเป็นส่วนที่สั่งให้ฮาร์ดแวร์ทำงานตามที่ผู้ใช้ต้องการ ส่วนที่ทำให้ระบบยูนิคซ์ต่างจากระบบคอมพิวเตอร์อื่น ๆ ก็คือส่วนซอฟต์แวร์ของระบบยูนิคซ์ ซึ่งจะเป็นส่วนของชุดคำสั่งที่ทำให้ฮาร์ดแวร์ส่วนต่าง ๆ ของระบบทำงานด้วยกันได้อย่างมีประสิทธิภาพและรวดเร็ว

ซอฟต์แวร์ระบบยูนิคซ์ที่จริงก็คือระบบปฏิบัติการเหมือน ๆ กับระบบปฏิบัติการเอ็มเอส-ดอสที่นิยมใช้ในเครื่องไมโครคอมพิวเตอร์ ระบบปฏิบัติการจะเป็นส่วนที่คอยทำงานตามคำสั่งที่ผู้ใช้หรือตามที่โปรแกรมแอปพลิเคชันอื่นสั่ง เช่น แสดงข้อความการเก็บข้อมูลหนึ่งเรคอร์ด หรือการพิมพ์ข้อความต่าง ๆ เป็นต้น

ระบบปฏิบัติการยูนิคซ์ไม่เหมือนกับระบบปฏิบัติการระบบอื่น ๆ ตรงที่สามารถทำงานหลายคำสั่งได้พร้อมกันในเวลาเดียวกัน ตัวอย่างเช่น สมมติว่าผู้ใช้ต้องการดูข้อมูลที่เก็บอยู่ในฐานข้อมูล จึงพิมพ์คำสั่งสั่งให้เครื่องคอมพิวเตอร์ทำงานหาข้อมูลที่ต้องการให้ ในขณะที่เครื่องคอมพิวเตอร์กำลังค้นหาข้อมูลอยู่นั้น ผู้ใช้สามารถที่จะเรียกโปรแกรมเอดิเตอร์มาร่างจดหมาย และส่งส่งผ่านระบบเอกสาร เป็นเอกสารที่ส่งวันเวลาหรือการเขียนเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปรษณีย์อิเล็กทรอนิกส์ไปยังเพื่อนร่วมงานคนอื่น ๆ ได้ในเวลาเดียวกัน ระบบปฏิบัติการยูนิกซ์สามารถทำงานให้เราได้มากกว่าหนึ่งงานในเวลาเดียวกัน หรือศัพท์ทางเทคนิคจะเรียกระบบยูนิกซ์ว่าเป็นระบบปฏิบัติการแบบมัลติทาสก์ (Multitasking Operating System)

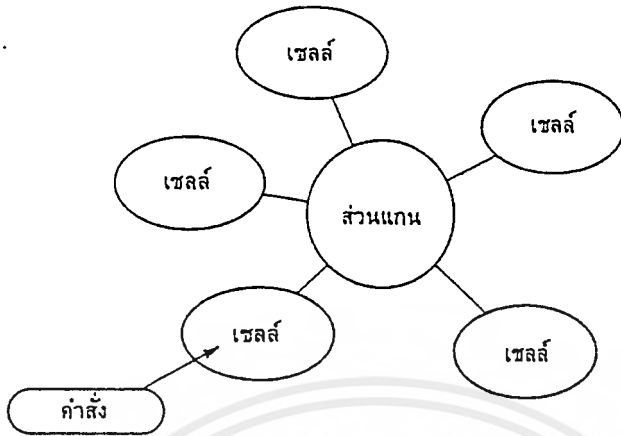
นอกจากที่ระบบปฏิบัติการยูนิกซ์จะสามารถทำงานหลายคำสั่งได้ในเวลาเดียวกันแล้ว ยังยอมให้ผู้ใช้หลายคนสามารถใช้งานเครื่องคอมพิวเตอร์หลักเครื่องเดียวกันพร้อมกันได้ การทำงานในลักษณะนี้เรียกว่า การทำงานแบบผู้ใช้หลายคน (Multiuser) ซึ่งอนุญาตให้ผู้ใช้หลายคนทำงานพร้อมกันโดยใช้เครื่องคอมพิวเตอร์เครื่องเดียวได้ จะเห็นได้อย่างชัดเจนในจุดนี้ถึงข้อได้เปรียบของระบบปฏิบัติการแบบมัลติทาสก์อย่างเช่น ในบริษัทหนึ่งมีเครื่องคอมพิวเตอร์หลักที่มีความสามารถสูง ๆ เพียงเครื่องเดียวก็สามารถให้พนักงาน 10 คนใช้ทำงานในเวลาเดียวกันได้โดยจัดเทอร์มินัล 10 เครื่องสำหรับผู้ใช้แต่ละคน แทนที่จะต้องจัดหาเครื่องคอมพิวเตอร์ที่มีความสามารถสูง 10 เครื่องสำหรับพนักงาน 10 คน ก็เพียงแต่จัดหาเทอร์มินัลซึ่งมีราคาถูกกว่า หรือหาเครื่องไมโครคอมพิวเตอร์ที่มีความเร็วสูงไม่มากนักมาทำเป็นเทอร์มินัล

ระบบปฏิบัติการยูนิกซ์สามารถทำงานหลายคำสั่งพร้อมกันได้อย่างไร คำตอบของคำถามนี้ค่อนข้างจะซับซ้อน แต่หลักการงานจะอยู่ที่การแบ่งการทำงานของคำสั่งเป็นสองส่วนคือส่วนแกน (Kernel) และเชลล์ (Shell) ส่วนแกนของระบบเป็นศูนย์รวมการประมวลผลทั้งหมดของระบบปฏิบัติการส่วนเชลล์คือส่วนย่อยของโปรแกรมต่าง ๆ หรือโปรเซสที่ทำงานบนยูนิกซ์

เชลล์บนระบบยูนิกซ์มีอยู่หลายชนิด เช่น บอร์นเชลล์ (Bourne) หรือ ซีเชลล์ (C-Shell) เมื่อผู้ใช้เรียกเข้าสู่ระบบยูนิกซ์ก็จะมีการสร้างเชลล์ใหม่ขึ้น ชนิดของเชลล์ที่สร้างขึ้นจะขึ้นอยู่กับข้อกำหนดค่าเริ่มต้นของการใช้งานของเรา หรือเมื่อมีการเรียกใช้โปรแกรมในบางครั้งก็จะเป็นการสร้างเชลล์ใหม่ขึ้นมา เชลล์ใหม่ที่สร้างขึ้นมาแต่ละครั้งก็จะเป็นอีกส่วนหนึ่งที่จะทำงานอยู่บนระบบยูนิกซ์ โดยที่ส่วนเชลล์ใหม่ที่สร้างขึ้นนี้จะสามารถทำงานได้เหมือนกับเชลล์เดิมที่เคยมีอยู่แล้ว ลักษณะการทำงานแบบเชลล์นี้ช่วยเพิ่มความสามารถของระบบยูนิกซ์ในการทำงานคำสั่งต่าง ๆ ได้อย่างดี

เมื่อเราป้อนคำสั่งให้เครื่องคอมพิวเตอร์หลักทำงาน เชลล์จะทำหน้าที่รับคำสั่งและแปลความหมายของคำสั่งนั้นให้อยู่ในรูปแบบที่ส่วนแกนเข้าใจและทำงานตามคำสั่งได้ ส่วนแกนนี้จะมีหน้าที่ในการแบ่งเวลาการทำงานของซีพียูกับเชลล์อื่นด้วย ผลลัพธ์จากการทำงานของส่วนแกนจะถูกส่งกลับไปที่ส่วนเชลล์ และแสดงผลลัพธ์ให้ผู้ใช้ทราบ ในรูปที่ 3.2 แสดงถึงรูปโครงสร้างของส่วนแกนและส่วนเชลล์

ส่วนเชลล์และส่วนแกนทำงานร่วมกันเหมือนกับลักษณะการทำงานของพนักงานเสิร์ฟ และพ่อครัวในร้านอาหาร เราซึ่งเปรียบได้กับลูกค้าจะเป็นผู้สั่ง



รูปที่ 3.2 โครงสร้างของระบบยูนิคซ์ส่วนแกนและเซลล์

3.2 คุณสมบัติของระบบยูนิคซ์

Software Tools โปรแกรมยูนิคซ์จะแบ่งตัวเองออกเป็นส่วนย่อย ๆ และสามารถใช้งานส่วนย่อยเหล่านั้นร่วมกันระหว่างหลาย ๆ โปรแกรมได้

Portability เนื่องจากยูนิคซ์สามารถนำไปใช้กับเครื่องแบบต่าง ๆ ได้มากมาย โปรแกรมที่ใช้งานบนยูนิคซ์จะสามารถนำไปใช้ได้ด้วยเช่นกัน

Flexibility ยูนิคซ์มีความยืดหยุ่นในการใช้งานสูง สามารถนำไปใช้งานกับงานเล็ก ๆ หรืองานใหญ่ ๆ ก็ได้

Power สามารถทำงานที่ซับซ้อนได้ดี และมีความสามารถในด้านต่าง ๆ มากกว่าระบบปฏิบัติการอื่น ๆ

Multi-users & Multitasking สามารถมีผู้ใช้งานได้ที่ละหลาย ๆ อย่างได้พร้อมกัน

Elegance หลักการทำงานของส่วนต่าง ๆ จะเหมือนกัน ดังนั้นเมื่อผู้ใช้ทำงานส่วนหนึ่งได้ก็จะเรียนรู้และใช้งานส่วนอื่นได้ง่าย

Network Orientation ยูนิคซ์เป็นระบบปฏิบัติการที่สนับสนุนการใช้งานเครือข่าย โดยเฉพาะเครือข่าย TCP/IP ซึ่งใช้ในระบบอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างของระบบยูนิกซ์

ในระบบยูนิกซ์นั้นประกอบด้วย 4 ส่วนสำคัญด้วยกันคือ

3.3.1 Hardware

คือตัวเครื่องคอมพิวเตอร์นั่นเอง

3.3.2 UNIX Kernel

Kernel เป็นส่วนที่สำคัญที่สุดของยูนิกซ์คือเป็นส่วนหลักที่เก็บตัวโปรแกรมของยูนิกซ์ และจะกำหนดความสามารถทั้งหมดของยูนิกซ์นั้น ๆ เมื่อเราเปิดเครื่องสิ่งแรกที่ทำก็คือ อ่าน Kernel เข้าสู่หน่วยความจำ แล้วยูนิกซ์ก็จะเริ่มทำงานได้

3.3.3 Shell

Shell เป็นตัวกลางที่ติดต่อระหว่างผู้ใช้กับ Kernel และเป็นตัวกำหนดหน้าตาของยูนิกซ์ที่จะปรากฏแก่ผู้ใช้ Shell ที่ใช้กันทั่ว ๆ ไปในยูนิกซ์ได้แก่ Bourne Shell, Korn Shell และ C Shell

3.3.4 User Program

คือโปรแกรมที่ผู้ใช้เรียกใช้งานนั่นเอง

3.4 โครงสร้างไดเรกทอรีในยูนิกซ์

/ Root Directory เป็นเป็นจุดเริ่มต้นของ โครงสร้างทั้งหมด

/bin จะเก็บคำสั่งใช้งานทั่ว ๆ ไปไว้

/lib เก็บ run-time library

/etc เก็บ config และ ตารางที่ใช้กำหนดค่าต่าง ๆ ของระบบ

/dev จะเก็บ Device driver

/tmp เก็บข้อมูลชั่วคราว และจะถูกลบทุกครั้งที่เปิดเครื่อง

/var จะเก็บบันทึกต่าง ๆ ของระบบไว้ตรวจสอบ

/home มักจะเก็บ ไดเรกทอรีเริ่มต้นของผู้ใช้ต่าง ๆ ในระบบ

/user จะเก็บ โปรแกรมและคำสั่งต่าง ๆ ที่มักจะต้องใช้อยู่ร่วมกันระหว่างผู้ใช้หลายคน

/usr/local มักจะใช้เก็บโปรแกรมที่นำมาใส่เพิ่มเติมให้กับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

HTML

HTML ย่อมาจากคำว่า Hyper Text Markup Language ซึ่งเป็นภาษาหลักสำหรับการสร้างโฮมเพจ เพิ่มเอกสาร HTML ที่สร้างขึ้นจะนำไปแสดงผลได้ด้วยโปรแกรม Web Browser ลักษณะของ Hypertext ที่เราพบเห็นกันอยู่เสมอก็คือการใช้ระบบ Help บนวินโดวส์

WWW แบ่งออกเป็นสองส่วนคือ ส่วนที่เป็น Client และส่วนที่เป็น Server เหมือนกับที่มีแบ่งกันในระบบเครือข่ายทั่วไป ทั้งสองส่วนจะถูกเชื่อมโยงถึงกันผ่านทางอินเทอร์เน็ต โดยมี HTML เป็นส่วนฐานข้อมูลสำคัญ เมื่อ Web Browser ส่งข้อความร้องขอข้อมูลที่อยู่ในรูปของไฟล์ HTML จากเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ผ่าน โมเด็มหรืออุปกรณ์สื่อสารข้อมูลอื่นไปยังศูนย์บริการอินเทอร์เน็ตตามโปรโตคอลที่กำหนดไว้ผ่านทาง URLs และเมื่อข้อมูลเดินทางมาถึง Web Server ศูนย์บริการปลายทางที่ผู้ใช้ต้องการ ณ ที่นี้เครื่อง Web Server อื่นอีก หลังจากจบสิ้นกระบวนการทั้งหมดแล้วจะทำการจัดส่งข้อมูลคำตอบย้อนกลับมายังเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ โปรแกรม Web Browser ที่เครื่องคอมพิวเตอร์ของเราก็จะแปรงสัญญาณคำสั่งและแสดงผลเป็นข้อความ รูปภาพ เสียง

HTML นอกจากใช้ในการสร้างฐานข้อมูลบนอินเทอร์เน็ตแล้ว ยังมีความสามารถทางด้านการเชื่อมโยงข้อมูลอื่นบนอินเทอร์เน็ต ไม่ว่าจะเป็น E-mail, FTP, Gopher, Telnet หรือ News ทำให้เราสามารถเรียกใช้บริการเหล่านี้ได้ทันที

4.1 โครงสร้างพื้นฐานของ HTML

โครงสร้างของ HTML ประกอบไปด้วยส่วนของคำสั่ง 2 ส่วน ได้แก่ ส่วนที่เป็นส่วนหัว (Head) และเนื้อหา (Body) โดยมีรูปแบบของภาษาดังนี้

```
<HTML>
<HEAD>
<TITLE>ชื่อโปรแกรมหรือข้อมูลที่ต้องการแสดง</TITLE>
</HEAD>
<BODY>
คำสั่งหรือข้อความที่ต้องการให้แสดง
</BODY>
</HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในความง่ายของภาษา HTML นั้นเป็นเพราะภาษานี้ไม่มีโครงสร้างใด ๆ มากำหนดนอกจากโครงสร้างพื้นฐานเท่านั้นหรือแม้แต่จะไม่มีโครงสร้างพื้นฐานอยู่ โปรแกรมที่เขียนขึ้นมาก็สามารถทำงานได้เสมือนมีโครงสร้าง ทั้งนี้เป็นเพราะว่าตัวโปรแกรม Web Browser จะมองเห็นทุกสิ่งทุกอย่างในโปรแกรม HTML เป็นส่วนเนื้อหาทั้งสิ้น ยกเว้นในส่วนหัวที่ต้องมีการกำหนดแยกออกไปให้เห็นชัดเท่านั้นจะเขียนคำสั่งหรือข้อความที่ต้องการให้แสดงอย่างไรก็ได้ เป็นเสมือนกับการพิมพ์งานเอกสารธรรมดาทั่วไป เพียงแต่ที่ตำแหน่งใดมีการกำหนดรหัสพิเศษขึ้นมา เว็บเบราว์เซอร์ก็จะแสดงผลออกมาตามที่ถูกกำหนดโดยใช้คำสั่งให้ตรงกับที่กำหนดเท่านั้น

4.2 URLs

URLs ย่อมาจากคำว่า Uniform Resource Location เป็นชื่อที่ใช้เรียกตำแหน่งข้อมูลบนอินเทอร์เน็ต จะทำงานจากไฮเปอร์เท็กซ์ที่เชื่อมต่อกับข้อมูลภายในเว็บเพจไปยังโปรแกรมประยุกต์อื่นตามโปรโตคอลซึ่งกำหนดโดยผู้ใช้ มีข้อกำหนดรูปแบบของการเชื่อมต่อดังนี้

Protocol://hostname/filename

ตัวอย่างที่เราใช้อยู่เสมอเวลาเรียกใช้งานอินเทอร์เน็ต เช่น <http://www.kmit.ac.th/~9013160> ที่ผ่านมามาเราได้ใช้รูปแบบการเชื่อมโยงแบบนี้โดยตลอด

URLs แบ่งรูปแบบของการเชื่อมต่อเป็น 2 แบบ ดังนี้

-โปรโตคอลที่ต่อท้ายด้วย // ประกอบไปด้วย

- ftp:// ทำหน้าที่โอนถ่ายไฟล์หรือดาวน์โหลดจาก FTP Server
- http:// ทำหน้าที่เรียกไฮเปอร์เท็กซ์ขึ้นมาแสดงผลจาก WWW Server
- telnet:// ทำหน้าที่เป็นรีโมทคอนโทรล
- file:// ทำหน้าที่ค้นหาและแสดงผลข้อมูลจากไฟล์ที่กำหนดแบบ local
- gopher:// ทำหน้าที่เรียกไฟล์ขึ้นมาใช้งานจาก Gopher Server

-โปรโตคอลที่ไม่ได้ต่อท้ายด้วย // ประกอบไปด้วย

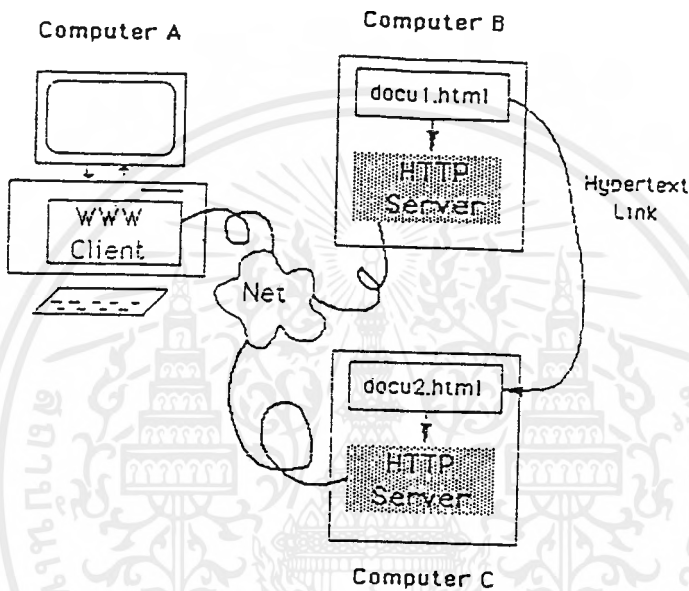
- mailto ทำหน้าที่สื่อสารข้อมูลโดยจดหมาย
- news ทำหน้าที่สื่อสารข้อมูลแบบจดหมายข่าว

ปกติการเรียกใช้งาน URLs จะถูกกำหนดให้เป็นส่วนหนึ่งของการเชื่อมโยงข้อมูล เมื่อต้องการใช้งานโปรโตคอลเพียงแต่กำหนดชนิดของโปรโตคอลนั้นเข้ากับ LINKS เท่านั้นก็ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การสร้างแบบสอบถาม

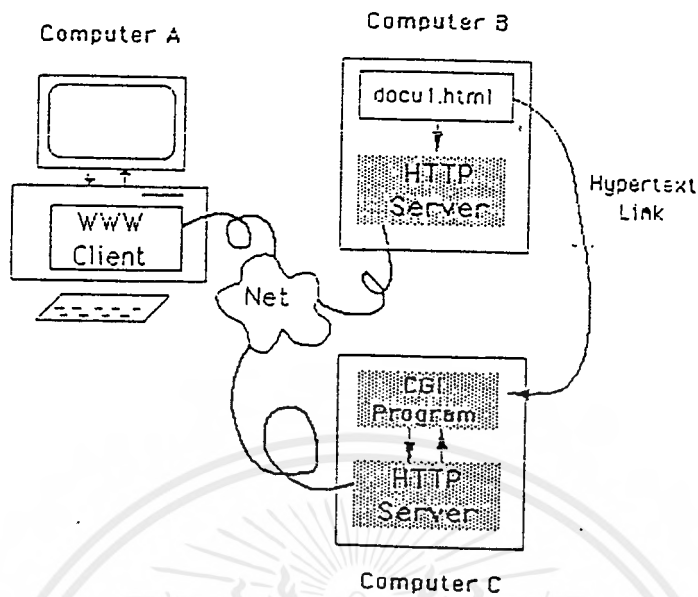
แต่เดิม HTML จะมีลักษณะของการสื่อสารข้อมูลแบบทางเดียวคือใช้อ่านข้อมูลได้เพียงอย่างเดียว ด้วยความสามารถของ HTML รุ่นใหม่ทำให้เราสามารถสื่อสารกับผู้อื่นได้ในลักษณะสองทาง นั่นคือนอกจากอ่านข้อมูลได้เพียงอย่างเดียวแล้ว เรายังส่งข้อมูลกลับไปยังศูนย์บริการได้อีกด้วย



รูปที่ 4.1 การสื่อสารข้อมูลแบบสื่อทางเดียว

เมื่อเครื่องคอมพิวเตอร์ของผู้ใช้ A ส่งคำร้องไปยังศูนย์บริการ B ได้เพิ่มเอกสาร docu1.html และจากการเชื่อมโยงไปยังเพิ่มเอกสาร docu2.html ที่ศูนย์บริการ C นี้เป็นลักษณะของ HTML ในรุ่นแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การสื่อสารข้อมูลแบบสองทาง

ในรุ่นต่อมาได้มีการพัฒนาภาษา HTML ให้มีความสามารถเพิ่มมากยิ่งขึ้น การสื่อสารระหว่างเครื่องคอมพิวเตอร์จึงมีลักษณะ 2 ทาง สามารถโต้ตอบกันได้ จากที่มีการโต้ตอบไปมา จะมีโปรแกรมตัวกลางทำหน้าที่เชื่อมโยงข้อมูลระหว่างเครื่องคอมพิวเตอร์ผู้ใช้กับเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ที่เรียกว่า CGI

Common Gateway Interface หรือ CGI เป็นตัวกลางที่ทำหน้าที่เชื่อมโยง WWW Server เข้ากับข้อมูลเพื่อนำไปใช้งานต่าง ๆ ต่อไป โดยสามารถรวบรวมข้อมูลที่ผู้ใช้ป้อนเข้ามาประมวลผลส่งกลับไปยังผู้ใช้ก็ได้ โปรแกรม CGI นี้สามารถเขียนเองได้ด้วยภาษา C++, Perl, Pascal หรือ JAVA จะทำให้ WWW Server แต่ละแห่งสามารถสร้างฐานข้อมูลของตัวเองได้

ในการสร้างฟอร์มนั้นจะประกอบด้วยข้อมูล 2 ส่วน ส่วนแรกเรียกว่าส่วนเชื่อมต่อ (Interface) และส่วนควบคุม (CGI)

ส่วนเชื่อมต่อนั้นเราจะสร้างด้วยภาษา HTML ส่วน CGI จะสร้างโดยภาษา Perl ส่วนเชื่อมต่อนั้นจะสร้างเก็บไว้ในแฟ้มเอกสาร .html ก็ได้ แต่สำหรับส่วน CGI ต้องเก็บไว้ที่ WWW Server (Server Side CGI) เพราะว่าจะขณะนี้ยังไม่มีโปรแกรม Web Browser ตัวใดสนับสนุน Client Side CGI ที่จะทำให้เราเขียน CGI ผสมกับแฟ้มเอกสาร .html ได้

รูปแบบของการใช้ Perl เขียน CGI ใน WWW จะได้นำเสนอในบทที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

Perl และ ASP

Perl ย่อมาจาก Practical Extraction and Reporting Language เป็นภาษาหนึ่งที่ถูกบัญญัติขึ้นโดยนาย Larry Wall ในปี 1986 ซึ่งเริ่มใช้สำหรับเขียนโปรแกรมในระบบยูนิกซ์ ก่อนที่จะถูกพัฒนาให้ใช้กับระบบปฏิบัติการอื่น ๆ ได้ เช่น WindowsNT Perl เป็นโปรแกรมภาษาที่เขียนอยู่ในรูปของสคริปต์ (Scripts Language) และเก็บอยู่ในรูปของเท็กซ์ไฟล์ จึงทำให้การแก้ไขและการทำความเข้าใจได้ง่ายกว่า ภายในไฟล์โปรแกรมนั้นจะประกอบไปด้วยคำสั่งของ Perl ซึ่งลักษณะของคำสั่งและรูปแบบการเขียนนั้น จะคล้าย ๆ กับภาษาซี แต่มีรูปแบบที่ง่ายกว่าและมีความยืดหยุ่นภายในตัวภาษาเอง Perl ถูกออกแบบมาให้รองรับกับการเขียนโปรแกรมบนเว็บ (Web programming) ซึ่งใช้สำหรับสร้าง CGI (Common Gateway Interface) สำหรับจัดการข้อมูลให้กับ Web เช่น การกรอกแบบสอบถาม (Form) หรือ การสร้างเว็บเพจใหม่ (Dynamic Webpage) เป็นต้น เหตุที่ Perl เป็นภาษาที่ได้รับความนิยมก็เพราะเหตุผลต่อไปนี้

1. Perl ง่ายต่อการศึกษา เพราะโครงสร้างของภาษาไม่ซับซ้อน
2. ตัวภาษามีความยืดหยุ่นสูง (ที่มีลักษณะเหมือนกันภาษาซี)
3. การเขียน Perl จะเขียนอยู่ในรูปของเท็กซ์ไฟล์ โดยใช้โปรแกรม NotePad (บน Windows)

หรือใช้โปรแกรม Vi

4. การประมวลผลหรือรันโปรแกรมที่เขียนด้วย Perl ไม่จำเป็นต้อง Compile และ Link ให้เป็น Machine-Code เช่นเดียวกับภาษาซี เพราะการรันโปรแกรมจะต้องใช้ตัวแปลภาษาของ Perl เป็นตัวประมวลผล

5. ที่สำคัญ... Perl เป็นฟรีแวร์ (Freeware)

5.1 ตัวแปลภาษาของ Perl

ตัวแปลภาษาของ Perl นั้น เป็น อินเตอร์พรีเตอร์ หรือ คอมไพเลอร์ เมื่อเราศึกษาภาษาใด เราก็ควรรู้ว่าภาษานั้น ใช้ตัวแปลภาษาแบบใด ซึ่งโดยทั่ว ๆ ไปแล้ว เราจะแบ่งออกเป็น 2 ประเภท นั่นก็คือ อินเตอร์พรีเตอร์ (Interpreter) และ คอมไพเลอร์ (Compiler) ความแตกต่างของตัวแปลภาษาทั้งสองประเภท มีดังนี้.. อินเตอร์พรีเตอร์ การทำงานของตัวแปลภาษาแบบนี้ จะทำอ่านซอร์สโค้ดโปรแกรมที่ละบรรทัดการประมวลผลเป็นบรรทัดไป หากเกิดข้อผิดพลาด ตัวแปลภาษาจะหยุดประมวลผลและแสดงข้อผิดพลาดทันที ภาษาที่ใช้ตัวแปลแบบนี้ก็เช่น ภาษา Basic

คอมไพเลอร์ การทำงานของตัวแปลภาษาแบบคอมไพเลอร์ จะเริ่มประมวลผลโดยอ่าน และตรวจเช็คข้อผิดพลาดภายในซอร์สโค้ด โปรแกรมทั้งหมดและทำการแปลงให้อยู่ในรูปของไฟล์เอ็กซิกิวต์ (Execute) จากนั้นจึงทำการรัน ตัวอย่างภาษาที่ใช้ตัวแปลแบบนี้ก็คือ ภาษาซี

สำหรับตัวแปลภาษาของ Perl นั้น อยู่ระหว่างความเป็น อินเตอร์พรีเตอร์ และ คอมไพเลอร์ เพราะเมื่อคุณรัน โปรแกรมหรือสคริปต์ ตัวแปลภาษาของ Perl จะคอมไพล์ซอร์สโค้ดโปรแกรมให้อยู่ในรูปของไฟล์เอ็กซิกิวต์และเก็บไว้ในหน่วยความจำ จากนั้นจึงทำการรัน โปรแกรม ทำให้ประมวลผลนั้นมีความรวดเร็วยิ่งขึ้น เพราะไม่มีการเขียนไฟล์เอ็กซิกิวต์ลงบนดิสก์ เหมือนกับคอมไพเลอร์ภาษาซี

5.2 มองหา Perl สำหรับระบบ

ก่อนที่คุณจะติดตั้งตัวแปลภาษาของ Perl คุณจะต้องดูเสียก่อนว่า คุณใช้ระบบปฏิบัติการใด เพื่อที่คุณจะได้ติดตั้งให้เหมาะสมกับระบบของคุณ ถ้าคุณใช้ระบบปฏิบัติการยูนิกซ์ (เช่น Linux) คุณก็จะต้องดาวน์โหลดตัวแปลภาษา Perl สำหรับยูนิกซ์มา จากนั้นทำการติดตั้งลงในระบบของคุณ ซึ่งการติดตั้ง จะทำให้เสร็จเรียบร้อยเอง ถ้าคุณใช้ระบบปฏิบัติการวินโดวส์เอ็นที (Windows NT) คุณก็จะต้องดาวน์โหลดตัวแปลภาษา Perl สำหรับวินโดวส์เอ็นที (Perl for Win32) มาติดตั้ง และถ้าคุณใช้วินโดวส์ 95 แต่คุณต้องการศึกษาการเขียน Perl ไว้ เพื่อต่อไปคุณอาจจะต้องใช้มัน หรือ อยากรู้เพื่อทำความเข้าใจกับรูปแบบการเขียนและโครงสร้างของภาษา คุณก็สามารถทำได้ โดยใช้โปรแกรม

Win Perl หรือ โปรแกรม Perl for Win32 ติดตั้งลงบนวินโดวส์ 95 เพื่อทดลองฝึกเขียนและทำความเข้าใจกับ Perl และ CGI ได้

5.3 ติดตั้ง Perl สำหรับยูนิกซ์

ในการติดตั้ง Perl สำหรับยูนิกซ์ จะขอยกตัวอย่างของ Linux ในตอนที่ติดตั้ง Linux ถ้าคุณติดตั้งโปรแกรมในหมวด D (D Series) คุณก็จะได้ ติดตั้งตัวแปลภาษาของ Perl ด้วย ถ้าคุณไม่แน่ใจให้คุณใช้คำสั่งต่อไปนี้ที่ยูนิกซ์เชลล์ของคุณ

```
$ whereis perl
```

ถ้าในระบบยูนิกซ์ของคุณมีการติดตั้งตัวแปลภาษา Perl เอาไว้แล้ว คุณจะได้รับรายงานมาว่า /usr/bin/perl ซึ่ง /usr/bin/perl นั้นก็คือพาท ไคเร็คทอรีของตัวแปลภาษานั้นเอง และถ้ายูนิกซ์รายงาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น การนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

คุณว่า no perl หรือหาไม่พบในพาธใดเรีคทอรี ก็ให้สอบถามไปยังผู้ดูแลระบบ (Administrator) หรือทำการติดตั้งถ้าเป็นระบบของคุณเอง (Linux เรียก setup หรือ unzip)

5.4 การสร้างและรันโปรแกรม

การเขียนโปรแกรม Perl มีวิธีง่าย ๆ ก็คือ คุณใช้อิติดเตอร์ใด ๆ ก็ได้ในระบบสร้างเท็กซ์ไฟล์ ขึ้นมาไฟล์หนึ่ง และทำการเซฟโดยกำหนดนามสกุลให้เป็น .pl จากนั้นทำการรัน โดยตัวแปลภาษาของ Perl

การรันโปรแกรม Perl บนยูนิกซ์

สำหรับระบบยูนิกซ์ ถ้าคุณใช้ยูนิกซ์ คุณสามารถสร้างได้โดยใช้โปรแกรม vi อิติดเตอร์ สร้างไฟล์ใหม่ที่มีนามสกุล .pl ขึ้นมา โดยพิมพ์คำสั่งต่อไปนี้ที่ยูนิกซ์เชลล์

```
$ vi easy.pl
```

หากระบบยูนิกซ์ของคุณไม่มี vi อิติดเตอร์ คุณก็สามารถใช้อิติดเตอร์ตัวอื่นหรือใช้คำสั่ง cat >easy.pl ก็ได้ จากนั้นให้คุณพิมพ์โปรแกรมด้วยภาษา Perl ลงไป สิ่งหนึ่งที่คุณต้องจำเมื่อคุณจะเขียนโปรแกรม Perl บนยูนิกซ์ก็คือ บรรทัดบนสุดของไฟล์คุณจะต้องเขียนดังนี้

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

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

บรรทัดบนสุดนี้เป็นการบอกถึงพาธของตัวแปลภาษาของ Perl ซึ่งส่วนมากแล้ว ตัวแปลภาษาของ Perl จะถูกเก็บอยู่ใน Directory ในข้างต้น แต่ถ้าคุณไม่แน่ใจ ก็ให้คุณพิมพ์คำสั่ง whereis perl เพื่อตรวจสอบดูอีกครั้ง เมื่อคุณพิมพ์โปรแกรมเสร็จแล้ว ให้ทำการเซฟโดยกดปุ่ม ESC + ":" + wq และ Enter เพื่อทำการเซฟและออกจาก vi จากนั้นทำการรันโปรแกรมโดยพิมพ์ดังนี้

```
$ perl easy.pl
```

ถ้าโปรแกรมมีข้อผิดพลาด ตัวแปลภาษาจะแสดงข้อความบอกถึงจุดที่ผิดพลาดมาให้เพื่อแก้ไข หรือคุณอาจจะใช้วิธีการรัน โดยการแปลงไฟล์ให้อยู่ในโหมด Execute เสียก่อน โดยพิมพ์คำสั่งต่อไปนี้

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

เมื่อคุณทำการรัน โปรแกรม คุณก็พิมพ์แค่ easy เลข ๆ โปรแกรมก็จะรันโดยอัตโนมัติ

การใช้งาน Perl

เนื่องจาก Perl ก็คล้ายโปรแกรมบน DOS โปรแกรมหนึ่ง เวลาที่ทำงานจะทำงานอยู่เบื้องหลัง ถ้าไม่มีการเรียกใช้ ก็จะไม่ทำงานเราสามารถเรียกใช้ใน Perl ด้วย Command line แบบคำสั่ง DOS ได้ การใช้ภาษา Perl เขียน โปรแกรม CGI ที่สามารถแสดงผลออกมาที่ Browser นั้นสามารถทำได้โดยการที่เราพิมพ์คำสั่งดังต่อไปนี้

```
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 ถ้าไม่ใส่ก็ไม่เป็นไรแต่ผลของโปรแกรมที่ส่งมาให้ Browser เมื่อ view source ดูเห็นติดกันหมดเป็นบรรทัดเดียว เมื่อเสร็จแล้วให้ save as เป็นไฟล์ เช่น test2.pl โดยไฟล์นี้ให้เก็บไว้ใน directory ที่กำหนดไว้ส่วนใหญ่ใน Server จะเก็บไว้ใน Cgi-Bin หรือ Directory อื่นที่สามารถ execute ไฟล์ได้ จากนั้นให้คุณพิมพ์เรียกในบราวเซอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

`http://you-server-name/cgi-bin/test2.pl`

คือชื่อ server ที่คุณตั้งไว้ ถ้าไม่รู้ให้แทนส่วนนี้ด้วยเลขเช่น 161.246.29.11 ซึ่งส่วนมากจะใช้ได้ และถ้าคุณไม่ได้ใช้ Cgi-Bin ให้แทนด้วยอันอื่นที่คุณใช้เช่น script เป็นต้น การเรียกใช้ test2.pl หรือไฟล์ .pl อื่นๆจะต้องเรียกจาก server เท่านั้น โดยการพิมพ์เรียกลักษณะดังตัวอย่าง การเปิดแบบเปิดไฟล์ มันจะไม่ทำงาน เมื่อคุณเรียกโดยผ่าน Browser แล้ว จะปรากฏเป็นคำว่า hello my Perl ขึ้นมาบนหน้าจอ จะเห็นว่าในขณะนี้เราสามารถให้ Perl เขียนโปรแกรมให้แสดงข้อความบนหน้าจอ Browser ได้แล้ว ซึ่งเป็นตัวแปรภาษา Perl ให้ทำการแปรคำสั่งในไฟล์ test2.pl เมื่อ perl.exe แปรเสร็จแล้ว จะส่งผลออกมาให้ Web Server และทำการปิดตัวเองลง เมื่อ Web Server ได้รับผลมาแล้ว จะส่งผลนั้นกลับไปให้บราวเซอร์ที่เรียกในรูปของ html ไฟล์ เนื่องจากใน test2.pl มีการกำหนดคำสั่งด้วย `print "Content-type:text/html\n\n";` เพราะฉะนั้นการที่จะเขียน CGI ด้วย Perl ให้แสดงผลที่หน้าจอ Browser จะต้องใส่คำสั่ง `print "Content-type:text/html\n\n";` ลงไปด้วย

หมายเหตุ สำหรับผู้ใช้ UNIX ต้องใส่บันทึกแรกของทุกโปรแกรมด้วย `#!/usr/bin/perl` หรือที่คล้ายๆ แบบนี้ซึ่งบันทึกนี้คือคำสั่งที่บอกให้ Server ไปเรียกโปรแกรมสำหรับแปลภาษา Perl ให้ทำงาน ง่ายๆ ก็คือที่อยู่ของตัวโปรแกรม perl.exe นั่นเอง ถ้าไม่รู้ให้ถามผู้ดูแล Server หรือ ใช้คำสั่ง `whereis perl` สำหรับผู้ใช้ Windows 95/98/NT ไม่จำเป็นต้องใส่ อย่างนี้
95/98/NT ฉลาดกว่า UNIX หรือเปล่า ?

```
print "Content-type:text/html\n\n";
```

```
print "<html><head><title>เขียน CGI ด้วย Perl</title></head>";
```

```
print "<body bgcolor=\"black\" text=\"white\">";
```

```
print "<center>";
```

```
print "ข้อความที่ต้องการแสดงบนหน้าจอ Browser ถ้ามีเครื่องหมาย \@ หรือเครื่องหมาย \"<br>";
```

```
print "คุณจะต้องใช้เครื่องหมาย \ ใส่ไว้ข้างหน้าเครื่องหมายดังกล่าวด้วย ถ้าไม่ใส่จะเกิด error ขึ้น<br>";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
print "ในการใช้ภาษา Perl ถ้าคุณต้องการใส่ comment ให้ใช้เครื่องหมาย # ใส่ไว้ข้างหน้าบรรทัดที่  
<br>";
```

```
print "ต้องการ comment ดูตัวอย่างที่กล่าวมาแล้วทั้งหมดได้ใน source code ของไฟล์นี้ (test3.pl)  
<br>";
```

```
print "</center>";
```

```
print "</body></html>";
```

จากตัวอย่างข้างบนเมื่อพิมพ์เสร็จแล้ว save เป็นไฟล์ test3.pl จากนั้นลองเรียกดูด้วย Browser เป็นไปตามที่บอกไว้ในไฟล์ ถึงตรงนี้ถ้าคุณเกิดความคิดขึ้นมาอีกว่า ไม่ต้องใช้คำสั่ง print กันเลยหรือหรือ? ถ้ามีไฟล์ใหญ่ๆถ้าเกิดมีข้อความมากๆ คำตอบคือ ใช่ แต่เราสามารถใส่คำสั่ง print โดยวิธีลัดก็ได้ดังตัวอย่างที่จะให้ผลเช่นเดียวกัน

```
print "Content-type:text/html\n\n";
```

```
print <<myHTML;
```

```
<html><head><title>เขียน CGI ด้วย Perl</title></head>
```

```
<body bgcolor=\"black\" text=\"white\">
```

```
<center>
```

```
ข้อความที่ต้องการแสดงบนหน้าจอ Browser ถ้ามีเครื่องหมาย \@ หรือเครื่องหมาย \"<br>
```

```
คุณจะต้องใช้เครื่องหมาย \ ใส่ไว้ข้างหน้าเครื่องหมายดังกล่าวด้วย ถ้าไม่ใส่จะเกิด error ขึ้น<br>
```

```
ในการใช้ภาษา Perl ถ้าคุณต้องการใส่ comment ให้ใช้เครื่องหมาย # ใส่ไว้ข้างหน้าบรรทัดที่<br>
```

```
</center>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
</body></html>
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MyHTML

จะเห็นว่าแบบนี้จะใช้คำสั่ง print น้อยลง ส่วนที่เป็นคำว่า myHTML จะเปลี่ยนเป็นชื่ออื่นที่ต้องการก็ได้เช่น ENDHTML เป็นต้น จาก Code จะเห็นว่าต้องใส่ myHTML สองที่คือเริ่มต้นและสิ้นสุด

ASP

ASP ย่อมาจาก Active Server Pages อ่านออกเสียงว่า "เอ เอส พี"เจ้าของหรือผู้คิดค้นคือบริษัท Microsoft ASP เป็นโปรแกรมคอมพิวเตอร์ชนิดหนึ่งที่เจ้าของได้ให้นิยามไว้คือ "Server side scripting" ซึ่งหมายถึงภาษาทางโปรแกรมที่ทำงานในฝั่งของเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Web Server ที่ให้บริการเอกสารหรือสื่อต่างๆ ในอินเทอร์เน็ตหรืออินทราเน็ต ASP ใช้สำหรับสร้างงาน (Application) ขึ้นสูง ในอินเทอร์เน็ต-อินทราเน็ต ASP ได้เสริมการทำงานที่ไฟล์ html ธรรมดาไม่สามารถทำได้ หรือต้องการให้งานต่างๆ เป็นไปอย่างอัตโนมัติโดยผู้ใช้ไม่ต้องทำการปรับปรุงข้อมูลเอง เหมาะสำหรับผู้ที่มีอาชีพหรือผู้ที่สนใจอย่างจริงจัง สำหรับผู้ใช้ที่ไม่ต้องการเว็บไซต์ที่มีลักษณะดังกล่าวข้างต้น หรือมีข้อมูลน้อยเพียงไม่กี่หน้า นานๆ จะปรับปรุงข้อมูลสักครั้ง ก็ไม่จำเป็นต้องใช้ก็ได้ สำหรับงานที่ต้องการให้เป็นอัตโนมัติเช่น Guest Book, Counter, สถิติ, ห้องสนทนา ก็สามารถสมัครใช้บริการหรือใช้ cgi สำเร็จรูปได้ซึ่งมีหลายเว็บไซต์ให้บริการโดยไม่คิดค่าใช้จ่าย สร้างงานจาก ASP ก็ไม่ต้องลงทุนอะไรเพราะโปรแกรมต่างๆ เป็นโปรแกรมที่ใช้ได้โดยไม่เสียค่าใช้จ่าย

ความต้องการพื้นฐานในการใช้ ASP ในส่วนของนักพัฒนาโปรแกรม/เว็บไซต์

- เครื่องคอมพิวเตอร์ PC ความเร็ว CPU 486 ขึ้นไป RAM 16 MB ขึ้นไป ถ้ายิ่งสูงจะยิ่งดีแนะนำที่ Pentium 150 MHz ขึ้นไป RAM 32 MB
- ระบบปฏิบัติการ(OS) Windows 95/98 หรือ Windows NT 4.0 (server/work-station)
- สำหรับ Windows 95/98 ต้องมี

1. บราวเซอร์ IE 3-5

2. โปรแกรมจำลองเครื่องคอมพิวเตอร์ PC ให้เป็น web server เช่น Personal Web Server(PWS) ของ Microsoft เป็นต้น(มีอยู่ใน Windows 98)

3. โปรแกรม ASP (จะมี ASP รวมอยู่ใน PWS เวอร์ชัน 4 ขึ้นไป) แต่ถ้าลง โปรแกรม web server อื่นๆ หรือ PWS ต่ำกว่าเวอร์ชัน 4 จะต้องลง ASP ด้วย หาได้จากชุด MS Visual Studio สำหรับ Windows NT 4.0 ต้องมี IIS 3.0(Internet Information Server) ขึ้นไป และ ASP พร้อมอยู่แล้วหรือถ้ายังให้ลงโปรแกรมเพิ่มเติม

- ความรู้ในการใช้โปรแกรม Windows 95/98+PWS หรือ NT+IIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความรู้ในการใช้ HTML ในระดับ Source Code รวมถึงการใช้โปรแกรม Text Editor ใดๆเช่น Notepad เป็นต้นสำหรับผู้ที่ใช้โปรแกรมสำเร็จรูปในการสร้างไฟล์ html ควรทราบถึงวิธีการแทรก script ใน html source code ด้วย
- ความรู้ในการใช้ภาษา Java script, VB script, Perl script ใดๆอย่างหนึ่ง
- ความรู้ในการใช้เครือข่ายอินเทอร์เน็ต โดยจะต้องใช้โปรแกรม FTP เป็นในระดับหนึ่ง (upload/download/create/editdirectory)
- ถ้าต้องการใช้ฐานข้อมูลในอินเทอร์เน็ตด้วยก็ควรทราบถึงหลักการฐานข้อมูล และการใช้โปรแกรมฐานข้อมูลเช่น MS Access, FoxPro, Dbase หรือ Flat file database ตลอดจน ภาษา SQL และ ODBC

5.5 หลักการทำงานของ ASP

เริ่มจากผู้ใช้ ASP สร้างไฟล์ที่มีนามสกุลเป็น .asp ขึ้นมา จากนั้นนำไฟล์นั้นไปไว้ในเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น web server ที่ติดตั้งโปรแกรม ASP ไว้ และเชื่อมต่ออยู่กับเครือข่ายอินเทอร์เน็ต จากนั้นเมื่อมีผู้ใช้รายใดเรียกใช้ไฟล์นั้นผ่านโปรแกรมบราวเซอร์ (IE, Netscape ..) โปรแกรม ASP ใน web server จะเรียกไฟล์นั้นขึ้นมาอ่านแล้วทำตามคำสั่งต่างๆที่ผู้สร้างไฟล์นั้นได้กำหนดขึ้น จึงส่งผลที่ได้กลับไปให้ผู้เรียกใช้โดยแสดงผลที่โปรแกรม Browser ของผู้เรียก ซึ่งขั้นตอนข้างต้นเป็นหลักการทำงานโดยทั่วไปของ ASP ไฟล์ .asp ก็คือไฟล์ .htm หรือ .html ที่จับมาเปลี่ยนนามสกุลไฟล์เสียใหม่ให้เป็น .asp นั่นเอง ซึ่งการที่เปลี่ยนไปแต่นามสกุลก็ถือว่าเป็น ASP แล้ว แต่คงไม่ง่ายๆแค่นี้ เนื่องจากผู้ใช้สามารถแทรกภาษาการโปรแกรมไว้ในไฟล์นี้ได้ เหมือนๆกับการแทรก Java Script หรือ VB Script กำหนดให้เกิดงานที่ต้องการได้ตามขอบเขตที่ ASP สามารถทำได้จะแตกต่างกันก็คือ การแทรก Script ในไฟล์ .htm/.html Script จะทำงานที่ฝั่งของผู้เรียกใช้ไฟล์ แต่การแทรกในไฟล์ .asp สามารถกำหนดให้ทำงานฝั่ง server หรือฝั่งของผู้เรียกใช้ (client) ก็ได้ perl cgi ASP แต่ละไฟล์ก็เปรียบคล้ายๆกับไฟล์ .cgi/.pl แต่มนามสกุลต่างกันและมีตัวแปลภาษาต่างกัน กล่าวคือ .cgi/.pl จะใช้ perl ในการแปล แต่ .asp จะใช้ ASP เป็นตัวแปล script ในไฟล์ .asp และสิ่งที่เป็นความจริงส่วนใหญ่อีกก็คือการสร้าง ASP จะง่ายกว่าการสร้าง cgi ทั่วไปอย่างมาก

5.6 ขั้นตอนในการสร้าง ASP

1. สร้างไฟล์ asp ไฟล์แรก และทดลองใช้ขั้นตอนที่ 1 เปิดโปรแกรม notepad หรือ text editor อื่นๆขึ้นมาแล้วให้ สร้างไฟล์ขึ้นมาเช่น

```
[ /asp/hello.asp ]
```

```
<HTML>
```

<BODY>เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<% For i = 3 To 7 %>
<FONT SIZE=<% = i %>>
Hello World!</FONT><BR>
<% Next %>
</font> ตัวอย่างนี้มาจากไฟล์ตัวอย่างของ ASP</a>
</BODY>
</HTML>

```

ขั้นตอนที่ 2 เปิดโปรแกรมบราวเซอร์และพิมพ์เรียกที่เอาไฟล์ ASP ไปไว้ สรุป ผลที่ได้จะเห็นว่า มีคำว่า Hello World! อยู่ 5 ตัวและมีขนาดจากเล็กสุดไปใหญ่สุดซึ่งถ้าใช้ html ธรรมดาให้เกิดผลแบบนี้จะต้องเขียน code กำหนดขนาด font ถึง 5 ชุด แต่ถ้าใช้ asp code ก็จะใช้เพียงไม่กี่บรรทัดเท่านั้น จากนี้ให้ลอง view source code ของผลที่ได้ดู สิ่งที่คุณจะพบคือ html code ธรรมดา ไม่มี vb script อยู่ในนั้น ซึ่งแสดงให้เห็นว่าไฟล์ hello.asp ได้ถูกอ่านแล้วประมวลผลออกมาตามรูปแบบที่กำหนดและไม่ว่าจะเรียกใช้กี่ครั้งผลก็จะเหมือนเดิม ตัวไฟล์ hello.asp จะไม่เปลี่ยนไป และผู้เรียกโดย Browser ก็ไม่สามารถดู code ตัวจริงในไฟล์ได้

2. การแทรก vb script ในไฟล์ตัวบอกรายละเอียดของ script ที่โปรแกรม ASP จะต้องแปลและส่งผลออกมาที่ Browser จะเปิดด้วย <% และปิดด้วย %> สิ่งที่อยู่ระหว่างเครื่องหมาย 2 อันนี้จะต้องเป็น script ถ้านำ html มาใส่จะ error เมื่อเรียกใช้การพิมพ์มีดังตัวอย่าง คือเปิดหน้าคำว่า for ด้วย <% และเริ่มแทรก script คือ for i = 3 to 7 จากนั้นจึงปิดหลังตัว i ด้วย > ซึ่งในขณะนี้คือหมดในส่วนของ script แล้ว จากนั้นก็จะเป็น html ธรรมดาไปจนถึงคำว่า size= ก็จะเปิด script อีก แล้วนำตัวแปร i มาใส่ สรุปได้ว่าในส่วนของ script จะต้องเปิดด้วย <% และปิดด้วย %> การแทรก script นี้จะแทรกในส่วนใดของไฟล์ asp ก็ได้แต่ส่วนที่แสดงผลออกมาให้ผู้เรียกใช้ไฟล์ผ่าน Browser คือส่วนที่อยู่ระหว่าง<HTML> จนถึง </HTML>

[/asp/hello2.asp]

```
<HTML>
```

```
<BODY>
```

```
<%
```

- กำหนดให้ i มีค่าตั้งแต่ 3-7 และนำค่านี้ไปใส่แทน i ในส่วนของ size =

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วนไปเรื่อยๆ ตั้งแต่ 3 4 ... ไปจนถึง 7 จากนั้นจึงเปลี่ยน คำสั่งไปที่ next แล้วจึงปิดไฟล์ REM ถ้ามีการแทรก คำสั่งต่อจากบรรทัดนี้ คำสั่งต่างๆจะถูกแปลให้ทำงานได้

```
For i = 3 To 7 %>
```

```
<FONT SIZE=<% = i %>>
```

```
Hello World!</FONT><BR>
```

```
<% Next %>
```

<!-- แต่ถ้านำคำสั่งมาแทรกข้างล่างบรรทัดนี้หรือแทนที่ คำสั่งไม่สามารถทำงานได้

เพราะอยู่นอกเหนือตัวบอกรขอบเขตของ script ที่โปรแกรม ASP จะต้องแปล -->

```
</BODY>
```

```
</HTML>
```

3. การ comment ใน asp ไฟล์การแทรกคำอธิบายในส่วนของ source code มีดังในตัวอย่างด้านบนคือ ถ้าแทรกระหว่าง script จะใช้คำว่า REM หรือ เครื่องหมาย ' (การใช้เครื่องหมาย ' จะประหยัดเนื้อที่ และ ขนาดไฟล์มากกว่า)หรือถ้าแทรกระหว่าง html code จะใช้ <!-- comment here -->

4. การแทรกวันที่และไฟล์ ในไฟล์ .asp

```
[ /asp/hello3.asp ]
```

```
<%
```

- กำหนดตัวแปรคำว่า showdate ให้เป็นวันที่ธรรมดา

- ซึ่งคำว่า showdate/shownow สามารถเปลี่ยนเป็นคำอื่นๆที่ไม่ขัดต่อการตั้งตัวแปรได้ เช่นคำว่า

apple,

```
i_want_to_see_date เป็นต้น
```

```
showdate = date
```

- กำหนดตัวแปรคำว่า shownow ให้เป็นวันที่แบบยาว

```
shownow = now
```

```
%>
```

```
<HTML>
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<% For i = 3 To 7 %>
<FONT SIZE=<%= i %>>
Hello World!</FONT><BR>
<% Next

showdate2 = date
shownow2 = now

%>

<HR>
นำวันที่จากตัวแปรที่กำหนดไว้ข้างบน tag HTML มาแสดง <BR>
showdate = <%=showdate%> และ shownow = <%=shownow%>

<HR>
นำวันที่จากตัวแปรที่กำหนดไว้ข้างล่าง script คำว่า next มาแสดง <BR>
showdate2 = <%=showdate2%> และ shownow2 = <%=shownow2%>

<HR>
และต่อไปนี้เป็น การแทรกวันที่โดยตรง โดยไม่ต้องตั้งตัวแปรก่อน <BR>
<%=date%> หรือ <%=now%>

</BODY>
</HTML>

```

การแทรกไฟล์ ในไฟล์ .asp มีประโยชน์คือลดการซ้ำซ้อนในการที่จะต้องใส่ข้อความที่เหมือนกัน หรือใส่ตัวแปร / รูปภาพในทุกๆไฟล์ ซึ่งจะช่วยให้สะดวกในการแก้ไขในภายหลัง ไม่ต้องเสียเวลานำไฟล์ทั้งหมดมาแก้เพียงแก้ที่ไฟล์ที่แทรกไฟล์เดียวก็สามารถปรากฏผลความเปลี่ยนแปลงได้พร้อมกันทั้งหมด ให้เห็นภาพง่ายคือข้อความที่เป็น footnote ของ Web page เป็นต้น

5.7 การแทรกไฟล์

ขั้นตอนที่ 1 ให้สร้างไฟล์โดยใช้ notepad พิมพ์อะไรลงไปก็ได้ 2-3 บรรทัด จากนั้นให้ save โดยตั้งชื่อและนามสกุลไฟล์ให้เป็น insertit.inc โดย save ไฟล์นี้ไว้ใน folder asp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 save ไฟล์ต่อไปนี้ และเรียกเพื่อดูผลผ่านบราวเซอร์

[/asp/hello4.asp]

```
<HTML>
<BODY>
<% For i = 3 To 7 %>
<FONT SIZE=<% = i %>>
Hello World!</FONT><BR>
<% Next %>
</font> ตัวอย่างนี้มาจากไฟล์ตัวอย่างของ ASP</a>
<hr>
ต่อไปนี้มาจากไฟล์ insertit.inc<br>
<!--#include virtual="/asp/insertit.inc"-->
</BODY>
</HTML>
```

5.8 สาเหตุที่จะทำให้เกิดปัญหา

1. การเปิดปิด asp tag <% และ %>
2. การใช้ script เช่นเปิด if else แต่ลืม ปิด end if เป็นต้น หรือพิมพ์คำสั่งผิด การตั้งตัวแปรที่ผิดหลักการของ VB script
3. ถ้าหากมีการใช้ตารางร่วมด้วย ถ้าเปิดตารางขึ้นมาผิด หรือปิดผิด จะทำให้ ไม่มีผลแสดงออกมาต้องระวังด้วย
4. กำหนดขั้นตอนของ script ผิดพลาดเช่น เรียกใช้ตัวแปรที่ยังไม่ได้กำหนด
5. การใช้คำสั่ง include ไฟล์ แต่ไม่มีไฟล์อยู่หรือวางตำแหน่งผิด
6. การอ้าง alias ของ directory หรือไฟล์ ผิดที่
7. การเรียกใช้ฐานข้อมูล โดยที่ไม่ได้ add DSN name, การพิมพ์ชื่อ table หรือ field ผิด
8. ภาษาของฐานข้อมูลไม่เข้ากันกับ web server

นอกจากนี้ก็ยังมีส่วนอีกหลายประการ ที่ต้องพบและทดลองแก้ปัญหาเหล่านั้น เมื่อมีข้อความแสดงความคิดพลาดขึ้นมาบน Browser สิ่งที ข้อความ error จะแจ้งมาด้วยคือ บรรทัดที่ error แต่ถ้ามีแต่หน้าจอว่างๆ ไม่มีข้อความให้ใช้วิธี view source ดูซึ่งอาจจะมีข้อความแสดงอาการ error อยู่ในนั้น เมื่อได้บรรทัดที่ error แล้วให้กลับไปดูใน source codeแล้วหาสาเหตุให้พบและทำการแก้ไขตามอาการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับบางอาการอาจจะไม่มีข้อความแสดงขึ้นมา แต่จะมีบางตัวแปรค่าของมันจะหายไป มีวิธีค้นหาข้อผิดพลาดคือให้สร้างไฟล์ที่รับค่าตัวแปรเหล่านั้น จะทำให้ทราบว่าตัวแปรหายไป

5.9 การตรวจสอบ Browser ว่าสามารถใช้ ASP ได้หรือไม่

เนื่องจาก Browser แต่ละยี่ห้อมีความสามารถไม่เหมือนกัน หรือยี่ห้อเดียวกันแต่ต่างรุ่นความสามารถก็ต่างกันจึงต้องมีวิธีตรวจสอบยี่ห้อและรุ่น เพื่อที่จะส่งหน้าที่เหมาะสมไปให้ผู้ใช้งาน Browser ต่างๆกันนั้น เช่น Netscape ไม่สามารถใช้ ActiveX ได้ แต่ IE ใช้ได้ ดังนั้นเมื่อมีใครเข้ามาในเว็บไซต์ที่มี ASP จึงต้องกำหนดหน้าที่เหมาะสมให้กับ Netscape (หน้าที่ไม่มี ActiveX) และส่งหน้าธรรมดาหรือหน้าที่มี ActiveX ให้กับ IE เป็นต้น

5.10 การเรียกใช้และตรวจสอบ

[/asp/browsercap.asp]

- <%
- เปิดเครื่องมือช่วยในการตรวจสอบ Browser (เรียกใช้ ASP component) จะต้องเปิดแบบนี้ทุกครั้งถ้าต้องการใช้
- แต่อาจจะเปลี่ยน คำว่า bc ให้เป็นตัวแปรอื่นๆได้ แต่ถ้าเปลี่ยนแล้วส่วนอื่นๆที่เกี่ยวข้องจะต้องเปลี่ยนไปด้วย
- Set bc = Server.CreateObject("MSWC.BrowserType")
- ตั้งตัวแปร bname ให้แทนยี่ห้อของ browser ผลที่ออกมาจะเป็น unknown, Netscape , IE เป็นต้น
bname=bc.browser
- ตั้งตัวแปร brver ให้แทน version ของ browser ผลที่ออกมาจะเป็นตัวเลขเช่น 1 2 3 4 5
brver= CINT(bc.version)
- ตัวแปรต่อไปนี้แสดงความสามารถของ browser ถ้ามีค่าเท่ากับ 1 แสดงว่าสามารถใช้ได้ แต่ถ้าออกมาเป็น 0 แสดงว่าไม่สนับสนุน

- เช่น ถ้า frames = 0 หมายความว่า browser ไม่สามารถใช้ Frame ได้ แต่ถ้าเท่ากับ 1 จะสามารถใช้ Frame ได้เป็นต้น

ActiveX = bc.ActiveXControls

javaapplets = bc.javaapplets

javascript = bc.javascript

vbscript = bc.vbscript

frames = bc.frames

cookies = bc.cookies

backgroundsounds = bc.backgroundsounds

- platform คือระบบปฏิบัติการของผู้ใช้ ค่าที่ส่งออกมาจะเป็น Windows 95 เป็นต้น
platform = bc.platform

- การประยุกต์ใช้ โดยมีหลักการคือ เราจะส่งผู้เข้าเว็บไซต์ของเราไปยังหน้าหลักที่เหมาะสมกับ

- Browser ที่เขาใช้ โดยใช้ ภาษา vb script select - case เข้ามาช่วย

select case bname

case "IE"

- ถ้าเป็น IE ก็มาเลือก version ต่อแล้วจึงย้าย ผู้เยี่ยมชมเว็บไซต์ไปตาม version ของ Browser ที่เขาใช้

select case brver

case "4"

response.redirect("/asp/indexie4.asp")

case "3"

response.redirect("/asp/indexie3.asp")

case else

response.redirect("/asp/indexie.asp")

end select

case "Netscape"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในทำนองเดียวกันกับ IE ถ้าเป็น Netscape ก็ย้ายไปตาม version ที่เหมาะสมเช่นกัน

```
select case brver
case "4"
response.redirect("/asp/indexnet4.asp")
case "3"
response.redirect("/asp/indexnet3.asp")
case else
response.redirect("/asp/indexnet.asp")
end select
```

- ถ้าไม่ใช่ทั้ง IE หรือ Netscape ก็ย้ายไปที่ไฟล์ index.asp

```
case else
response.redirect("/asp/index.asp")
end select
%>
[ /asp/activex_check.asp ]
<%
```

- การตรวจสอบเฉพาะสิ่งที่ต้องการแล้ว redirect ไปยังไฟล์ที่กำหนด

```
Set bc = Server.CreateObject("MSWC.BrowserType")
ActiveX = bc.ActiveXControls
if ActiveX = 1 then
response.redirect("/asp/can_use_activex.asp")
else
response.redirect("/asp/can_not_use_activex.asp")
end if
%>
[ /asp/activex_check2.asp ]
<%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การตรวจสอบเฉพาะ ActiveX แล้วแทรก ActiveX ในไฟล์ หรือไม่แทรกถ้า Browser ไม่สามารถใช้ได้

```
Set bc = Server.CreateObject("MSWC.BrowserType")
```

```
ActiveX = bc.ActiveXControls
```

```
%>
```

```
<html>
```

```
<body>
```

```
<% if ActiveX = 1 then %>
```

สามารถใช้ ActiveX ได้ ให้เริ่มแทรก ActiveX Code ได้เลย

```
<% else %>
```

ไม่สามารถใช้ ActiveX ได้ ให้ใช้วิธีอื่นในการทำงานแทน ActiveX

```
<% end if %>
```

```
</body>
```

```
</html>
```

5.11 ไฟล์ Global.asa

global.asa เป็นไฟล์เสริมเพิ่มเติมสำหรับการใช้งาน ASP ในชั้นสูง ไฟล์นี้มีประโยชน์คือ จะทำหน้าที่เก็บค่าต่างๆไว้ใช้ในภายหลัง ไฟล์นี้จะต้องเก็บอยู่ที่ Root ของ ASP application โดย web server จะอ่านค่าต่างๆในไฟล์นี้มาประมวลผลและเก็บไว้ในหน่วยความจำ รอกการเรียกใช้จากผู้ใช้ทั่วไปหรือผู้พัฒนาโปรแกรม การแก้ไขไฟล์นี้ในภายหลังจะไม่เกิดผลการเปลี่ยนแปลงเลย เหมือนกับไฟล์ .asp ทั่วไป จะต้องทำการ Restart Server หรือโปรแกรม PWS/IIS ถึงจะเกิดการเปลี่ยนแปลง ภายในไฟล์จะแบ่งเป็น 2 ส่วนหลักๆคือ Application Events และ Session Events

5.12 การสร้างไฟล์ Global.asa

```
[ /global.asa ]
```

```
<SCRIPT LANGUAGE=VBSCRIPT RUNAT=SERVER>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เปิดไฟล์ Text สำหรับบันทึกจำนวนผู้เข้าเยี่ยมชมเว็บไซต์

- ในส่วนของ Sub Application_OnStart นี้สามารถใส่ application อื่นๆ ได้อีกตามต้องการ

- แต่สิ่งที่ใส่ในส่วนนี้ควรเป็น โปรแกรมที่ต้องมีการ share หรือใช้ตัวแปรนี้ร่วมกันทุกคนที่เข้ามาเว็บไซต์

- จำนวน visitor หรืออาจจะนำมานับจำนวนคนที่กำลัง connect กับเว็บไซต์อยู่ก็ได้

Sub Application_OnStart

```
VisitorCountFilename = Server.MapPath("/cgi-bin/visitors.txt")
```

```
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
```

```
Set Out= FileObject.OpenTextFile (VisitorCountFilename, 1, False, False)
```

```
Application("visitors") = Out.ReadLine
```

```
Application("VisitorCountFilename") = VisitorCountFilename
```

End Sub

- ถ้าเปิด Application ใดๆ ใน Application_OnStart ควรต้องมาปิดในนี้ด้วย

Sub Application_OnEnd

```
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
```

```
Set Out= FileObject.CreateTextFile (Application("VisitorCountFilename"), True, False)
```

```
Out.WriteLine(Application("visitors"))
```

```
Out.Close
```

End Sub

- Sub Session_OnStart นี้ใช้สำหรับส่งหรือบันทึกตัวแปรไว้เพื่อให้ผู้ใช้เว็บไซต์แต่ละคน

Sub Session_OnStart

- การนำ session สองตัวแปรนี้ไปใช้ในไฟล์ .asp

```
Session("PageTheme") = "background=/bg.jpg text=#000000 bgcolor=#ffffff link=#0000ee
```

```
vlink=#551A8B
```

```
alink=#ff0000"
```

```
Session("PageTitle") = " Website.com - Your website title here "
```

- การเขียนจำนวนผู้ใช้เว็บเพิ่มอีก 1 คน

```
Application.Lock
```

```
Application("visitors") = Application("visitors") + 1
```

```
t_visitors = Application("visitors")
```

```
Application.Unlock
```

```
Session("VisitorID") = t_visitors
```

```
If t_visitors MOD 15 = 0 Then
```

```
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
```

```
Set Out= FileObject.CreateTextFile (Application("VisitorCountFilename"), True, False)
```

```
Application.Lock
```

```
Out.WriteLine(t_visitors)
```

```
Application.Unlock
```

```
End If
```

```
End Sub
```

- การลบตัวแปรที่กำหนดไว้ใน Sub Session_OnStart

```
Sub Session_OnEnd
```

```
Session("PageTheme") = Nothing
```

```
Session("PageTitle") = Nothing
```

```
End Sub
```

```
</SCRIPT>
```

5.13 การเรียกใช้ตัวแปรและ Application ที่กำหนดไว้ใน Global.asa

1. Application ในการนับจำนวนผู้ใช้เว็บไซต์จะต้อง สร้างไฟล์ text ว่างๆชื่อ visitors.txt ไว้ใน /cgi-bin ก่อนจากนั้นให้เรียกใช้เว็บไซต์จากส่วนใดก็ได้ที่มีนามสกุลไฟล์ .asp ซึ่งจะทำให้ Webserver อ่านไฟล์ Global.asa จากนั้น ASP จะเขียนจำนวนตัวเลขไว้ในไฟล์ text ดังกล่าวในครั้งต่อไปที่เรียกไฟล์ .asp ใดๆในเว็บไซต์อีก จำนวนตัวเลขที่อยู่ในไฟล์นี้ก็จะเพิ่มขึ้นมาอีก 1 ส่วนการนำตัวเลขในไฟล์ visitors.txt มาแสดงในเว็บไซต์ ก็เพียงแต่แทรกcode ต่อไปนี้ในไฟล์.asp ในตำแหน่งที่ต้องการ

```
You are visitors No. <%=Application("visitors")%> Since mm/dd/yy
```

2. การนำ session PageTheme และ Session PageTitle ไปใช้ในไฟล์ .asp ใดๆในเว็บไซต์ให้คุณสร้างไฟล์ .asp อะไรก็ได้ขึ้นมา 1 ไฟล์ แต่ในส่วนหัวของไฟล์ให้ใส่ code ต่อไปนี้ไว้จากนั้นลองเรียกไฟล์ดูเพื่อดูผล ไฟล์ทุกไฟล์ในเว็บไซต์ ถ้าใส่ code นี้ไว้จะทำให้มี Title และรูปแบบสีของลิงค์ ตัวอักษร ภาพพื้นหลังเหมือนกัน ถ้าต้องการเปลี่ยนส่วนใดส่วนหนึ่งก็เพียง ไปเปลี่ยนที่ไฟล์ global.asa เท่านั้นแล้ว restart IIS หรือ PWS ใหม่ รูปแบบก็จะเปลี่ยนไปทันที โดยไม่ต้องไปเสียเวลาแก้ไฟล์แต่ละไฟล์เหมือนกับที่ใช้ .htm หรือ .html ธรรมดา

```
<html>
<head>
<title><%=Session("PageTitle")%></title>
</head>
<body <%=Session("PageTheme")%>>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

CGI PROGRAM ON THE WORLD WIDE WEB

CGI ย่อมาจาก COMMON GATEWAY INTERFACE เป็นวิธีนำเสนอข้อมูลแบบ DYNAMIC ทำให้เราสามารถสร้าง Web Page ในช่วงเวลาที่ผู้ใช้เข้ามา ทำให้ได้ข้อมูลที่เป็นจริงในขณะนั้น เช่น เวลา, วันที่, จำนวนผู้ที่เข้ามาใช้บริการ และอื่นๆ อีกมากมาย รวมทั้งรับข้อมูลจากผู้ใช้เพื่อนำไปดำเนินการ เช่น การค้นหาข้อมูล, การทำสมุดเยี่ยม, การทำแบบสอบถาม เป็นต้น

CONFIGURING THE SERVER

ก่อนที่จะทำการรัน CGI ได้ ต้องทำการ CONFIGURING SERVER ก่อนโดยแก้ไขไฟล์/etc/http

/conf/httpd.conf สิ่งแรกคือ SERVERROOT DIRECTIVE เพื่อชี้ DIRECTORY ของ SERVER SOFT-WARE

```
SERVERROOT /home/http/html
```

RUNNING CGI SCRIPTS

SCRIPTALIAS DIRECTIVE ในไฟล์ SERVER RESOURCE MAP (/etc/http/conf/srm.conf)

เพื่อชี้ DIRECTORY ที่เก็บ CGI SCRIPT เพื่อความปลอดภัยของระบบ

```
SCRIPTALIAS /cgi-bin/ /home/httpd/cgi-bin/
```

ตัวอย่างเช่น เมื่อผู้ใช้อ้างถึง URL

```
http://your_host.com/cgi-bin/welcome
```

ใน SERVER จะหมายถึง

```
/home/httpd/cgi-bin/welcome
```

สามารถกำหนดได้มากกว่า 1 DIRECTORY

```
SCRIPTALIAS /cgi-bin / /home/httpd/cgi-bin/
```

```
SCRIPTALIAS /my-cgi-bin/ /home/httpd/my-cgi-bin/
```

มี DIRECTIVE ที่อนุญาตให้รันโปรแกรมที่อยู่นอกเหนือจาก DIRECTORY ที่กำหนดขึ้นกับนามสกุลของไฟล์คือ

```
ADDDTYPE application/x-httpd-cgi .pl .sl .cgi
```

PROGRAMMING IN CGI BY PERL

นี่คือตัวอย่างของ CGI PERL PROGRAM ที่สร้าง VIRTUAL DOCUMENT แบบง่ายๆ

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

```
print "Content-type:      text/plain", "\n\n";
```

```
print "Welcome to my WWW Server!", "\n\n";
```

```
$remote_host = $ENV{'REMOTE_HOST'};
```

```
print "You are visiting form", $remote_host, ". .";
```

```
print "Happy navigating!", "\n";
```

```
exit (0);
```

บรรทัดแรกของโปรแกรมจะบอกว่า SERVER จะรันตัวแปร PERL จาก /usr/bin/perl เพื่อ EXECUTE

โปรแกรม โปรแกรมส่งผลลัพธ์ที่เป็น HTTP HEADDER เพื่อบอกชนิดของเอกสารให้กับ Browser ในที่นี้ content-type เป็น text/plain

การขึ้นบรรทัดใหม่ 2 บรรทัด (\n) หลังจาก HEADER เพราะ HTTP ต้องการบรรทัดว่าง 1 บรรทัด ระหว่าง HEADER กับ BODY

คำสั่ง print ที่ 2 หลังจาก HEADER เป็นการแสดง REMOTE HOST ของผู้ใช้ ซึ่งได้ข้อมูลมาจาก ENVIRONMENT VARIABLE ชื่อ REMOTE_HOST

USING ENVIRONMENT VARIABLES

ข้อมูลสำคัญ ๆ ที่ CGI APPLICATION ต้องการส่วนมากจะอยู่ใน ENVIRONMENT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้นำไปใช้ประโยชน์ด้านการค้า VARIABLE ดังที่แสดงในตารางต่อไปนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ENVIRONMENT VARIABLE	รายละเอียด
GATEWAY_INTERFACE	REVISION ของ CGI ที่ SERVER ใช้
SERVER_NAME	SERVER HOSTNAME หรือ IP ADDRESS
SERVER_SOFTWARE	ชื่อและ VERSION ของ SERVER SOFTWARE ที่ใช้ตอบ clint
SERVER_PROTOCOL	ชื่อและ REVISION ของ INFORMATION PROTOCOL ที่ใช้ ในการเข้ามา
SERVER_PORT	PORT NUMBER ที่ HOST ที่ SERVER กำลังรันอยู่
REQUEST_METHOD	วิธีการรับข้อมูลที่ส่งมา
PATH_INFO	EXTRA PATH INFORMATION ที่ส่งไปให้ CGI PROGRAM
ENVIRONMENT VARIABLE	รายละเอียด
PATH_TRANSLATED	PATH_INFO ที่แปลงแล้ว
SCRIPT_NAME	VIRTUAL PATH ของ SCRIPT ที่กำลัง EXECUTE
DOCUMENT_ROOT	DIRECTORY ของ WEB DOCUMENT
QUERY_STRING	QUERY INFORMATION ที่ส่งมายังโปรแกรมอยู่ต่อกับ URL ด้วยเครื่องหมาย “?”
REMOTE_HOST	REMOTE HOSTNAME ของผู้ใช้ที่เข้ามา
REMOTE_ADDR	REMOTE IP ADDRESS ของผู้ใช้ที่เข้ามา
AUTH_TYPE	วิธีที่ใช้รับรองความถูกต้องของผู้ใช้
REMOTE_USER	ชื่อที่ถูกรับรองแล้วของผู้ใช้
CONTENT_TYPE	MIME TYPE ของ QUERY DATA เช่น “text/html”
CONTENT_LENGTH	ความยาวของข้อมูล (เป็น ไบต์หรือจำนวนตัวอักษร) ที่ส่งผ่าน มายัง CGI PROGRAM
HTTP_FORM	E-mail ADDRESS ของผู้ใช้ที่เข้ามา บราวเซอร์ส่วนมากไม่ สนับสนุนตัวแปรนี้
HTTP_ACCEPT	รายการของ MIME TYPE ที่ clint สามารถรับได้
HTTP_USER_AGENT	บราวเซอร์ clint ที่ใช้เข้ามา
HTTP_REFERER	URL ของเอกสารที่ clint ชี้ไป ก่อนที่จะเข้า CGI PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะใช้โปรแกรมตัวอย่างเพื่อแสดงว่าตัวแปรเหล่านี้ถูกใช้ใน CGI PROGRAM อย่างไร

ABOUT THIS SERVER

```
#!/usr/bin/perl

print "Content-type: text/html", "\n\n";

print "<HTML>", "\n";

print "<HEAD><TITLE>About this server</TITLE></HEAD>", "\n";

print "<BODY><H1>About this server</H1>", "\n";

print "<HR><PRE>";

print "Server Name :", $ENV{'SERVER_NAME'}, "<BR>", "\n";

print "Running on port :", $ENV{'SERVER_PORT'} "<BR>", "\n";

print "Server Software :", $ENV{'SERVER_SOFTWARE'} "<BR>", "\n";

print "Server Protocol :", $ENV{'SERVER_PROTOCOL'} "<BR>", "\n";

print "CGI Revision :", $ENV{'GATEWAY_INTERFACE'} "<BR>", "\n";

print "<HR></HTML>", "\n";

exit (0);
```

ผลลัพธ์ของโปรแกรมอาจจะเป็นดังนี้

```
<HTML>
<HEAD><TITLE>About this server</TITLE></HEAD>
<BODY><H1>About this server</H1>
<HR><PRE>

Server Name :      vans.kaiwal.com

Running on port :  80

Server Software :  Apache/1.4.2

Server Protocol   : HTTP/1.0

CGI Revision      :   CGI/1.1

<HR></PRE>

</BODY></HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FROM AND CGI

การใช้ฟอร์มรับข้อมูลแบ่งเป็น 2 ส่วนสำคัญๆคือส่วน HTML CODE และ CGI PROGRAM ส่วน HTML CODE จะสร้างฟอร์มเพื่อรับข้อมูล และ CGI PROGRAM จะนำข้อมูลเหล่านี้ไปใช้

THE FORM TAG

รูปแบบของฟอร์มขึ้นต้นด้วย

```
<FORM ACTION="/cgi-bin/program.pl" METHOD="POST">
```

<FORM> TAG เป็นตัวเริ่มฟอร์ม ATTRIBUTE อีก 2 ตัวใน <FORM> TAG (ACTION และ METHOD) ACTION ATTRIBUTE กำหนด URL ของ CGI PROGRAM ที่จะนำข้อมูลเหล่านี้ไปใช้ METHOD ATTRIBUTE กำหนดว่า SERVER จะส่งข้อมูลของฟอร์มไปยังโปรแกรมอย่างไร POST จะส่งข้อมูลจาก STANDARD INPUT และ GET จะส่งข้อมูลที่ได้จาก ENVIRONMENT VARIABLE ถ้าไม่กำหนดค่า DEFAULT จะเป็น GET การรับข้อมูลส่วนใหญ่ในฟอร์มจะรับโดย <INPUT> TAG TYPE ATTRIBUTE ใน <INPUT> TAG จะเป็นตัวกำหนดชนิดของข้อมูลที่จะรับ มีดังนี้

Text field - รับข้อมูลที่เป็นตัวอักษร

```
<INPUT TYPE="text" NAME="name" VALUE="value" SIZE=size  
MAXLENGTH=length>
```

ข้อมูลที่รับเข้าไปจะอยู่ในรูปของ name=value เช่น

```
<INPUT TYPE="text" NAME="user" SIZE=30>
```

ถ้าผู้ใช้ใส่ข้อมูลในช่องนี้เป็น "John" ข้อมูลที่ส่งไปยัง CGI PROGRAM จะเป็น user=John SIZE ATTRIBUTE จะกำหนดขนาดช่องที่จะแสดงถ้ากรอกข้อมูลเกินขนาดช่องก็จะ SCROLL MAXLENGTH ATTRIBUTE จะกำหนดความยาวสูงสุดของข้อมูลที่อนุญาตให้รับ

Password field - รับข้อมูลโดยแสดงเครื่องหมาย * แทนข้อมูลจริง

```
<INPUT TYPE="password" NAME="name" VALUE="value" SIZE=size  
MAXLENGTH=length>
```

Hidden field - ไม่แสดงข้อมูลในส่วนนี้

```
<INPUT TYPE="hidden" NAME="name" VALUE="value">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Checkbox - แสดงตัวเลือกแบบเลือกได้พร้อมกันมากกว่า 1 ตัว

```
<INPUT TYPE="checkbox" NAME="name">
```

บราวเซอร์จะแสดงปุ่มหรือช่องสี่เหลี่ยมให้ผู้เลือก ถ้าผู้ใช้เลือก ค่าที่ส่งจะเป็น name=on เราสามารถกำหนดค่าเองแทนค่า "on" ได้โดยกำหนด VALUE ATTRIBUTE เช่น

```
<INPUT TYPE="checkbox" NAME="name" VALUE="yes">
```

ถ้าผู้ใช้เลือกค่าที่ส่งไปจะเป็น name=yes

Radio button - แสดงตัวเลือกแบบกลุ่มที่เลือกได้ตัวเดียว

```
<INPUT TYPE="radio" NAME="name" VALUE="value">
```

รูปแบบจะเป็น name=value ตั้งค่า

หลักการตั้งตัวเลือกแบบ radio คือทุกตัวเลือกตั้งใช้ชื่อเดียวกัน เพื่อให้บราวเซอร์รู้ว่าเป็นตัวแปรกลุ่มเดียวกัน

ตัวเลือกทั้งแบบ Checkbox และ Radio สามารถกำหนดค่า DEFAULT ได้โดยเพิ่ม CHECKED ATTRIBUTE เข้าไป

Menu - แสดงตัวเลือกแบบ dropdown menu

```
<SELECT NAME="name" SIZE=1>
```

```
<OPTION SELECTED>First
```

```
<OPTION>Second
```

```
|
|
|
```

```
</SELECT>
```

Scrolled list - แสดงตัวเลือกแบบ Scroll list

```
<SELECT NAME="name" SIZE= n MULTIPLE>
```

```
<OPTION SELECTED>First
```

```
<OPTION>Second
```

```
|
|
|
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 </SELECT>
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Menu และ Scroll list ใช้ <SELECT> TAG เหมือนกัน ค่า SIZE ATTRIBUTE จะกำหนดว่าเป็น Menu หรือ Scroll List โดยค่า 1 จะสร้าง Menu และอีกค่าที่มากกว่า 1 จะสร้าง Scroll List ซึ่งค่า SIZE จะเป็นตัวกำหนดจำนวนตัวเลือกที่จะแสดง ตัวเลือกสามารถกำหนดค่า DEFAULT ได้โดยเพิ่ม SELECTED ATTRIBUTE เข้าไปใน <OPTION> TAG และถ้าต้องการให้เลือกตัวเลือกได้มากกว่า 1 ตัว ให้เพิ่ม MULTIPLE ATTRIBUTION เข้าไปใน <SELECT> TAG

Multiline text field - รับข้อมูลที่หลายๆ บรรทัด

```
<TEXTAREA ROWS=yy COLS=xx NAME="name">
```

```
</TEXTAREA>
```

โดย ROWS จะกำหนดจำนวนบรรทัด (ความสูง) และ COLS จะกำหนดจำนวนคอลัมน์ (ความกว้าง) ของช่องที่จะแสดง เราสามารถกำหนดค่า DEFAULT ได้โดยเพิ่มเข้าไประหว่าง <TEXTAREA> กับ </TEXTAREA> TAG เช่น

```
<TEXTAREA ROWS=10 COLS=40 NAME="comment">
```

```
This is a default information.
```

```
a little bit...little more
```

```
</TEXTAREA>
```

ขึ้นบรรทัดใหม่โดยไม่ต้องใส่
 TAG ข้อมูลที่แสดงจะแสดงจะเหมือนกับที่พิมพ์เข้าไป

Submit buttons - ส่งข้อมูลจาก FORM ไปยัง CGI PROGRAM มี 3 แบบ

```
<INPUT TYPE="submit" VALUE="message">
```

Submit แบบปุ่มเดียว

```
<INPUT TYPE="submit" NAME="name" VALUE="option1">
```

```
<INPUT TYPE="submit" NAME="name" VALUE="option2">
```

Submit แบบหลายปุ่ม

ถ้าผู้ใช้คลิกบน "option1" ข้อมูลที่ส่งไปจะเป็น name="option1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 <INPUT TYPE="IMAGE" SRC="/image" NAME="name" VALUE="value">
 ไม่วาทกรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้รูปภาพแทนปุ่มเมื่อผู้ใช้คลิกรูปภาพ ข้อมูลที่ส่งไปจะเป็น COORDINATE ของจุดที่คลิก
name.x=250&name_y=20

Reset button - CLEAR คำทั้งหมดใน FORM

```
<INPUT TYPE="reset" VALUE="message">
```

จบ form - สิ้นสุดการใช้ฟอร์ม

```
</FORM>
```

มี 2 วิธีในการส่งข้อมูลของฟอร์ม: GET และ POST ข้อแตกต่างของทั้ง 2 วิธีคือวิธีการที่จะส่งข้อมูลของฟอร์มไปยัง CGI PROGRAM ถ้าใช้วิธี GET, QUERY STRING จะถูกส่งต่อไปกับ URL ของโปรแกรม QUERY STRING นี้สามารถอ้างถึงได้โดย ENVIRONMENT VARIABLE ชื่อ "QUERY_STRING" แต่ละตัวแปรจะแบ่งด้วยเครื่องหมาย "&" ข้อมูลที่เข้ามาถ้าเป็นช่องว่างจะแทนด้วยเครื่องหมาย "+" และสัญลักษณ์พิเศษจะเป็นเลขฐาน 16 ต้อง DECODE ด้วยคำสั่ง

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

เพื่อเปลี่ยน (TRANSLATE) เครื่องหมาย "+" เป็นช่องว่าง

```
$value =~ s/%([\dA-Fa-f][\dA-Fa-f])/pack ("C" , hex ($1))/eg;
```

เปลี่ยนค่าที่เป็นเลขฐาน 16 เป็นตัวอักษร

SERVER SIDE INCLUDE: SSI

SSI เป็นคำสั่งที่ใส่ไว้ในเอกสาร HTML เพื่อใช้ EXECUTE โปรแกรมภายนอกหรือแสดง ENVIRONMENT VARIABLE เช่นการแสดงวัน, เวลา หรือการแสดงผล COUNTER เป็นต้น

CONFIGURATION

สิ่งแรกคือ SET นามสกุลของไฟล์ที่จะให้ SERVER PARSE โดยแก้ไขไฟล์

```
/etc/http/conf/srm.conf
```

```
AddType text/x-server-parsed-html .shtml
```

แก้ไขไฟล์ /etc/httpd/conf/access.conf กำหนดว่าจะอนุญาตให้ใช้ SSI DERECTIVE ชนิดใด

บ้างในเอกสาร HTML

ถ้าต้องการแสดง ENVIRONMENT VARIABLE และ FILE STATISTIC ให้เพิ่ม

```
Includes
```

ถ้าต้องการ EXECUTE โปรแกรมภายนอกจากเอกสาร HTML ให้เพิ่ม ExecCGI

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการใช้ทั้ง 2 แบบ

Option Includes ExecCGI

SSI DIRECTIVES มีรูปแบบคือ `<!--#command parameter="argument"`

ตารางแสดง SSI DIRECTIVE

คำสั่ง	PARAMETER	รายละเอียด
Echo	var	แสดงค่าของ SSI VARIABLE
Include		แสดงเอกสารใน current file
	file	อ้างถึงเอกสาร
	virtual	Virtual path ของเอกสาร
Fsize	file	แสดงขนาดของไฟล์ที่กำหนด
Flastmod	file	แสดงวันและเวลาที่ไฟล์เปลี่ยนแปลงครั้งสุดท้าย
Exec		EXECUTE โปรแกรมภายนอกแล้วแสดงผลลัพธ์ในเอกสารปัจจุบัน
	cmd	APPLICATION ใด ๆ ที่อยู่บน HOST
	cgi	CGI PROGRAM
Config		แก้ไขรูปแบบต่าง ๆ ของ SSI
	errmsg	DEFAULT ERROR MESSAGE
	sizefmt	รูปแบบสำหรับขนาดของไฟล์
	timefmt	รูปแบบของเวลา

เอกสาร HTML ที่ใช้ SSI DIRECTIVE นั้น จะกำหนด EXTENTION เป็น .shtml เพื่อให้ SERVER รู้ว่าเป็นไฟล์ที่ต้อง parse

```
<!--#echo var="REMOTE_HOST- ->
```

แสดงค่า ENVIRONMENT VARIABLE

```
<!--#include file="address.html"- ->
```

แสดงเอกสาร HTML ตามที่กำหนด

```
<!--#include virtual="/public/address.html"- ->
```

แสดงเอกสาร HTML ตามที่กำหนดแบบกำหนด path

เอกสาร HTML ที่ใช้ SSI DIRECTIVE นั้น จะกำหนด EXTENTION เป็น .shtml เพื่อให้ SERVER รู้ว่าเป็นไฟล์ที่ต้อง parse

งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงขนาดของไฟล์ที่กำหนด

```
<!-- #flastmod file= "/guideline.ps" - ->
```

แสดงวันที่ครั้งสุดท้ายที่ไฟล์ถูกเปลี่ยนโปรแกรม

```
<!-- #exec cmd= "/usr/bsd/finger $REMOTE_USER@$REMOTE_HOST" - ->
```

รันโปรแกรมที่กำหนดแล้วแสดงผลลัพธ์ในเอกสาร HTML

```
<!-- #exec cgi= "/cgi-bin/counter.pl" - ->
```

รัน CGI โปรแกรมที่กำหนดแล้วแสดงผลลัพธ์ในเอกสาร HTML

```
<!-- #config errmsg= "Error", contact vans@kaiwal.com" - ->
```

SET คำ ERROR MESSAGE

```
<!-- #config sizefmt= "byte" - ->
```

SET รูปแบบการแสดงผลขนาดไฟล์

```
<!-- #config timefmt= "%D %r" - ->
```

SET รูปแบบการแสดงผลเวลา

Introduction to SQL

สร้าง database

```
create table Salary_Info
```

```
(
```

```
    man    character (30) not null,
```

```
    age    integer,
```

```
    salary integer,
```

```
);
```

แสดงข้อมูลทั้งหมดใน database

```
select * from Salary_Info;
```

แสดงข้อมูลเฉพาะบาง field

```
select name from Salary_Info;
```

แสดงข้อมูลแบบมีเงื่อนไข

```
select * from Salary_Info where (salary > 10000);
```

แสดงข้อมูลเฉพาะบาง field แบบมีเงื่อนไข

```
select name from Salary_Info where (salary > 10000);
```

แสดงข้อมูลเฉพาะบาง field แบบมีเงื่อนไขมากกว่า 1 เงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เฉพาะในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
select name from Salary_Info where (salary > 10000) and (age > 30);
```

ใช้ wildcard ในการแสดงข้อมูล

```
select * from Salary_Info where name LIKE '% Johnson';
```

จะแสดงข้อมูลเฉพาะที่มี filed name ลงท้ายด้วย Johnson Update database

เช่นเมื่อเวลาผ่านไป 1 ปี field age ต้องเพิ่มค่าใน field age ได้โดย

```
update Salary_Info set age = (age + 1);
```

เพิ่มข้อมูลลงใน database

```
insert into Salary_Info values ('John Wayne', 27, 15000);
```

เพิ่มข้อมูลลงใน database เฉพาะบาง field โดยกำหนดเฉพาะชื่อ field ที่ต้องการเพิ่ม

```
insert into Salary_Info (name,salary) values ('John Wayne', 15000);
```

ลบข้อมูลใน database

```
delete Salary_Info where name = 'Jack';
```

ลบทั้ง field จาก database

```
alter table Salary_Info drop column age;
```

ลบทุก record จาก database

```
delete from Salary_Info;
```

ลบ database

```
drop table Salary_Info;
```

Sprite

Sprite เป็น Module ที่ Perl ใช้ เพื่อเรียกใช้ไฟล์โดยคำสั่งแบบ SQL use Sprite;

ประกาศว่ามีการใช้ Sprite module

```
$rdb = new Sprite ();
```

ประกาศใช้ Database และเก็บข้อมูลในตัวแปร \$rdb

```
$rdb->set_delimiter ("Read", "::");
```

\$rdb->set_delimiter ("Write", "::"); กำหนดตัวแบ่งฟิลด์ (Delimiter) ในการเขียนและอ่าน
จากไฟล์ ในที่นี้ใช้ "::" เป็นตัวแบ่ง

```
if (! (-e $database) ) {
```

```
    if ( open (DATABASE, ">" . $database) ) {
```

```
        flock (DATABASE, 2);
```

```
        $header = join ($delimiter, "Name", "Year", "Address");
```

```
        print DATABASE $header, "\n";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

flock (DATABASE, 8);
close (DATABASE);
} else {
    print "Error : Cannot create new database file.";
}
}

```

Check ว่ามี Database อยู่หรือไม่ ถ้าไม่มีก็สร้างไฟล์ของ Database ขึ้นมาใหม่ บรรทัดแรกของไฟล์จะต้องเป็นชื่อฟิลด์ (case sensitive) กั้นกลางด้วยตัว Delimiter

```

@data = $ rdb->sql (<<End_Of_Query)
    select * from student . db
    where (Year > 1996)
end_Of_Query
หรือ
@data = $ rdb->sql ("select * from student .db where $ query");

```

เป็นการ Query Sprite จะ record ที่พบกลับมาให้ @data ทุก field ในแต่ละ record จะต่อกันด้วย null character "\0" ค่าแรกใน Array จะไม่ใช่ record แต่จะเป็น flag ที่ชี้ว่า success (1) หรือ fail (0)

```
$ status = shift (@data);
```

เราใช้ค่าแรกของ Array เพื่อ check ว่าค้นหาด้วย success หรือ fail โดยลบค่าแรกจาก @data แล้วนำไปไว้ในตัวแปร \$ status

```
$ no_elements = scalar (@data);check จำนวน record ที่พบ
```

```
$ rdb->sql (<<End_Of_Insert);insert into $ database(Name, Year, Address)
```

values

```
('John Smith', '1998', 'Australia') End_Of_Insert
```

การ Insert record

```
$ rdb->sql (<<End_Of_Delete); delete from $ database
where (Year < 1990)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End_Of_Delete การลบ record

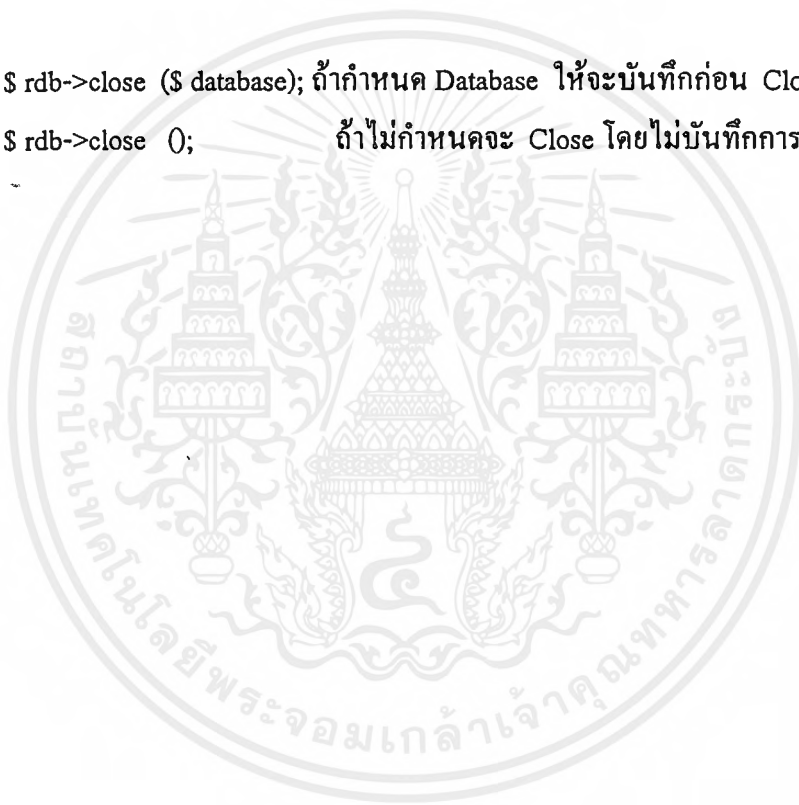
```
$ rdb->sql (<<Update_database); update $ dqtbase
set Year = 2000
where (address = 'Thailand');
```

End_Of_Delete

การ update database

```
$ rdb->close ($ database); ถ้ากำหนด Database ให้จะบันทึกก่อน Close
$ rdb->close (); ถ้าไม่กำหนดจะ Close โดยไม่บันทึกการ Close
```

database



บทที่ 7

การเลือกใช้ Server และ ภาษาที่จะใช้เขียน CGI

การเริ่มต้นเขียนเราต้องมาดูที่ตัว Server ที่เราเป็นเจ้าของกันก่อน โดยทั่วไปแล้ว Server จะมีอยู่ 2 แบบที่นิยมใช้กันก็คือ Server ที่เป็น UNIX และ Server ที่เป็น Windows NT เนื่องจากทั้ง 2 แบบมีข้อดีด้วยกันทั้งคู่ ซึ่งการใช้ในการเขียน CGI จริงๆ และได้ทดสอบทฤษฎีต่าง ๆ เกี่ยวกับ CGI ที่นี้ จึงมีข้อสรุปเกี่ยวกับการเขียน CGI บน UNIX ขึ้นมาดังนี้ โปรแกรม CGI ใน UNIX นั้น ถ้าเป็น Perl เราจะต้องเขียนในรูปของ Text File ซึ่งถ้าเราใช้ Windows ในการสร้าง CGI และ Upload ขึ้นไป เราอาจจะพบปัญหา ในเรื่องของระบบไฟล์ที่ต่างกัน รหัสขึ้นบรรทัดใหม่ของไฟล์ใน UNIX นั้นจะต่างจากไฟล์ในระบบ Windows หรือ DOS เราอาจจะต้องมีการแปลงไฟล์จาก DOS (Windows) ให้เป็น UNIX

ก่อนที่จะทำการ Upload หรือเขียนด้วย Editor ที่รองรับกับการเก็บข้อมูลแบบ UNIX เช่น Text Edit Plus CGI ใน UNIX จะต้องอยู่ในโหมด 755 หรือ +x ซึ่งจะเป็นโหมดที่ไฟล์ CGI นั้นจะสามารถ Execute ได้ หากไม่มีการเปลี่ยนโหมดเป็น 755 ก็จะไม่สามารถที่จะ Run Program ได้ถ้า Server ของเรานุญาตให้เราสามารถ Telnet เข้าไปได้เราจะต้องใช้ คำสั่ง chmod ครั้นในการเปลี่ยนโหมดไฟล์ แต่ถ้าไม่อนุญาต เราจะต้องใช้โปรแกรม FTP เช่น WSFTP หรือ Cute FTP ช่วยเปลี่ยนโหมดของไฟล์ให้ไฟล์ที่ CGI จะกระทำด้วย เช่นสั่งให้ CGI เขียนข้อมูลลงไฟล์ ต้องแน่ใจว่าจะต้องอยู่ในโหมด 777, 755 เป็นต้นหรือโหมดที่อนุญาตให้สามารถเขียนได้สำหรับการ Update วิธีที่ใช้ในการทดสอบ CGI ที่ดีคือต้องหาทางทดสอบกับ Server ก่อน ในที่นี้ทดสอบ UNIX ก่อนด้วยตนเอง โดยวิธีการต่อ LAN 2 เครื่องเพื่อทดสอบ โดยเอาเครื่องคอมพิวเตอร์เก่ารุ่น 486 มาลง Linux และต่อสายแบบ Coaxial ธรรมดา ๆ 3 เมตรก็ราคาไม่ที่ร้อย , การ์ดแลนก็ 1000 บาท 2 การ์ด (ของ DLink) และก็ต่อกัน ทดลอง Telnet , FTP และทดสอบ CGI กันด้วยวิธีนี้ ก่อน Upload ขึ้นไปใช้จริง ก็ไม่มีปัญหาเลย เป็นวิธีหนึ่งที่ สำหรับ Server ที่เป็น UNIX

ถ้า Server ที่เป็น Windows NT เราจะตัดปัญหาในด้านของการเปลี่ยนโหมดไฟล์ และระบบไฟล์ที่ต่างกันของ UNIX และ DOS , Windows ครั้น เราไม่ต้องเปลี่ยนไฟล์ , ไม่ต้อง convert ไฟล์ แต่ Server ที่เป็น NT นั้นเราจะไม่สามารถ Telnet เข้าไปจัดการรายละเอียดของไฟล์ต่าง ๆ ได้ เหมือนกับใน UNIX นะครับ การจัดการเรื่องของข้อมูล ใน Windows NT รู้ ๆ กันอยู่ว่า เราสามารถเชื่อมต่อระบบฐานข้อมูลได้หลายแบบ เช่น Access โดยใช้ CGI ที่เขียนด้วยภาษา C เช่น Visual C++ , C++ Builder , Delphi , Visual Basic เป็นต้น

แต่การจัดการข้อมูลบน UNIX ก็มีอีกรูปแบบหนึ่ง ซึ่งโดยมากจะอยู่ในรูปของ Text File และใช้
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น เมื่ออ่านแล้วให้นำไปใช้ประโยชน์ด้านการค้า
 เครื่องหมายกันระหว่างพีด และจะใช้ภาษา Perl เป็นส่วนมากในการเขียน เพราะสามารถจัดการ
 ได้วากรณ์ใดๆ ทั้งสิ้น อีกทั้งยังมีโหมดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

Text File ได้ดีและเนื่องจากที่เป็น Text File จึงทำให้มีความเร็วกว่า Windows NT ใน Project นี้จึงเลือกใช้ Server เป็น UNIX เพราะต้องการความเร็วในการส่งข้อมูลสูง และไม่เน้นทางด้าน Graphic

ภาษาที่เราจะใช้เขียน CGI สำคัญมากเราต้องมาดูก่อนว่า เราสามารถใช้ภาษาใดเขียน CGI ได้บ้างและในสภาวะแวดล้อมใด ๆ เอาเฉพาะภาษาที่เอาก ๆ ทั้งนี้

บน UNIX

- Perl
- C ใช้ GNU C/C++
- TCL/TK
- Shell Script

บน Windows NT

- Perl (Perl for Win32)
- C ใช้ Visual C++ , C++ Builder
- Pascal/Delphi
- Visual Basic

การที่จะดูว่าภาษาใดสามารถเขียน CGI ได้ เราจะดูที่ความสามารถในการติดต่อกับ Standard I/O ของภาษานั้น ๆ ว่าสามารถทำได้หรือไม่ หลักการเขียน CGI นั้น จะมีหลักการเดียวกัน แต่จะต่างกันตรงที่การใช้คำสั่งในภาษา

ยกตัวอย่างเช่น ในการเขียน CGI บรรทัดแรกสุด เราจะต้องกำหนด Content-type ให้เป็น text/html ตามด้วยรหัส \n\n แต่ถ้าเราเขียนภาษา Perl เราจะเขียนดังนี้

```
print "Content-type:text/html\n\n";
print ".....";
```

แต่ถ้าเราเขียนด้วยภาษา C (ใช้ Visual C++ หรือ Turbo C) เราจะต้องเขียนดังนี้

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

จากข้างต้น เป็นการเขียน CGI บน NT จะเห็นได้ว่าทั้ง 2 โปรแกรมนั้น บรรทัดแรกจะเริ่มที่ การกำหนด content-type ให้เป็น text/html โดยใช้ Standard I/O ธรรมดา ๆ เห็นได้ชัดว่าจะใช้คำสั่งที่ต่างกันภาษา Perl และ C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างระหว่างการใช้ภาษา C และ Perl นั่นคือ ถ้าใช้ Perl ไม่ว่าจะอยู่บน UNIX หรือ Windows NT เราจะบันทึกเป็น Text File เท่านั้นที่นำไปใช้ได้เลย แต่ถ้าเขียนด้วยภาษา C เราจะต้องคอมไพล์และลิงค์ก่อน เช่น

ถ้าเราใช้ Visual C++ เราจะต้องสร้าง Application แบบ Win32 Application คือแบบ Console เท่านั้น และจะต้อง Build โปรแกรม จากนั้นก็ Upload ไฟล์ .exe ที่ได้นั้นไปใช้ บน UNIX ก็เช่นกัน

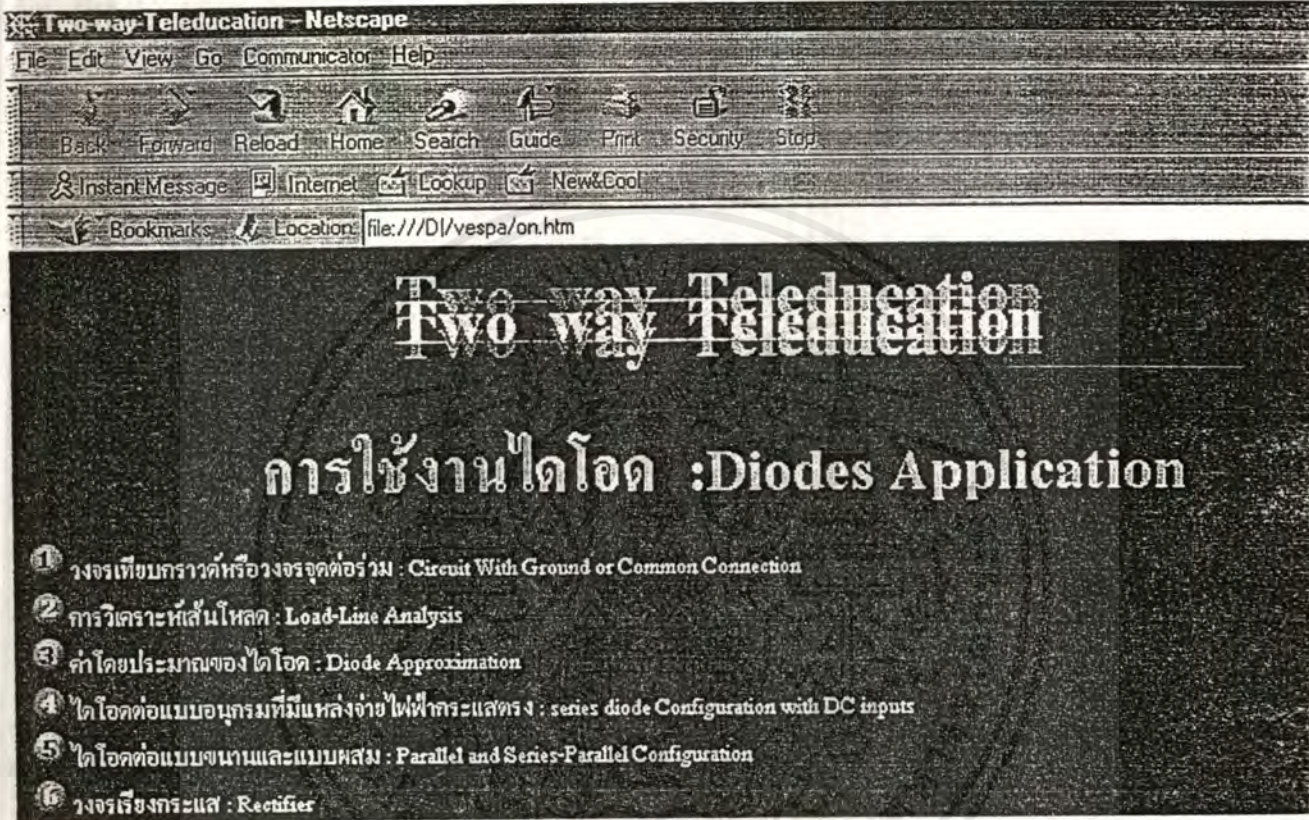
เขียนเสร็จแล้วเราจะต้องคอมไพล์ด้วยคำสั่ง cc หรือ gcc จากนั้นก็จึงนำเอาไฟล์ที่ได้ เปลี่ยนนามสกุลให้เป็น .cgi และนำไปใช้ มีภาษาไม่กี่ภาษาที่ใช้เขียน CGI ได้โดยไม่ต้องคอมไพล์คือให้เป็นไฟล์ Execute ก่อน นั่นก็คือ Perl , Tcl/Tk และ Shell Script เป็นต้น

สำหรับนามสกุลของไฟล์ที่เป็น CGI นั้น ขึ้นอยู่กับว่าโปรแกรม Web Server ที่รันอยู่จะกำหนดเป็นอะไรโดยผู้ดูแลระบบใหญ่ (บักมิน) จะเป็นผู้กำหนดตัวเลือกนี้ได้เอง ส่วนมาก บน UNIX ไฟล์ที่จะใช้เป็น CGI นั้นจะต้องเป็นไฟล์ที่มีนามสกุล .cgi (ส่วนมาก) เมื่อเราคอมไพล์ CGI ที่เขียนด้วย C ไฟล์ที่ได้จะไม่มียามสกุลเราก็อาจจะเปลี่ยนให้ไฟล์นั้นเป็น .cgi เพื่อให้เป็นที่รู้จัก และบน UNIX ไฟล์ที่เป็น CGI นั้นจะสามารถเก็บเอาไว้ได้ในทุกๆที่ ไม่จำเป็นต้องอยู่ใน Directory /Cgi-Bin อย่างเช่นที่ hypermart แต่ถ้าเป็นไฟล์ CGI บน Windows NT ถ้า Web Server ที่ใช้ IIS เราจะสามารถใช้ CGI ที่มีนามสกุลเป็น .exe , .dll (ISAPI) ได้ซึ่งถ้า Server นั้นมีการติดตั้งภาษา Perl เอาไว้ เราก็จะสามารถเขียน CGI ด้วยภาษา Perl ได้ โดยทางผู้ดูแลระบบอาจจะกำหนดให้ไฟล์ .pl หรือ .plx เป็นไฟล์ CGI ก็ได้แล้วแต่แต่ละ Server โปรแกรม CGI ที่จะรันบน NT นั้น ส่วนมากจะมี Cgi-Bin ให้กับเรา เพื่อจะสามารถกำหนด Directory ให้ execute ได้

บทที่ 8

ผลการทดลอง

จากผลการทดลองพบว่าเมื่อเข้าไปใน Program ที่เขียนเอาปรากฏภาพดังรูป

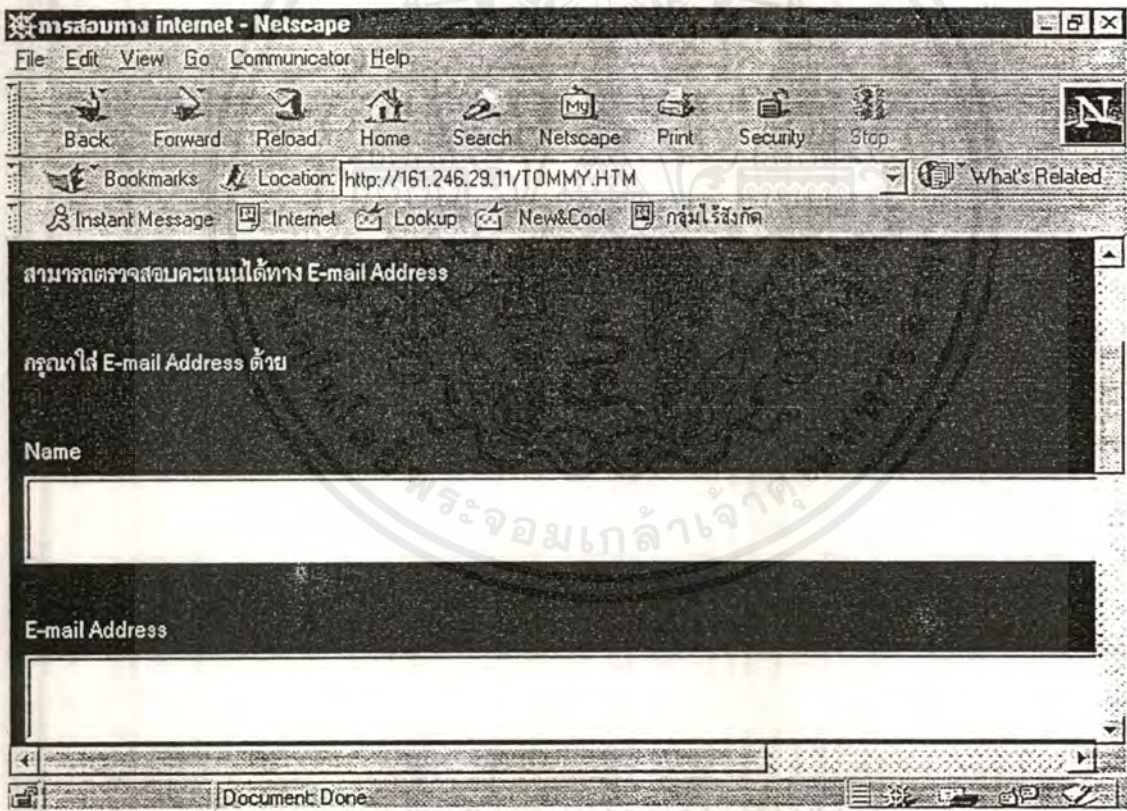


รูปที่ 8.1 แสดงการใช้งานของ Program (Two way teleducation)

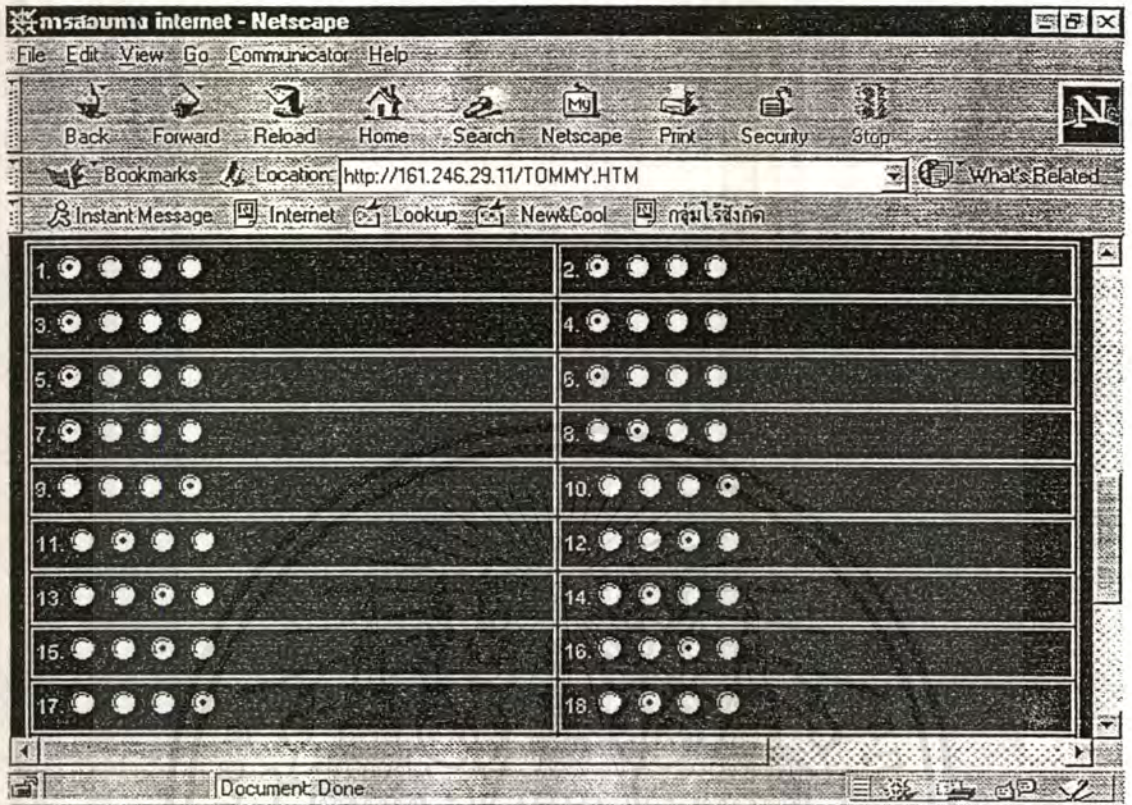
จากรูปจะเป็นตัวเนื้อหาทฤษฎีในการคำนวณไดโอดแบบต่างๆซึ่งเป็นการง่ายที่เราจะเข้าไปเปิดชม Project นี้เพราะเพียงแค่ว่ามีเครื่องคอมพิวเตอร์ต่ออยู่กับบ้านเท่านั้นเราก็สามารถที่จะเข้าไปใช้งานได้โดยไม่ต้องเสียเวลาในการเดินทางมาเรียน ที่สำคัญไม่เป็นการสิ้นเปลืองเงินด้วย เพราะว่าเราสามารถดูได้เลยจาก Browser และยังสามารที่จะทำข้อสอบเพื่อวัดผลได้อีกด้วย เมื่อเราเข้าไปดูการทำงานทั้งหมดของไดโอดแล้วเราก็มาเริ่มทำที่ตัวข้อสอบได้ โดยที่ตัวข้อสอบนี้จะใช้ภาษา Perl เป็นตัวประเมินผล โดยการที่เราเริ่มทำข้อสอบข้อสอบมีทั้งหมด 30 ข้อ เราสามารถดูข้อแรกได้โดยการที่เรา Click Mouse ไปที่หมายเลขที่ 1 ข้อสอบข้อที่ 1 ก็จะปรากฏออกมาและเมื่อเราทำข้อ 1 เสร็จแล้วให้เรากด ปิด Browser ด้วยเพราะว่าถ้าเราไม่กดปิด พอเราเปิดข้อที่ 2 ออกมาก็จะทำให้ Browser เกิดขึ้นใหม่อีกตัว ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อเราเปิดมากขึ้นจำนวน Browser ก็จะมากขึ้นตามและเมื่อจำนวน Browser มากขึ้นก็จะทำให้เครื่องคอมพิวเตอร์ของเราช้ากว่าที่เป็นไปมาก ดังนั้นจึงควรที่จะปิด Program Browser ด้วยทุกครั้งที่ทำกรเปิดข้อสอบ และเมื่อเราทำข้อสอบเสร็จแล้วก็สามารถที่จะส่งผลการสอบได้โดย Click ไปที่ปุ่ม OK จากนั้นคำตอบทั้งหมดจะถูกนำไปประมวลผลที่ Server ซึ่งที่ Server นี้เราได้ใช้ภาษา Perl เป็นตัวประเมินผลเมื่อ Perl ประมวลผลเสร็จก็จะส่ง Output ที่เป็นคะแนนออกมาที่ Browser เพื่อให้เราทราบถึงคะแนนที่เราทำไป

ส่วนที่เป็นตัวโจทย์คำถามนั้นเราได้ใช้เอกสาร HTML เป็นตัวแสดงข้อสอบที่เลือกดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 และ 8.3 เป็นรูปของตัวข้อสอบที่ใช้ในการประมวลผล

จากรูปจะพบว่ามีตัวข้อสอบอยู่ทั้งหมด 30 ข้อให้ทำแต่โดยที่แต่ละข้อเราสามารถที่จะใส่ได้เฉพาะคำตอบเท่านั้นซึ่งตัวโจทย์นั้นเราต้อง Click ไปที่เลขของข้อสอบโดยที่ถ้าเรา Click ที่หมายเลข 1 ข้อสอบข้อที่ 1 ก็จะปรากฏออกมาที่ Browser อีกตัวและถ้าต้องการทำข้อไหนก่อนเราก็สามารถ Click เลือกข้อนั้นได้เลยและเมื่อเราทำข้อสอบเสร็จหมดแล้วทุกข้อเราก็สามารถที่จะตรวจคำตอบได้โดยกดปุ่ม OK แต่ถ้าเราต้องการที่จะลบคำตอบทั้งหมดที่ทำได้โดย Click ที่ปุ่ม Reset ข้อมูลทั้งหมดที่ทำมาจะถูก Reset ใหม่หมดรวมทั้ง Name และ E-mail Address ด้วย

เมื่อทำข้อสอบเสร็จแล้วเมื่อเรา Click ที่ปุ่ม OK ข้อมูลที่ทำมาทั้งหมดจะถูกส่งไปที่ Server ที่ Directory /Cgi-Bin/ โดยที่ Directory นี้จะเก็บตัวประเมินผลไว้ในที่นี้เราใช้ภาษา Perl เป็นตัวประเมินผลเมื่อ Perl ได้รับ Data ที่ Browser ส่งมาแล้วจะทำการประเมินผลได้ทันทีแล้วส่งผลการ

ประมวลผลออกมาที่ Browser ตัวเดิมให้เราได้ทราบผลของการประมวลผล



รูปที่ 8.4 เป็นรูปการแสดงผลการประมวลผลโดยใช้ภาษา Perl โดยที่เราใส่ Name และ E-mail Addresss

จากรูปจะเห็นว่า Browser จะแสดงผลออกมาว่าให้เราใส่ Name และ E-mail Address เราจะต้องกลับไปใส่ Name และ E-mail Address ใหม่โดยที่เรา Click ที่ปุ่ม Back ที่ตัว Browser ของเราหน้าจอของ Browser ก็จะกลับไปสู่หน้าจอเก่าจากนั้นเราก็สามารถที่จะไปใส่ Name และ E-mail Address ได้ และเมื่อเรา Click ที่ปุ่ม OK เราก็จะสามารถประเมินผลการสอบได้อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยนาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

สรุปและวิจารณ์

จากการทดลองพบว่าการที่เลือกใช้ภาษา Perl เป็น Text File สามารถประมวลผลได้เร็วและยังรองรับระบบ Internet ได้ เป็นอย่างดีเพราะว่าเป็นภาษาที่เขียนไว้เฉพาะสำหรับ CGI โดยเฉพาะ เพราะว่ามีกรอกรอก Web ได้ง่ายกว่าการที่เราเลือกใช้ ASP เพราะข้อจำกัดของ ASP ก็คือจะต้องใช้ Server เป็นของ Microsoft อย่างเดียว

ส่วนภาษา Perl นั้นเราใช้ได้ทั้งของ Microsoft และ UNIX เพราะภาษา Perl นั้นได้ออกมารองรับความต้องการของผู้ที่ต้องการใช้งานของค่าย Microsoft ด้วย (Perl For Win32)

ส่วนในการที่เราเลือกใช้งานทาง WWW เพราะเราสามารถนำ Program ในระยะทางไกลๆ ได้ โดยไม่มีปัญหาไม่ว่าเราจะอยู่ที่ใดก็ตาม

ข้อดีของการส่งข้อมูลทาง Web

1. สามารถใช้กับเครื่องชนิดกันต่างระบบได้เพราะ Web Browser ถูกออกแบบมาให้ใช้งานกับเครื่องทุกประเภท และเนื่องจาก HTML และ Javascript เป็นภาษาที่ไม่ใช้การ Compile แต่จะให้ตัว Browser ทำหน้าที่แปลคำสั่งต่างๆเอง จึงมีคุณสมบัติไม่ขึ้นอยู่กับตัวเครื่อง สามารถส่งโปรแกรมไปให้เครื่องทุกเครื่องที่อยู่ในระบบเครือข่ายใช้งานได้ทันที
2. ใช้ได้โดยทั่วไปโดยไม่จำกัดขนาดขององค์การ ผู้ใช้ไม่ว่าอยู่ที่ระบบเครื่องไหนในเครือข่ายสามารถเข้าถึงแบบฟอร์มที่อยู่บนตัว Server ได้โดยไม่ต้องมีการ login เข้าไปทำการติดตั้งใดๆ และไม่จำนวนจำกัด
3. ติดต่อผู้ใช้ด้วยมาตรฐานเดียวกัน การใช้ Web form ทำให้การติดต่อกับผู้ใช้จะเป็นไปในรูปแบบเดียวกันทั้งระบบ นั่นคือจะใช้ส่วนของตัว GUI (Graphical User Interface) ของตัว Browser นั้นเอง ดังนั้นงานทุกอย่างจึงมีวิธีการใช้งานที่เป็นรูปแบบเดียวกันทั้งหมด ผู้ใช้ Browser เป็นก็สามารถใช้งานได้โดยไม่ต้องเสียเวลาอบรมผู้ใช้สำหรับการงานแต่ละอย่าง
4. สามารถใช้ Program ในระยะทางไกล (Remote Host) ได้

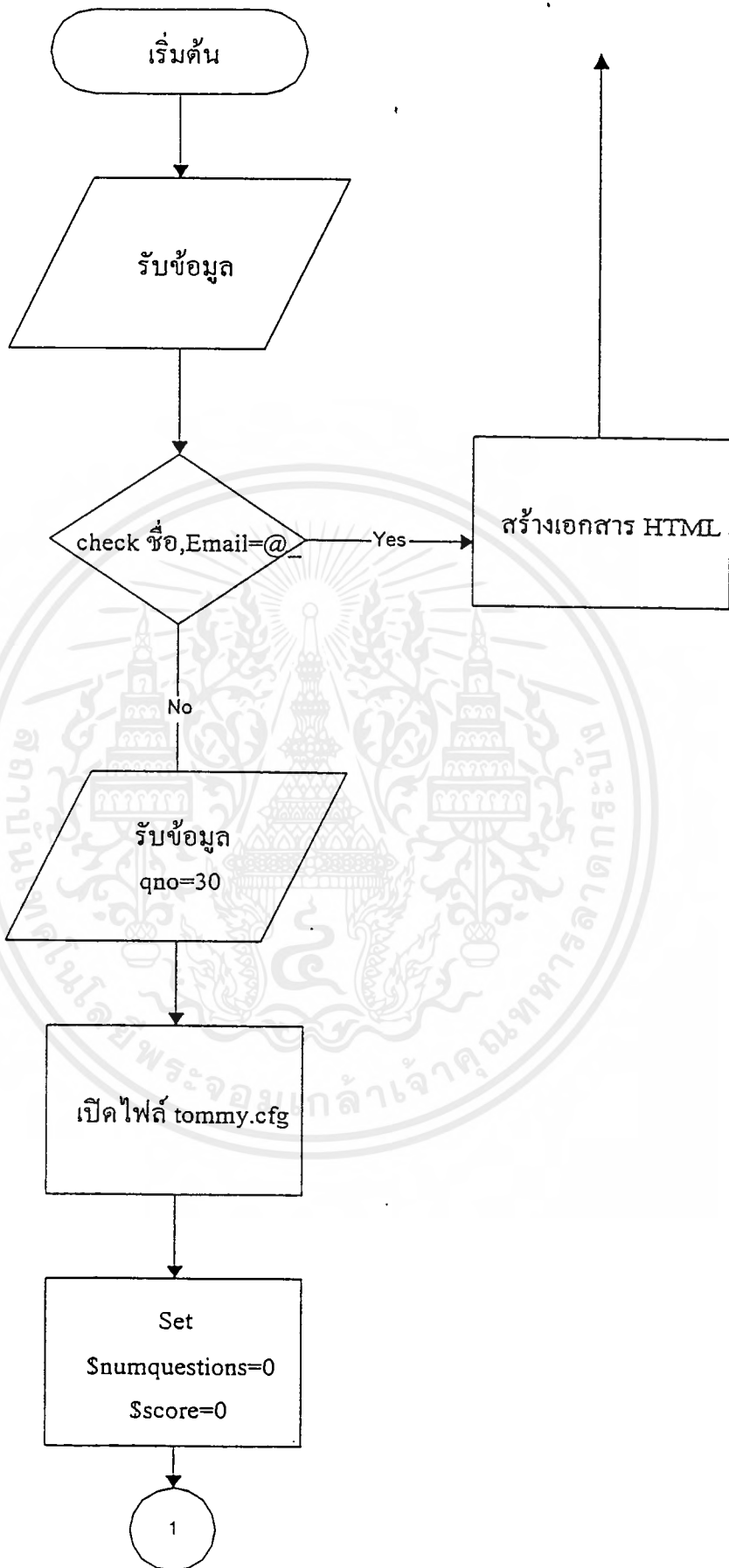
ข้อเสียในการส่งข้อมูลทาง Web

1. การส่งข้อมูลทาง Web นั้นจะช้ากว่าแบบที่เป็น Local (แบบที่อยู่กับที่) เพราะต้องใช้การดึงข้อมูลในระยะทางที่ไกลกว่า
2. บางครั้งถ้าฝ่ายผู้ส่งเกิด Down Server ขึ้นเราจะไม่สามารถรับส่งข้อมูลได้เลย
3. รูปแบบข้อการนำเสนอผ่านทาง Web น้อยกว่าแบบที่เป็น Local เพราะข้อจำกัดในการส่งของมีสูง เช่น ถ้าส่งข้อมูลรูปภาพเป็นแบบ Bitmap จะทำได้ช้ามาก

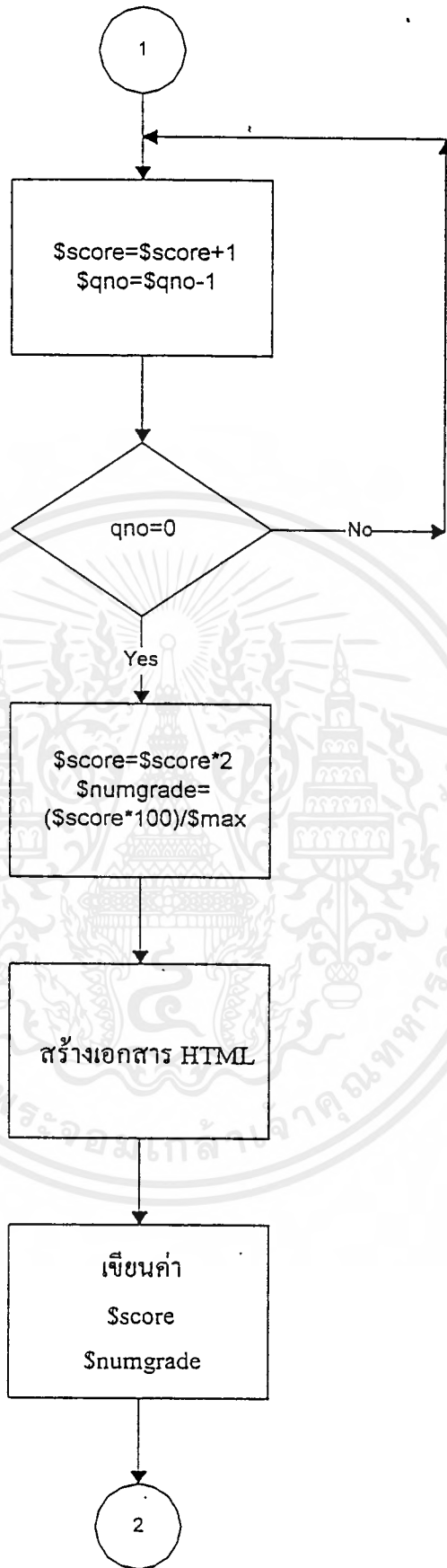
4. ในการ Run Program ที่ Execute แล้วจะทำได้ ช้าเพราะ File พวกนี้ส่วนใหญ่จะมีขนาดใหญ่ทำให้การรับส่งข้อมูลเป็นไปได้ลำบาก

ดังนั้นในการออกแบบ Web ส่วนใหญ่จะนิยมใช้ Text File , .GIF , .JPG , Check Box ฯลฯ เพราะว่าการใช้ Text File และ พวกรูปภาพเล็กๆ จะส่งได้เร็วกว่า เพราะว่าผู้ใช้ส่วนใหญ่ต้องการข่าวสารมากกว่า Graphic ที่สวยงาม ดังนั้น Project นี้จึงเน้นที่ความรวดเร็วในการรับส่งข้อมูลของข่าวสารมากกว่าที่จะมีรูปแบบที่สวยงาม , ออก Web ได้เร็วและสามารถประมวลผลได้จริงตามที่ผู้ใช้ต้องการ

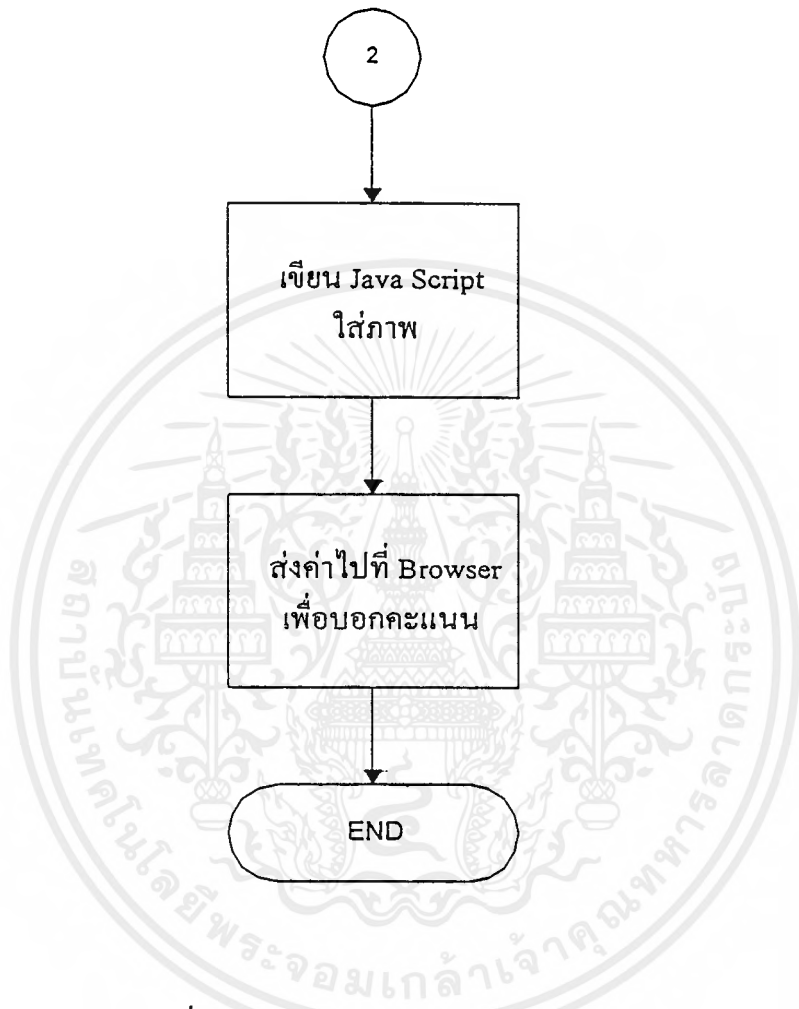




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.8 Flow Chart ของระบบ CGI

บรรณานุกรม

1. จิตเกษม พัฒนาศิริ , “เริ่มสร้างโฮมเพจด้วย HTML” ,สำนักพิมพ์ ซีเอ็ดยูเคชั่น
2. Dave Till , “Teach Yourself Perl” ,สำนักพิมพ์ SAMS PUBLISHING
3. Randal L. Schwartz, “Learning Perl” ,สำนักพิมพ์ O’Reilly & Associates, Inc
4. นิตยสาร “Internet Today” ,สำนักพิมพ์ เอ.อาร์ อินฟอร์เมชันแอนด์พับลิเคชั่น
5. นิตยสาร “Computer Time” ,สำนักพิมพ์สงวนศักดิ์การพิมพ์
6. นิตยสาร “Internet Intranet” สำนักพิมพ์ อินทราเน็ตคอม จำกัด
7. [HTTP://WWW.PERL.COM/](http://WWW.PERL.COM/)
8. [HTTP://WWW.MICROSOFT.COM/](http://WWW.MICROSOFT.COM/)
9. [HTTP://WWW.DOWNLOAD.COM/](http://WWW.DOWNLOAD.COM/)
10. [HTTP://WWW.THAIDEV.COM/](http://WWW.THAIDEV.COM/)

