ระบบบริการเชื่อมต่อข้อมูลเครือข่ายคอมพิวเตอร์ระดับโลก 1

WORLD WIDE WEB *I*

โดย

นายบุญส่ง แซ่เลี่ยง เลขประจำตัว 39013278

นายพิเศษ วินิจฉัยกุล เลขประจำตัว 39013281

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

หัวข้อปริญญานิพนธ์	ระบบบริการเชื่อมต่อข้อมูลเครือข่ายคอมพิวเตอร์ระดับโลก 1			
Project Report	WORLD WIDE WEB *I*			
ผู้จัดทำ	นายบุญส่ง	แซ่เลี่ยง	เลขประจำตัว	39013278
	นายพิเศษ	วินิจฉัยกุล	เลขประจำตัว	39013281
อาจารย์ที่ปรึกษา	อาจารย์พิทักษ์	ธรรมวาริน		
	อาจารย์มยุรี	เลิศเวชกุล		
ภาควิชา	เทคนิคอุตสาหกรรม			
คณะ	วิศวกรรมศาสตร์			
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง				
ปีการศึกษา	2541			

หัวข้อปริญญานิพนธ์	ระบบบริการเชื่อมต่อข้อมูลเครือข่ายคอมพิวเตอร์ระดับโลก 1			
Project Report	WORLD WIDE WEB *I*			
ผู้จัดทำ	นายบุญส่ง	แซ่เลี่ยง	เลขประจำตัว	39013278
	นายพิเศษ	วินิจฉัยกุล	เลขประจำตัว	39013281
อาจารย์ที่ปรึกษา	อาจารย์พิทักษ์	ธรรมวาริน		
	อาจารย์มยุรี	เลิศเวชกุล		
คณะ	วิศวกรรมศาสตร์			
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง				
ปีการศึกษา	2541			

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

ระบบบริการเชื่อมต่อข้อมูลเครือข่ายคอมพิวเตอร์ระดับโลก 1

ผู้จัดทำ	นายบุญส่ง	แซ่เลี่ยง	เลขประจำตัว	39013278
	นายพิเศษ	วินิจฉัยกุล	เลขประจำตัว	39013281
อาจารย์ที่ปรึกษา	อาจารย์พิทักษ์	ธรรมวาริน		
	อาจารย์มยุรี	เลิศเวชกุล		
ปีการศึกษา	2541			

.....

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอระบบบริการเชื่อมต่อข้อมูลเครือข่ายคอมพิวเตอร์ระดับโลก โดยมีจุดมุ่งหมายเพื่อที่จะศึกษาการใช้ภาษาและโปรแกรมต่าง ๆ ที่มีความจำเป็นในการพัฒนาระบบให้บริการเชื่อมต่อข้อมูลไปยังเครือข่ายคอมพิวเตอร์ระดับโลก โดยได้ทดลองทำ Home Page ของภาควิชาเทคนิคอุตสาหกรรมและวิศวกรรมสารสนเทศเป็นต้นแบบ โดยในโครงงานนี้จะประกอบด้วยหน้าเอกสารแบบไฮเปอร์เท็กซ์สำหรับ ประวัติภาควิชา,หลักสูตร, อาจารย์ , โครงงานวิจัยและในบางส่วนของการบริการ โดยมีการทดลองใช้ภาษา JAVA เพื่อเชื่อมโยงกับฐานข้อมูล ในการให้บริการข้อมูลผลการเรียนของรายวิชาต่าง ๆ ของภาควิชา

WORLD WIDE WEB *I*

By Mr. Boonsong Saeliang ID 39013278
Mr. Pises Winijchaikul ID 39013281

Advisor Mr. Pitak Thumwarin
Mrs. Mayuree Lertwatechakul

Academic year 1998

.....

Abstract

This thesis proposes the World Wide Web system . The objective of this project is learning of HTML and JAVA language & essential software for setting up world wide web server. This project can cerns with setting up Industrial Technology Department's web server and design home page like history of the Department's , cerriculum , lecturer and some part of service page. For subjects grade information service page , we used JAVA to connect Oracle Database also.

กิตติกรรมประกาศ

การทำโครงงานรวมทั้งการทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ก็โดยได้รับความช่วยเหลือ และได้รับคำแนะนำอย่างดียิ่ง จากท่านอาจารย์พิทักษ์ ธรรมวาริน และท่านอาจารย์มยุรี เลิศเวชกุล ซึ่งต้องรับภาระอันหนักหน่วงในการเป็นอาจารย์ที่ปรึกษาให้จนงานเสร็จสิ้น

ปริญญานิพนธ์ฉบับนี้จะสำเร็จไปไม่ได้ถ้าหากขาดเหล่าคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้จนมีความสามารถทำโครงงานนี้ได้

ด้วยเหตุนี้จึงขอขอบพระคุณบุคคลดังกล่าวข้างต้นนี้ ทั้งที่เอ่ยนามและไม่ได้เอ่ยนามเป็นอย่างสูง (ด้วยความสำนึกในพระคุณ)

คณะผู้จัดทำ

สารบัญ

บทที่ 1
บทนำ

WWW (World Wide Web) คือรูปแบบหนึ่งของระบบการเชื่อมโยงเครือข่าย ข่าวสาร ที่ใช้ในการค้นหาข้อมูลข่าวสารบนเครือข่ายอินเตอร์เน็ต จากแหล่งข้อมูลหนึ่งไปยังอีกแหล่งข้อมูลหนึ่งที่อยู่ห่างไกลออกไปให้มีความง่ายต่อการใช้งานมากที่สุด โดยจะแสดงผลอยู่ในรูปแบบของเอกสารที่เรียกว่า ไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นฐานข้อมูลชนิดหนึ่งที่ทำหน้าที่รวบรวมข้อมูลข่าวสารที่อยู่กันอย่างกระจัดกระจายไปในที่ต่างๆ ทั่วโลก ให้สามารถนำมาใช้ได้เสมือนอยู่ในที่เดียวกันคล้ายกับเส้นใยที่ถักทอเส้นสายเชื่อมโยงกันไปมา แม้ว่าจะมีเส้นใยเป็นจำนวนมาก แต่ละเส้นก็จะถูกจัดวางทับกัน โดยจะมีจุดเชื่อมต่อที่ทำให้ตัวแมงมุมสามารถที่จะเดินไปยังจุดใดๆ บนเส้นใยเหล่านี้ได้

1.1 จุดประสงค์ของโครงงาน

1. เพื่อศึกษาพื้นฐานและลักษณะทั่วไปของ WWW
2. เพื่อศึกษาการติดตั้งระบบ WWW เซิร์ฟเวอร์ และวิเคราะห์ถึงปัจจัยในการตัดสินใจเลือกใช้ระบบให้บริการ เซิร์ฟเวอร์ ตลอดจนการอำนวยการบริหาร เซิร์ฟเวอร์
3. เพื่อศึกษาหลักการทำงานของระบบสื่อสาร HTTP และมาตรฐานการรับส่ง FTP
4. เพื่อศึกษาการรักษาความปลอดภัยของ เซิร์ฟเวอร์ ในระบบ WWW
5. เพื่อศึกษาซอร์ฟแวร์ที่ใช้ในการสร้างเอกสาร ไฮเปอร์เท็กซ์ ซึ่งมีความจำเป็นอย่างมากในระบบ WWW
6. เพื่อทำให้ภาควิชาเทคนิคอุตสาหกรรม และวิศวกรรมเทคโนโลยีสารสนเทศได้มี Home Page ที่จะใช้ในการประชาสัมพันธ์ข้อมูลข่าวสารให้ออกกว้างสู่สายตาสำหรับผู้ที่สนใจและผู้ที่เกี่ยวข้องทั่วไป โดยข้อมูลข่าวสารนี้สามารถที่จะพัฒนาให้เป็นข้อมูลที่เป็นปัจจุบันอยู่เสมอ

1.2 ขอบข่ายของโครงงาน

ในหัวข้อโครงงาน www นี้ได้เป็นการนำเอาความรู้ทางด้านคอมพิวเตอร์ และเทคโนโลยีที่ได้จากการศึกษาพร้อมทั้งค้นคว้าเพิ่มเติมในเรื่องของเครือข่ายเวิล์ดไวด์เว็บ บนการทำงานของระบบอินเตอร์เน็ต โดยขอบข่ายของโครงงานนี้จะประกอบด้วย

1. การ Setup Server
2. การเขียน Home page ของภาควิชาเทคนิคอุตสาหกรรม และวิศวกรรมสารสนเทศ

บทที่ 2
พื้นฐานในการศึกษาและลักษณะทั่วไปของ WWW

2.1 ประวัติความเป็นมาของ WWW

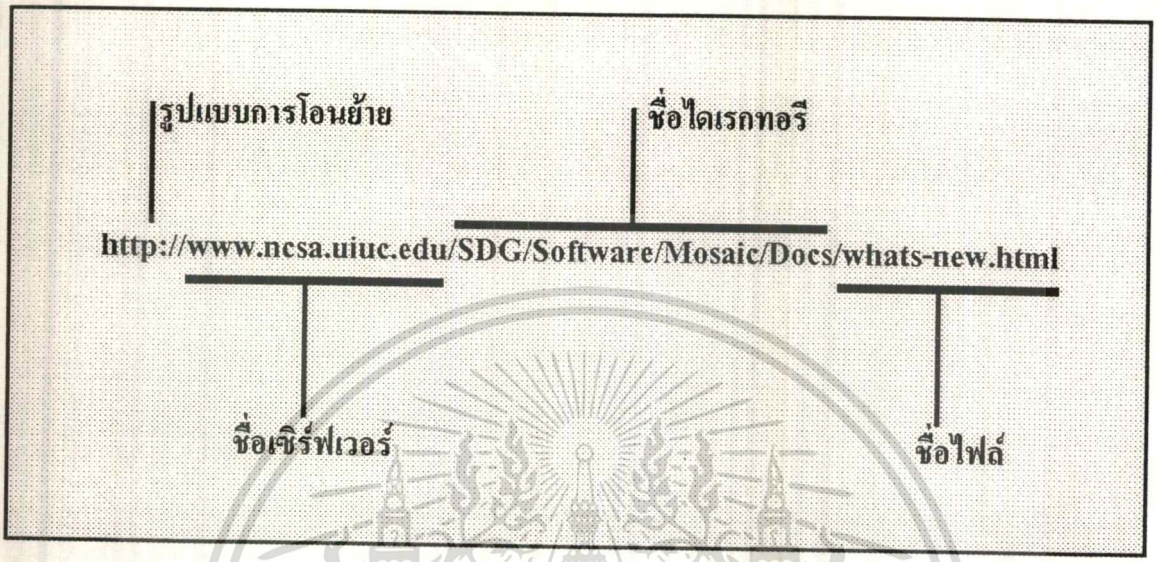
WWW (World Wide Web) เป็นระบบสืบค้นข้อมูลที่ได้รับการพัฒนาและริเริ่มเมื่อปี พ.ศ. 2533 โดย ทิม เบิร์นเนอร์ส-ลี (Tim Berners-Lee) และ โรเบิร์ต ไกล์เลีย (Robert Cailliau) นักวิทยาศาสตร์ของสถาบัน CERN ซึ่งเป็นห้องปฏิบัติการทางฟิสิกส์แห่งยุโรป (European Particle Physics Laboratory) ตั้งอยู่ที่นครเจนีวา ประเทศสวิตเซอร์แลนด์ โดยมีวัตถุประสงค์เพื่อการสื่อสารข้อมูลคอมพิวเตอร์ผ่านทางเครือข่ายอินเตอร์เน็ต โดยสามารถสื่อสารได้หลายรูปแบบ เช่น ข้อมูลที่เป็นไฟล์กราฟฟิกซึ่งอาจจะเป็นรูปภาพหรือข้อความ และไฟล์ข้อมูลที่เป็นเสียง ดังนั้นในตัวคอมพิวเตอร์จึงจำเป็นที่จะต้องได้รับการติดตั้งระบบที่เรียกว่า " มัลติมีเดีย " (Multimedia)

เว็บเป็นระบบเครือข่ายไคลเอนต์เซิร์ฟเวอร์ประเภทหนึ่ง คอมพิวเตอร์ที่ทำหน้าที่เก็บรวบรวมข้อมูลเรียกว่า " เซิร์ฟเวอร์ " (Server) ส่วนคอมพิวเตอร์ที่ทำการแสดงผลของข้อมูลเรียกว่า " บราวเซอร์ " (Browser) หรือ " เว็บบราวเซอร์ " (Web Browser) และถ้าต้องการที่จะใช้บริการเวิลด์ไวด์เว็บนั้นก็จะต้องมีซอร์ฟแวร์เฉพาะที่ใช้สำหรับเว็บเซิร์ฟเวอร์และเว็บบราวเซอร์ เช่น Netscape Navigator หรือ Microsoft Internet Explorer เป็นต้น ซึ่งเว็บนี้จะใช้ระบบ Hypertext ซึ่งเป็นระบบที่อนุญาตให้สามารถเข้าถึงข้อมูลได้ด้วยการเชื่อมโยงกันระหว่างข้อมูลแบบไม่มีการจัดเรียงลำดับหรือทำเป็นรายการไว้ การที่จะเข้าถึงข้อมูลได้จะต้องใช้วิธีการเชื่อมโยงกันระหว่างข้อมูล

2.2 ซอร์ฟแวร์สำหรับระบบ WWW

โปรแกรมสำหรับแสดงข้อมูลจากแหล่งต่างๆ โดยการเลือกเชื่อมโยงกับ WWW เซิร์ฟเวอร์ ตามรายการที่ปรากฏบนซอร์ฟแวร์ ซึ่งเรียกว่าโปรแกรม www browser โปรแกรมประเภทนี้ได้แก่โปรแกรม NCSA Mosaic โปรแกรม Cello และโปรแกรม Netscape สำหรับการเชื่อมโยงข้อมูลของโปรแกรมเหล่านี้ได้ถูกกำหนดให้เป็นแบบการสืบ

ค้นแหล่งข้อมูลเรียกว่า URL (Umiform Resource Locators) โดยมีรูปแบบดังตัวอย่างที่ได้แสดงในรูปที่ 1.1



รูปที่ 1.1 แสดงรูปแบบของรหัสสืบค้นข้อมูล URL

รูปแบบของ URL เป็นรูปแบบมาตรฐานสำหรับระบบ WWW โดยกำหนดให้ขึ้นต้นด้วยคำว่า HTTP (Hypertext Transfer Protocol) ซึ่งหมายถึงการโต้ตอบเพื่อการสื่อสารกันแบบ ไฮเปอร์เท็กซ์ สำหรับคำจำกัดความของไฮเปอร์เท็กซ์คือ คำหรือวลีที่ประกอบด้วยคำอธิบายซ่อนอยู่เบื้องหลัง หากใช้เมาส์ดับเบิลคลิกไปที่คำหรือวลีนั้น ๆ ก็จะปรากฏคำอธิบายหรือรายละเอียดของคำหรือวลีดังกล่าว ซึ่งคำอธิบายที่ปรากฏขึ้นนั้นได้มาจากการเชื่อมโยงไฟล์ข้อมูลจากแหล่งต่าง ๆ ซึ่งอาจจะเป็นข้อมูลบนคอมพิวเตอร์ของผู้ใช้เอง หรือข้อมูลจากเซิร์ฟเวอร์อื่น ๆ สำหรับรูปแบบ URL ที่แสดงในรูปที่ 1.1 นั้นมีความหมายว่าโปรแกรมจะทำการเชื่อมโยงข้อมูลโดยระบบโต้ตอบข้อมูลตามมาตรฐานของ WWW ซึ่งถูกกำหนดด้วยเครื่องหมาย http:// โดยเชื่อมโยงเข้ากับ WWW เซิร์ฟเวอร์ชื่อ www.ncsa.uiuc.edu เพื่อการโอนย้ายข้อมูลจากไฟล์ชื่อ whats-new.html ซึ่งอยู่ในไดเรกทอรี /SDG /Software /Mosaic /Docs /

2.3 การใช้งาน World Wide Web (Web)

Web คือระบบสืบค้นสำหรับการติดต่อสื่อสารข้อมูลบนอินเตอร์เน็ต (Internet) เมื่อใช้กับโปรแกรมประยุกต์ที่เรียกว่าโปรแกรมขอใช้บริการ หรือ Web browser ก็จะสามารถใช้

เอกสารแบบมัลติมีเดียของ Web แบบไฮเปอร์ลิงก์ (Hyperlink) เรียกว่าหน้าเอกสาร (pages) หรือแหล่งข้อมูล (site) ซึ่งเก็บอยู่บนคอมพิวเตอร์ทั่วโลกที่เชื่อมต่อเข้ากับระบบอินเตอร์เน็ต เนื่องจาก Web ได้ใช้ความสามารถทางด้านมัลติมีเดียจึงทำให้ดูเหมือนว่า Web ได้ทำให้เกิดการปฏิวัติในสื่อทางด้านสิ่งพิมพ์ จากเดิมเป็นหนังสือ นิตยสาร วารสาร หรือหนังสือพิมพ์ที่ต้องพิมพ์ลงบนกระดาษ แต่เมื่อใช้ Web สื่อต่างๆ เหล่านี้ก็จะเปลี่ยนมาเป็นสิ่งพิมพ์ทางอิเล็กทรอนิกส์ (Electronics) ที่ประกอบด้วยสิ่งต่าง ๆ มากขึ้น เช่นมีภาพเคลื่อนไหว มีเสียงประกอบ และรวมถึงการค้นหาข้อมูลที่ต้องการก็สามารถทำได้ง่ายขึ้นด้วย

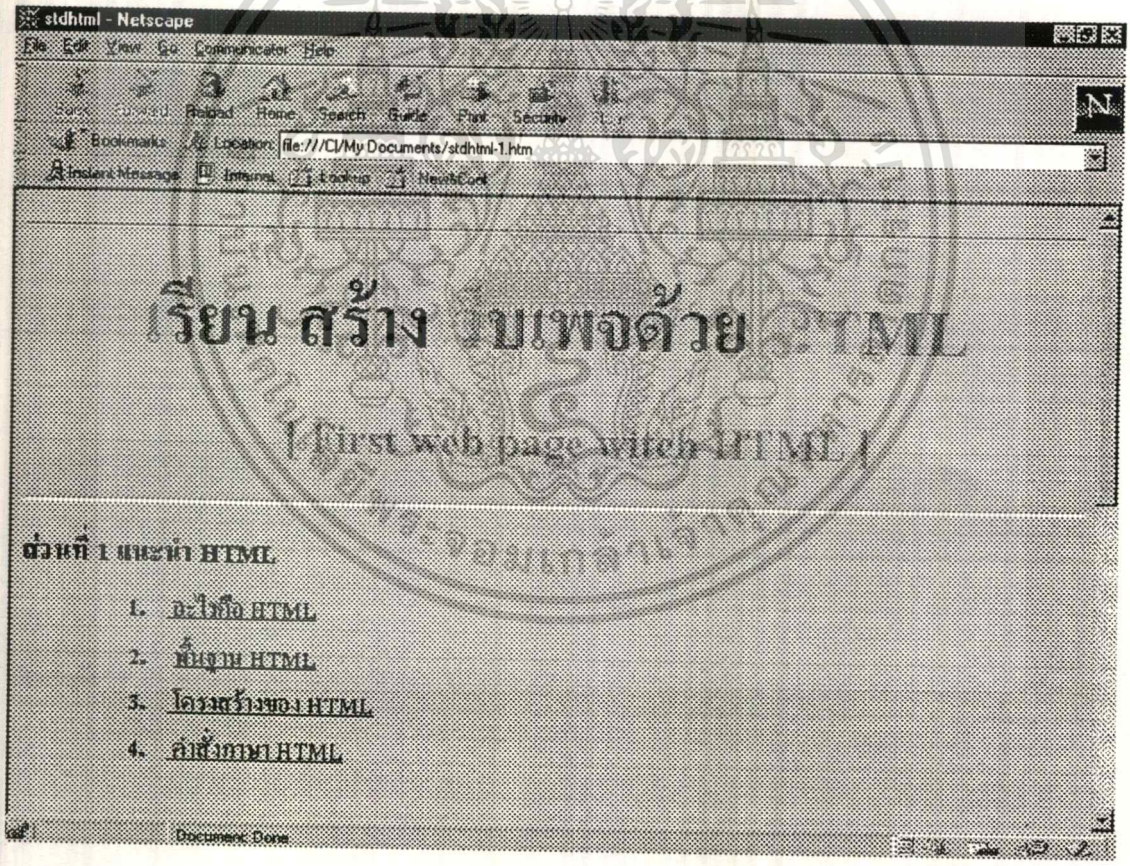
2.3.1 หลักการทำงานของ WWW

จุดมุ่งหมายสำคัญอย่างหนึ่งของการเกิดระบบ World Wide Web ก็คือการทำให้โปรโตคอล (Protocol) ต่าง ๆ ที่มีอยู่ในปัจจุบัน สามารถเข้าไปใช้งานได้ในรูปแบบมาตรฐานเดียวกัน และรูปแบบมาตรฐานนั้นก็คือ URL (Uniform Resouce Locator) ซึ่งทำให้ผู้ใช้สามารถจะกำหนดเจาะจงไปถึงสิ่งใด ๆ ก็ได้ในระบบอินเตอร์เน็ต รวมทั้งยังมีข้อมูลเพียงพอที่จะไปดึงเอาข้อมูลนั้นมาใช้ โปรโตคอลที่สามารถใช้ได้กับ URL ได้แก่ FTP (File Transfer Protocol) เว็บ เซิร์ฟเวอร์นั้นถูกออกแบบมาให้จัดการได้กับเอกสารที่เขียนด้วยภาษา HTML (Hypertext Markup Language) เหตุที่ต้องเป็นอย่างนั้นก็เนื่องจากมันมีความสามารถในการเชื่อมต่อกับเอกสารอื่ น ๆ ในสถานที่ต่าง ๆ จากภายในเอกสารนั้นทำได้ง่ายและรวดเร็ว และสิ่งนี้ก็คือคุณสมบัติที่ทำให้ WWW เป็นระบบที่ได้รับความนิยมเป็นอย่างสูง นอกจากนี้แล้ว HTML ยังสามารถสนับสนุนการอ้างถึง Object ต่าง ๆ ที่อยู่ภายนอกเอกสารอีกด้วย เช่น ภาพ เสียง หรือแม้แต่ภาพเคลื่อนไหวด้วย

ระบบ www ทำงานโดย Protocol HTTP ซึ่งสนับสนุนการทำงานแบบ Client/Server Protocol ซึ่ง HTTP นี้ปัจจุบันยังคงผูกตัวเองติดอยู่กับการทำงานบนระบบเครือข่ายในแบบ ทีซีพี/ไอพี (TCP/ IP) ซึ่งเป็นโปรโตคอลพื้นฐานของระบบอินเตอร์เน็ต เมื่อใช้โปรแกรม Web browser เรียกใช้ข้อมูลข่าวสารจาก Web server ที่ต่อเชื่อมอยู่ในอินเตอร์เน็ต ตัวเครื่องคอมพิวเตอร์และตัวโปรแกรม browser ก็จะทำงานเป็น Client และเครื่องคอมพิวเตอร์ที่ให้บริการข้อมูลก็จะทำหน้าที่เป็นผู้ให้บริการหรือเรียกว่า Server โดยระบบ www นี้จะให้บริการข้อมูลโดยผ่านโปรโตคอลแบบ HTTP ซึ่ง Client จะเริ่มเปิดการเชื่อมต่อกับผู้ให้บริการ Server โดยส่งคำร้องขอไปเพียงครั้งเดียวแล้วรอรับการตอบสนอง จากนั้นการเชื่อมต่อก็จะถูกปิดลงไป

2.3.1.1 การใช้งาน Netscape

โปรแกรม Netscape Navigator ถือได้ว่าเป็นโปรแกรมเว็บบราวเซอร์ที่ปฏิวัติวงการอินเตอร์เน็ตเลยก็ว่าได้ เนื่องจากในปัจจุบันนี้มีผู้ใช้มากกว่า 70 เปอร์เซนต์ของเว็บบราวเซอร์ที่มีใช้อยู่ทั้งหมด ซึ่งโปรแกรม Netscape Navigator นี้สามารถติดตั้งได้ง่าย โดยการดาวน์โหลดจากเว็บไซต์ของเน็ตเคป (http://www.netscape.com /) หรืออาจจะติดตั้งได้จากซีดีรอมที่มีโปรแกรม Netscape Navigator อยู่ โดยการคลิกเมาส์ไปที่ตัวโปรแกรมจากนั้นก็เพียงทำตามคำแนะนำที่ปรากฏบนหน้าจอไปเรื่อย ๆ จนเสร็จสิ้น แล้วก็ทำการรีสตาร์ตเครื่องใหม่เพื่อให้คอมพิวเตอร์ได้กำหนด Configuration ต่าง ๆ ได้โดยอัตโนมัติ



รูปที่ 1.2 แสดงตัวอย่างหน้าจอของ Netscape Navigator

หลังจากที่ทำการติดตั้งและกำหนดค่าต่าง ๆ ของ Netscape Navigator เสร็จสมบูรณ์แล้วนั้นก็จะได้พบกับไอคอนของ Netscape Communicator อยู่บนเดสก์ทอปในหน้าจอของเครื่องคอมพิวเตอร์ดังรูป

ในการเริ่มต้นการทำงานของ Netscape Navigator ก็เพียงแต่ดับเบิลคลิกไปที่ไอคอน Netscape Communicator เท่านั้นโปรแกรมก็จะค้นหาการเชื่อมต่ออินเตอร์เน็ต และถ้าหากคอมพิวเตอร์ได้เชื่อมต่อกับอินเตอร์เน็ตอยู่แล้ว หน้าจอก็จะภาพดังรูปที่ 1.2 ตามที่ได้แสดงไปแล้ว แต่ถ้าหากยังไม่ได้เชื่อมต่ออยู่กับอินเตอร์เน็ต โปรแกรมก็จะเรียกโปรแกรมที่ใช้ในการเชื่อมต่อกับอินเตอร์เน็ตขึ้นมาให้ เช่น Dial-up Networking เพื่อมาหมุนโทรศัพท์เชื่อมต่อกับอินเตอร์เน็ตก่อน

ความหมายของไอคอนต่าง ๆ บนหน้าจอ Netscape

Back	กลับไปยังเอกสารเว็บที่เคยไปมาแล้ว ก่อนหน้าที่อยู่ในขณะนี้ 1 หน้า
Forward	ไปยังเอกสารเว็บที่เคยไปมาแล้ว ที่อยู่ข้างหน้าของหน้าที่อยู่ในขณะนี้ 1 หน้า
Reload	ดาวน์โหลดเอกสารเว็บที่เปิดอยู่ในขณะนั้นใหม่ เพื่อนำเอาข้อมูลที่อัพเดตล่าสุดมาแสดงในจอภาพ
Home	ไปยังเอกสารเว็บหน้าเริ่มต้นตามที่ได้กำหนดไว้ (Start page)
Search	ค้นหาข้อมูลเอกสารเว็บตามที่ต้องการ
Guide	แนะนำเอกสารเว็บที่หน้าสนใจ
Print	พิมพ์เอกสารเว็บ ที่แสดงอยู่ในขณะนั้น
Security	แสดงข้อมูลแนะนำในเรื่องความปลอดภัยของเอกสารเว็บ

Stop หยุดการดาวน์โหลดเอกสารเว็บที่ดาวน์โหลดอยู่ในขณะนั้น โดยในขณะที่ดาวน์โหลดอยู่จะมีสีแดง เมื่อดาวน์โหลดเสร็จแล้วหรือหยุดดาวน์โหลดแล้วจะเป็นสีขาวดำตามเดิม

2.3.1.2 การใช้ไฮเปอร์ลิงก์ (Hyperlink)

ในเอกสารของเว็บหรือเอกสาร HTML จะมีลิงก์ที่เชื่อมต่อไปยังหน้าเว็บเพจอื่น ๆ ซึ่งลิงก์ที่เชื่อมต่อเหล่านี้ จะเรียกว่า ไฮเปอร์ลิงก์ โดยไฮเปอร์ลิงก์นี้ในบางครั้งอาจจะเป็นรูปภาพ บางครั้งอาจจะเป็นตัวอักษร ซึ่งจะสังเกตได้จากที่พอยน์เตอร์ของเมาส์จะเปลี่ยนไปเป็นรูปมือที่สามารถกดแล้วไปยังเอกสารเว็บอื่นต่อได้

ลิงก์เหล่านี้ยังแบ่งเป็น 2 ประเภท คือ ลิงก์ที่เคยไปมาแล้ว (Visited Link) และลิงก์ที่ยังไม่เคยไป (Unvisited Link) โดยปกติแล้วหากเป็นลิงก์ที่เป็นตัวอักษรก็จะแสดงสีออกมาแตกต่างกัน ซึ่งค่าดีฟอลต์ของ Netscape Navigator ลิงก์ที่เคยไปมาแล้วจะเป็นสีม่วง ส่วนลิงก์ที่ยังไม่เคยไปจะเป็นสีน้ำเงิน แต่ในเว็บไซต์บางแห่งอาจจะมีการกำหนดสีของลิงก์ที่เคยไปและลิงก์ที่ยังไม่เคยไปแตกต่างจากค่าดีฟอลต์ได้ด้วย

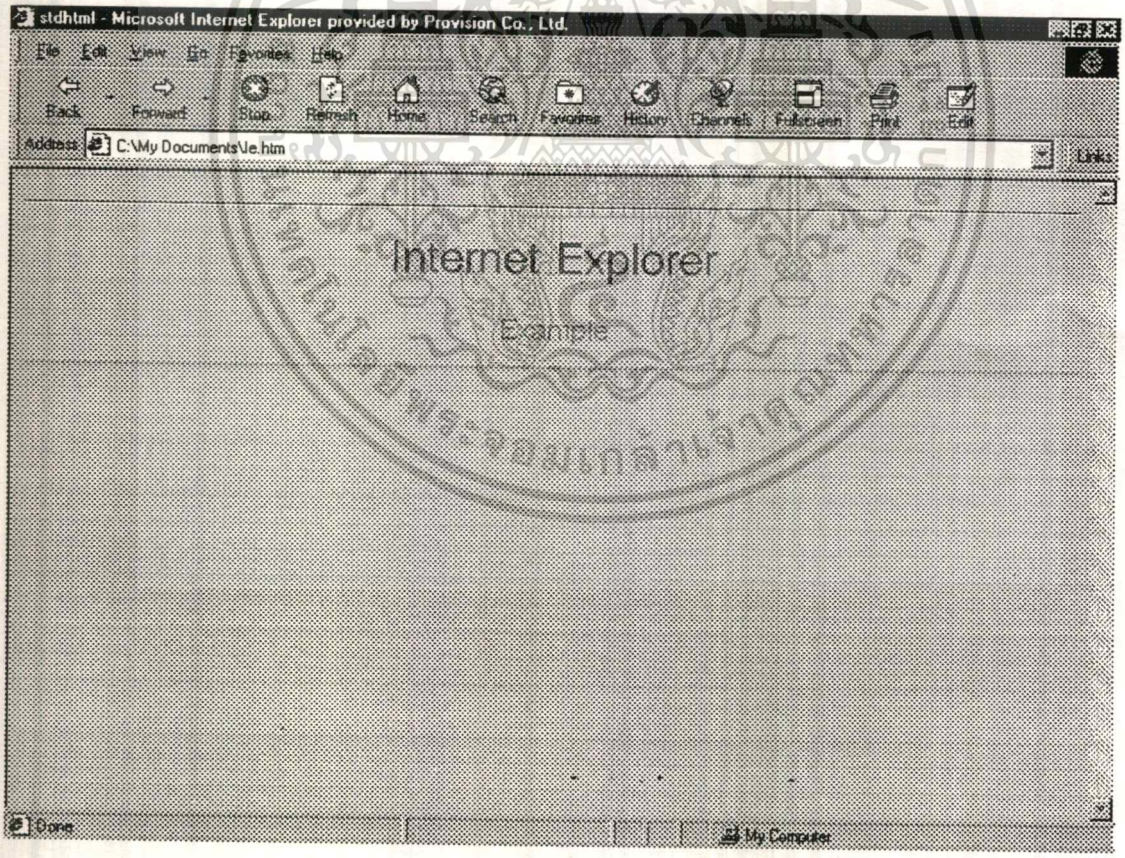
2.3.1.3 การใช้งาน Internet Explorer

โปรแกรม Internet Explorer เป็นโปรแกรมเว็บบราวเซอร์ที่ยังได้รับความนิยมอีกตัวหนึ่ง เนื่องจากเป็นโปรแกรมเว็บบราวเซอร์ที่ให้ประสิทธิภาพด้านการทำงานต่าง ๆ ครบครัน ไม่ว่าจะเป็นการท่องเที่ยวในเว็บไซต์ต่างๆบนอินเตอร์เน็ต การรับส่งจดหมายอิเล็กทรอนิกส์ การอ่านประกาศข่าวต่าง ๆ ในนิวส์กรุป พร้อมกันนั้นยังมีเทคโนโลยี ActiveX ที่ช่วยเพิ่มประสิทธิภาพในการทำงานมากยิ่งขึ้น โดยเฉพาะอย่างยิ่งในด้านระบบมัลติมีเดีย

การติดตั้งโปรแกรม Internet Explorer ก็ทำได้โดยดาวน์โหลดจากเว็บไซต์ของไมโครซอฟท์ (http://www.microsoft.com/ie/) หรือติดตั้งได้จากซีดีรอมที่มีโปรแกรม Internet Explorer อยู่ ซึ่งการติดตั้งก็เพียงแต่ปฏิบัติตามคำสั่งที่ปรากฏตามหน้าจอไปเรื่อย ๆ จนเสร็จ แล้วรีสตาร์ตเครื่องคอมพิวเตอร์ใหม่ จากนั้นก็เซตค่าต่าง ๆ ตามคุณสมบัติที่เป็นอยู่ก็เป็นอันเสร็จสิ้นสมบูรณ์

หลังจากที่ทำการติดตั้งและกำหนดค่าต่าง ๆ ของ Internet Explorer เสร็จสมบูรณ์แล้ว นั้นก็จะได้พบกับไอคอนของ Internet Explorer อยู่บนเดสก์ทอปในหน้าจอของเครื่องคอมพิวเตอร์ดังรูป

ในการเริ่มต้นการทำงานของ Internet Explorer ก็เพียงแต่ดับเบิลคลิกไปที่ไอคอน Internet Explorer เท่านั้นโปรแกรมก็จะเริ่มค้นหาการเชื่อมต่ออินเตอร์เน็ต และถ้าหากคอมพิวเตอร์ได้เชื่อมต่อกับอินเตอร์เน็ตอยู่แล้ว หน้าจอก็จะภาพดังรูปที่ 1.3 ตามที่ได้แสดงไปแล้ว



รูปที่ 1.3 แสดงตัวอย่างหน้าจอของ Internet Explorer

ความหมายของไอคอนต่าง ๆ บนหน้าจอ Netscape

Back	กลับไปยังเอกสารเว็บที่เคยไปมาแล้ว ก่อนหน้าที่อยู่ในขณะนี้ 1 หน้า
Forward	ไปยังเอกสารเว็บที่เคยไปมาแล้ว ที่อยู่ข้างหน้าของหน้าที่อยู่ในขณะนี้ 1 หน้า
Stop	หยุดการดาวน์โหลดเอกสารเว็บที่ดาวน์โหลดอยู่ในขณะนั้น โดยในขณะที่ดาวน์โหลดอยู่จะมีสีแดง เมื่อดาวน์โหลดเสร็จแล้วหรือหยุดดาวน์โหลดแล้วจะเป็นสีขาวดำตามเดิม
Refresh	ดาวน์โหลดเอกสารเว็บที่เปิดอยู่ในขณะนั้นใหม่ เพื่อนำเอาข้อมูลที่อัพเดตล่าสุดมาแสดงในจอภาพ
Home	ไปยังเอกสารเว็บหน้าเริ่มต้นตามที่ได้กำหนดไว้ (Start page)
Search	ค้นหาข้อมูลเอกสารเว็บตามที่ต้องการ
Favorites	แนะนำเอกสารเว็บที่หน้าสนใจ
Print	พิมพ์เอกสารเว็บ ที่แสดงอยู่ในขณะนั้น

2.3.1.4 ข้อเปรียบเทียบระหว่าง Netscape Navigator และ Internet Explorer

นับตั้งแต่ปี พ.ศ. 2539 เป็นต้นมาอินเตอร์เน็ตได้เข้ามามีบทบาทสำคัญในวงการทั่ว ๆ ไปมากยิ่งขึ้น พร้อมกันนั้นผู้คนทั่วไปก็สามารถสมัครเป็นสมาชิกของอินเตอร์เน็ตได้ง่ายและรวดเร็วขึ้น เครือข่าย www ในปัจจุบันและอีกต่อไปในอนาคตจะไม่ใช่เป็นเพียงโฮมเพจ หรือเป็นเว็บไซต์ที่มีแต่ภาพนิ่ง กับข้อความที่เป็นตัวอักษรแบบไฮเปอร์เท็กซ์ ให้ดูเพียงอย่างเดียวอีกต่อไปแล้ว เพราะด้วยความเร็วในการรับส่งข้อมูลที่สูงขึ้นทุกวันนี้ทำให้สามารถที่จะส่งข้อมูลในทุกรูปแบบผ่านทางเว็บได้ง่ายขึ้น ไม่ว่าจะเป็นภาพเคลื่อนไหว (Animation) หรือภาพยนต์พร้อมด้วยเสียงที่สมจริงสมจังก็ยังสามารถที่จะส่งได้ สำหรับในตอนนี้นั้นกระแสของอินเตอร์เน็ตยังให้ความสำคัญมุ่งสู่การพัฒนาเครือข่าย www อยู่ โดยจะมีการแข่งขันกันทางด้านในส่วนที่เป็นโปรแกรมเซิร์ฟเวอร์ และในส่วนที่เป็นโปรแกรมบราวเซอร์ ซึ่งในส่วนของโปรแกรมบราวเซอร์นั้นมีที่นิยมใช้กันมากก็คือ Netscape และ Internet Explorer ซึ่งทาง

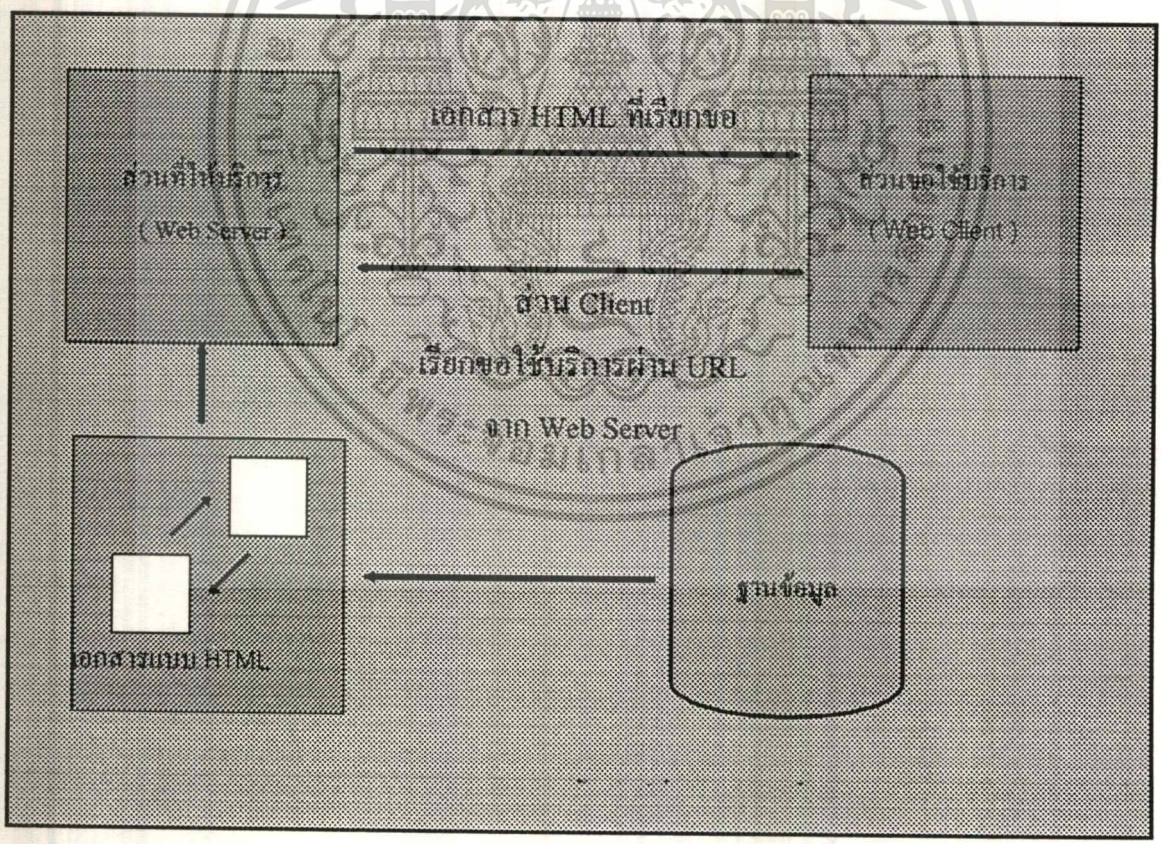
ด้าน Netscape Navigator นั้นได้อาศัยจุดเด่นของการเป็นผู้นำมาตรฐานใหม่ ๆ อยู่เสมอ รวมทั้งความเร็วในการดูข้อมูลที่รวดเร็วกว่าด้วย โดยร่วมมือกับทางบริษัท Sun Microsystem ผู้ผลิตจาวา (Java) ภาษาในการพัฒนาแอพพลิเคชัน (Application) บนเว็บเพื่อที่จะนำเอาภาษาจาวามาเป็นส่วนหนึ่งของโปรแกรม Web browser ซึ่งปัจจุบันก็สามารถสนับสนุนการทำงานของจาวาได้อย่างเต็มรูปแบบแล้ว นอกจากนี้ Netscape ยังเปิดโอกาสให้บรรดาผู้ผลิตซอฟท์แวร์รายอื่น ๆ สามารถที่จะทำการเพิ่มเติม Plugin เข้ากับ Netscape ได้เพื่อทำให้ Netscape มีความสามารถเพิ่มมากขึ้น

ส่วนทางด้าน Internet Explorer ของ Microsoft นั้น ก็ได้ยึดความสามารถในการทำการเชื่อมต่อ (Interface) กับผลิตภัณฑ์มาตรฐานหลาย ๆ ตัวของไมโครซอฟท์เอง เช่น Visaul Basic , Windows , Windows NT , Microsoft office และอื่น ๆ อีกมากมาย เป็นต้น

บทที่ 3
การติดตั้งระบบ WWW เซิร์ฟเวอร์

3.1 ลักษณะทั่วไปของ Web Server

ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ในระบบเครือข่ายนั้น เหตุผลที่ต้องมีการเชื่อมต่อก็เพื่อที่จะได้รับรายละเอียดข้อมูลข่าวสารจากอีกฝ่ายหนึ่ง โดยลักษณะของข้อมูลข่าวสารที่มีการแลกเปลี่ยนกันนั้นอาจจะอยู่ในรูปแบบของจดหมายอิเล็กทรอนิกส์ (E- mail) และการโอนย้ายแฟ้มข้อมูล (File Transfer) ในระยะหลังได้มีบริการต่าง ๆ เพิ่มขึ้นมาเพื่อรองรับ



รูปที่ 3.1 แสดงรูปแบบการทำงานของบริการ World Wide Web ซึ่งประกอบด้วย ส่วนให้บริการ (Server) และส่วนขอใช้บริการ (Client)

การใช้งานที่เพิ่มมากขึ้นนี้ด้วย โดยมีระบบช่วยในการจัดเก็บแฟ้มข้อมูลและรายละเอียดอย่างเป็นหมวดหมู่พัฒนาขึ้นมา เมื่อจำนวนข้อมูลมีเพิ่มมากขึ้นความต้องการในการสืบค้นหรือเรียกคืนข้อมูลก็ต้องมีขึ้นมาสำหรับใช้งาน เมื่อความสามารถของระบบเพิ่มมากขึ้นประกอบกับเทคโนโลยีของการรับส่งข้อมูลที่ดี จึงมีผู้ที่เล็งเห็นว่าควรที่จะพัฒนารูปแบบและลักษณะการสื่อสารข้อมูลในระบบเครือข่ายให้ดีขึ้นไปอีก โดยให้สามารถส่งแฟ้มข้อมูลที่มีแบบต่าง ๆ และสามารถแสดงผลตามประเภทของแฟ้มข้อมูลนั้น ๆ ได้เลย ในส่วนนี้จึงได้เกิดเป็นเทคโนโลยี World Wide Web ซึ่งมีการทำงานแบ่งเป็น 2 ส่วนด้วยกันคือ ส่วนขอใช้บริการ (Client)และส่วนให้บริการ (Server) โดยลักษณะการให้บริการสามารถดูได้จากรูปที่ 3.1

เว็บเซร์ฟเวอร์ คือซอฟต์แวร์ที่ให้บริการระบบเวิด์ล ไวด์ เว็บ ให้บริการด้านการถ่ายโอนข้อมูลเอกสาร HTML โดยได้รับการร้องขอมาจาก Client หรือ Browser ข้อมูลที่ทำการถ่ายโอนมีได้ทั้งที่เป็นตัวอักษร ภาพ เสียง และข้อมูลมัลติมีเดียอื่น ๆ โดย Server และ Browser จะสามารถติดต่อกันได้โดยผ่านโปรโตคอล HTTP ดังนั้นในการติดตั้ง Web Server จึงเป็นการติดตั้งเพื่อให้บริการ HTTP นั่นเอง โดย Web Server ดั้งเดิมที่เป็นที่รู้จักและนิยมใช้กันอย่างแพร่หลายได้แก่ เซิร์ฟเวอร์ของ NCSA (National Center for Supercomputing Applications) ซึ่งนิยมเรียกกันว่า NCSA httpd ศูนย์ของ NCSA ตั้งอยู่ที่มหาวิทยาลัยอิลลินอยส์ ซึ่งที่นี่เป็นที่กำเนิดของบราวเซอร์ Mosaic ซึ่งถือได้ว่าเป็นต้นแบบของโปรแกรมบราวเซอร์

ในสมัยเริ่มแรกที่มีระบบเครือข่ายการติดต่อระหว่างเครื่องต่างลักษณะกันนั้นจะทำได้ก็ต้องอาศัยกติกาของการติดต่อระหว่างเครื่องที่เรียกว่า โปรโตคอล (Protocol) สำหรับมาตรฐานหนึ่งที่ใช้กันในอินเตอร์เน็ตก็จะอาศัยโปรโตคอลของการติดต่อกันระหว่างเครื่องที่เรียกว่า TCP / IP (Transmision Control Protocol / Internet Protocol)

3.2 การติดตั้ง Web server

รายละเอียดและวิธีการติดตั้ง Web server หรือเครื่องที่ให้บริการ www นั้นจะขึ้นอยู่กับเครื่องและซอฟท์แวร์หรือโปรแกรมที่ใช้ทำหน้าที่เป็น server การที่ระบบ www เป็นที่แพร่หลายอย่างที่เห็นกันอยู่ในตอนนี้ ก็เนื่องการลักษณะการใช้งานที่ง่าย และเปิดเผยรายละเอียดการติดตั้งและการใช้งาน รวมทั้งโปรแกรมที่ใช้สามารถหามาติดตั้งได้ง่าย สำหรับขั้นตอนและ

วิธีการติดตั้งที่จะอธิบายนี้จะอธิบายเฉพาะในเครื่องที่ใช้ระบบปฏิบัติการแบบ UNIX และบน Windows บางส่วนเท่านั้น

3.2.1 ประเภทของแฟ้มข้อมูลที่ใช้ส่ง

วิธีการกำหนดประเภทของแฟ้มข้อมูลที่ใช้ส่งมาจาก server นั้น ในส่วนของ Web server จะกำหนดจากแฟ้มข้อมูล mime.conf ซึ่งเป็นแฟ้มข้อมูลที่ใช้ระบุว่าแฟ้มข้อมูลที่ส่งออกไปเป็นแฟ้มข้อมูลเกี่ยวกับอะไร มีนามสกุลของแฟ้มข้อมูลหรือ extension อะไร

# pragma ident "@(#) mime.type 1.2 98/10/16 NCSA"	
application / activemessage	
application / applefile	
audio / x-aiff	aif aiff aifc
audio / x-wav	wav
image / gif	gif
image / x-portable-bitmap	pbm
message / partial	
message / rfc822	
multipart / alternative	
text / html	html
text / plain	txt
Video / mpeg	mpeg mpg mpe
Video / quicktime	qt mov
Video / x-msvideo	avi
Video / x-sgi-movie	movie

ตัวอย่างแฟ้มข้อมูล mime.conf โดยแสดงเฉพาะบางส่วนของแฟ้มข้อมูล

3.2.2 โปรแกรม Daemon ที่ทำหน้าที่เป็น Web server

โปรแกรมของ Web server ที่สามารถหามาติดตั้งได้และมีผู้นิยมใช้มากโปรแกรมหนึ่งเป็นของ NCSA ซึ่งโปรแกรมที่ทำหน้าที่เป็น Web server จะเรียกว่า httpd ซึ่งหมายถึงโปรแกรมที่ทำหน้าที่เป็น daemon ในระบบปฏิบัติการแบบ UNIX ซึ่งสามารถทำงานหลาย ๆ งานได้ในเวลาเดียวกันนั้น จะอาศัยวิธีการจัดโปรเซสหรือโปรแกรมที่ประมวลผล ในตอนเปิดเครื่องขึ้นมา ระบบจะทำการโหลดโปรแกรมที่จะทำหน้าที่คอยรับคำสั่งหรือทำหน้าที่จัดการกับโปรเซสที่เกี่ยวข้องเมื่อมีการเรียกใช้ ในกรณีที่ไม่มีการเรียกใช้โปรแกรมหรือโปรเซสพวกนี้ก็จะไม่ทำอะไร โปรแกรมที่ทำหน้าที่ในลักษณะนี้เรียกว่า daemon ซึ่งทำหน้าที่คล้าย ๆ กับโปรแกรม resident ในเครื่องคอมพิวเตอร์แบบ PC ที่ใช้ DOS เป็นระบบปฏิบัติการ โปรแกรมพวกนี้จะสามารถสังเกตได้จากชื่อ ซึ่งจะมีคำลงท้ายเป็นตัวอักษร " d " เช่น ftpd , inetd , httpd เป็นต้น httpd เป็นโปรแกรม daemon ที่ทำหน้าที่คอยรอรับการร้องขอ (reques)จาก Web browser จากนั้น Web server ก็จะทำการส่งแฟ้มข้อมูลที่ทำการร้องขอเข้ามาไปให้ โดยจะมีการเพิ่มเติมส่วนที่เป็น heading ที่จะบอกให้ Web browser ทราบว่าแฟ้มข้อมูลที่ส่งมาให้เป็นแฟ้มข้อมูลประเภทใด

3.2.3 การกำหนด Port

ก่อนการลงมือติดตั้ง Web server จะต้องมาทำความรู้จักกับส่วนสำคัญในระบบ UNIX ที่ใช้ในการกำหนดส่วนที่ให้บริการหรือ Service กันก่อน โดยส่วนนี้จะถูกกำหนดไว้ในแฟ้มข้อมูลที่ชื่อว่า /etc/services โดยบริการหลัก ๆ จะถูกกำหนดขึ้นมาอยู่ในช่วงต้น ๆ และสำหรับบริการอื่นที่นอกเหนือจากนี้หรือส่วนที่ผู้ใช้กำหนดขึ้นมาจะอยู่ในส่วนหลัง ๆ โดยเป็นหมายเลขของ Port ที่ให้บริการ สำหรับบริการของเว็บจะใช้ Port 80 สำหรับ http ส่วน Port หมายเลข 8080 ใช้สำหรับ proxy http สำหรับความหมายของ proxy นั้นก็คือ ส่วนที่ให้บริการในลักษณะเป็น cache ของ Web page ซึ่งสำหรับหน่วยงานที่มีจำนวนผู้ใช้มาก ๆ ควรจะมีเครื่องที่ให้บริการในส่วนนี้โดยเฉพาะ เพราะจะช่วยลดขนาดความคับคั่งของจราจรบนถนนทางด่วนข้อมูลได้มาก

3.2.4 ขั้นตอนการติดตั้งโปรแกรม Web server บน UNIX

โปรแกรม httpd ที่เป็นแฟ้มข้อมูลแบบ Binarys หรือ code ที่ทำการแปลหรือคอมไพล์เรียบร้อยแล้วนั้น สามารถที่จะติดตั้งได้โดยเลือก Binarys ที่จัดเป็น precompile บนเครื่องรุ่นต่าง ๆ ที่มีอยู่

ถ้าต้องการทราบถึงรายละเอียดของในส่วนนี้ก็จะสามารถเข้าไปดูได้จากเว็บไซต์ของ http://hoohoo.ncsa.uiuc.edu/docs/setup/precomplied.html ซึ่งจะมีรายการที่บอกรายละเอียดถึงรุ่นของเครื่องที่สามารถใช้งานได้ หรืออาจจะหา pre-build distribution kit มาทำการติดตั้งเองก็ได้ โดยจะอยู่ในรูปแฟ้มข้อมูล httpd.tar.Z โดยต้องทำการ compile ก่อนดังนี้

ขั้นตอนการแยกแฟ้มข้อมูลจากแฟ้มข้อมูลที่ถูกย่อขนาด

1. ทำการแยกแฟ้มข้อมูลที่ทำการย่อขนาดไว้ก่อนโดยใช้คำสั่งต่อไปนี้

```
uncompress  w3c – httpd – 3 . 0A . tar . Z
tar  xvf  w3c – httpd –3 . 0A . tar . Z
```

ขั้นตอนการแปลงจากซอร์สโปรแกรมให้เป็นภาษาเครื่อง

2. ทำการ compile และ link โดยใช้คำสั่ง Build Script โดยใน Script จะมีส่วนให้เลือกรุ่นเครื่องที่จะใช้ถ้าเคย Build version ก่อนหน้านี้จะมีไดเรกทอรี www อยู่ซึ่งจะต้องทำขั้นตอนนี้เพิ่มเติมด้วย

```
cd  www
make  clobber
make
```

แต่ถ้าไม่แน่ใจว่ามีการสร้างรหัส (code) ใหม่ขึ้นมาให้หรือเปล่า ก็ให้ไปที่ไดเรกทอรีใหม่โดยการใช้คำสั่งดังต่อไปนี้

```
cd  www
make
```

ขั้นตอนการเรียกใช้โปรแกรม

3. ทำการเรียกโปรแกรม httpd ที่อยู่ในไดเรกทอรี ../ www / Daemon / sun 4 (สำหรับเครื่อง sun) ในกรณีที่เป็นเครื่องประเภทอื่นก็ให้เรียกโปรแกรมตามไดเรกทอรีที่อยู่ของโปรแกรมนั้น ๆ สำหรับโปรแกรม utulity จะอยู่ที่ไดเรกทอรีต่าง ๆ ต่อไปนี้ htadm , htimage , cgiparse และ cgiutils

3.2.5 แฟ้มข้อมูลที่ใช้เป็นส่วนกำหนด (Configuration File)

โปรแกรม httpd จะอาศัยแฟ้มข้อมูลที่เป็นส่วนกำหนด โดยแฟ้มข้อมูลนี้โดยทั่วไปจะอยู่ภายใต้ไดเรกทอรี / etc / โดยแฟ้มข้อมูลจะมีชื่อว่า httpd.conf ในกรณีที่ต้องการระบุว่าอยู่ที่ไดเรกทอรีอื่นก็สามารถใช้คำสั่งดังต่อไปนี้ได้

```
Httpd  -r  /other/place/httpd . conf
```

ส่วนที่ต้องทำเพิ่มเติมคือการเข้ามาแก้ไขแฟ้มข้อมูลนี้ โดยการระบุชื่อเครื่องและโดเมน พร้อมทั้งชื่อและที่อยู่ของ E – mail ที่จะติดต่อได้ของผู้ดูแลการให้บริการ ซึ่งจะมีชื่อว่า Webmaster หรือ Postmaster แล้วแต่จะกำหนด ส่วนหนึ่งในแฟ้มข้อมูลนี้ที่ระบุไว้ว่า Port 80 คือการกำหนดบริการของ www ให้ดูรายละเอียดในเรื่องนี้จากหัวข้อของการกำหนด Port ส่วนที่เหลือคือการกำหนด log file เพื่อจะได้ทราบปริมาณการใช้งานว่ามากน้อยขนาดไหน มีการเรียกขอใช้บริการจากที่ใดเข้ามาบ้าง

```
# From sun_ httpd/conf/httpd .conf . generic on 10 / 16 / 98  11:.6  AM
# standalone  configuration  ( not inetd spawned )
#  This is the main server configuration file .
#
ServerType     standalone
```

Port	80
User	nobody
Group	nobody
ServerName	hostname . domainname
ServerAdmin	postmaster@domainname
ServerRoot	/opt /SUNWweb/ sun_httpd
ErrorLog	logs/error_log
TransferLog	logs/access_log
PidFile	logs/httpd.pid

ตัวอย่างแฟ้มข้อมูลที่ใช้เป็นส่วนกำหนด (httpd.conf)

3.2.6 ขั้นตอนการทดสอบ

เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้วควรทดสอบว่า Server ที่ติดตั้งสามารถส่งแฟ้มข้อมูลตามที่กำหนดไว้ในแฟ้มข้อมูล mime.conf ได้หรือไม่ และมีการเก็บ access log หรือรายการที่มีผู้เข้ามาเรียกใช้จากที่กำหนดไว้ใน httpd.conf หรือไม่ โดยการทดลองเรียกแฟ้มข้อมูลที่ใส่ไว้ในไดเรกทอรีของเครื่องที่ติดตั้งเป็นเครื่อง Server ดู โดยใช้ Browser เรียกขอแฟ้มข้อมูลที่ต้องการมายังที่อยู่ของเครื่องที่เราติดตั้งไว้ ซึ่งถ้าสามารถเรียกดูแฟ้มข้อมูลที่ต้องการได้ก็เป็นอันว่าเสร็จขั้นตอนการติดตั้ง แต่ถ้ามีข้อความแจ้งข้อผิดพลาด ว่าหาเครื่องที่ให้บริการไม่เจอ ก็ต้องลองตรวจดูว่าระบบเครือข่ายมีปัญหาหรือไม่ หรือถ้าเครื่องแจ้งว่าไม่พบแฟ้มข้อมูลที่ใส่ไว้ใน Serwer ก็ให้ลองดูว่าใส่ชื่อผิดพลาดหรือไม่ หรือลองดูว่าใน Server ที่ติดตั้งมีแฟ้มข้อมูลนั้น ๆ อยู่หรือไม่ ข้อควรสังเกตนามสกุลของแฟ้มข้อมูลเว็บบนเครื่องที่ใช้ระบบปฏิบัติการ NUIX จะเป็น . html ในขณะที่บน Windows จะเป็น htm

3.2.7 ขั้นตอนการติดตั้ง Web Server บน Windows

สำหรับผู้ที่ต้องการลองใช้งาน Web Server บนเครื่องที่ใช้ระบบปฏิบัติการ Windows ก็สามารถทำได้โดยเลือก download โปรแกรมที่ทำหน้าที่เป็น Web Server มาลองใช้ดูก่อนได้ ซึ่งจะขอแนะนำ Fast Track ของบริษัท Nescape ซึ่งขั้นตอนการติดตั้งก็เพียงแต่เรียก

SETUP.EXE จากโปรแกรม จากนั้นโปรแกรมก็จะทำการติดตั้งแก้ไขรายการให้เอง จากนั้นก็เป็นการลองทดสอบใช้งานดูตามขั้นตอนที่ได้กล่าวมาแล้ว ข้อสำคัญคือเครื่องที่ทำการติดตั้งจะต้องติดตั้งส่วนของระบบเครือข่ายและโปรโตคอล TCP ไว้ให้เรียบร้อยก่อน และข้อสำคัญคือจะต้องกำหนด Port ที่จะใช้ในการติดต่อเป็น Port 80

3.3 ชนิดของ Web Server ที่มีให้บริการ

นอกจากจะมี Web Server ของ NCSA แล้วก็ยังมี Web Server บนเครื่องในแบบอื่น ๆ อีก แต่จำนวนผู้ใช้จะแตกต่างกันไป โดยจะขึ้นอยู่กับความสามารถของ Web Server และราคา เนื่องจากในตอนแรกนั้นชุดโปรแกรมต้นแบบสามารถนำไปพัฒนาเติมได้ ก็มีหลายบริษัทนำไปแก้ไขเพิ่มเติมโดยสามารถนำไปใช้งานในทางธุรกิจได้ ตัวอย่างของ Web Server อื่น ๆ เช่น Apache httpd , FastTrack Netscape Communication Server และ Web Site ซึ่งเป็นของบริษัท O'Reilly ซึ่งเป็นผู้พิมพ์หนังสือที่เกี่ยวข้องกับ Internet

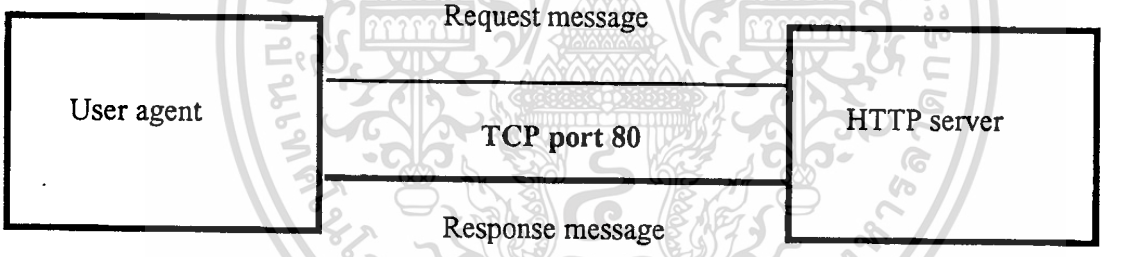
นอกจากการใช้งาน Web Server บนเครื่องที่ใช้ระบบปฏิบัติการแบบ UNIX แล้วก็ยังสามารถเลือกใช้ Web Server บนเครื่องใน Platform อื่นก็ได้ เช่นบนเครื่องที่ใช้ระบบปฏิบัติการ แบบ Windows NT หรือบน Windows 95 ก็มีผู้ที่พัฒนาขึ้นมาสำหรับใช้งาน โดยถ้าพิจารณาตลาดก็จะพบว่าคนส่วนใหญ่จะใช้เครื่องในแบบหลังมากกว่า และความต้องการใช้งานในลักษณะสำหรับให้บริการเป็น Web Server ก็เพิ่มมากขึ้นเพราะขั้นตอนการพัฒนา Web Page หรือ Home Page นั้นทำได้ง่ายและสะดวก มีภาษาและเครื่องมือสนับสนุนอยู่เป็นจำนวนมาก ทำให้ลักษณะของข้อมูลมีมากมายหลากหลายประเภทดังที่ปรากฏอยู่ในตอนนี้

จากหัวข้อที่ผ่านมาจะเห็นว่าการใช้งานและการติดตั้งระบบ www นั้นจะไม่ค่อยยุ่งยากมากนัก แต่สิ่งที่ยากกว่ากลับเป็นการที่จะดูแล Web Site ที่ตั้งขึ้นมาอย่างไร ควรที่จะนำเอาข้อมูลประเภทใดมาใส่ไว้ ทำอย่างไรให้มีคนสนใจเข้ามาชม Web Site ดังกล่าวนั้นให้มากที่สุด

บทที่ 4
หลักการทำงานของระบบสื่อสาร HTTP และมาตรฐานการรับส่ง FTP

4.1 รูปแบบการทำงานของ HTTP

HTTP ย่อมาจากคำว่า HyperText Transfer Protocol ซึ่งเป็นโปรโตคอลสื่อสารที่ทำงานอยู่บนระบบโปรโตคอล TCP/IP โดย HTTP ใช้อยู่ในระบบเครือข่ายเวิล์ล ไวด์ เว็บ ทำหน้าที่ในการแจกจ่าย รวมไปถึงการรับข้อมูล จากระบบสื่อกลางชั้นสูง (Hypermedia System) ที่ประกอบด้วยเครื่องให้บริการ (Server) ที่มีอยู่มากมายทั่วโลก ซึ่งลักษณะการทำงานโดยภาพรวมแล้วจะประกอบด้วยขั้นตอนดังที่แสดงในรูป 4.1



รูปที่ 4.1 แสดงขั้นตอนการขอและการส่งระหว่าง Browser และ Server

การเชื่อมต่อ (Connection) เป็นการสร้างการเชื่อมต่อระหว่าง Client ไปยัง Server โดยผ่านโปรโตคอล TCP/IP ที่ Port 80 ซึ่งจะถือว่าเป็น Port ที่ทราบกัน (default port) หรือถ้าจะผ่าน Port อื่นก็จะต้องมีการระบุใน URL

การขอ (Request) เป็นรายการที่ส่งมาจาก Client ของข้อความที่ขอ (Request) ไปยังเครื่อง Server

การตอบรับ (Response) เป็นรายการที่ส่งโดย Server ของการตอบรับ (Respond) กลับมายัง Client

การปิด (Close) เป็นการปิดการเชื่อมต่อระหว่างสองฝ่าย จะเกิดขึ้นเมื่อสิ้นสุดการส่งรายการ หรือการที่ฝ่ายใดฝ่ายหนึ่งปิดการติดต่อไป

รายละเอียดส่วนประกอบของโปรโตคอล

รายละเอียดและส่วนประกอบของโปรโตคอลจะประกอบด้วยหัวข้อดังนี้

- การขอ (The Request)
- วิธีการ (Methods)
- รายการส่วนหัวของข้อความที่แจ้งขอ (A list of headers in the request message)
- การตอบ (The Response)
- รหัสแจ้งสถานะ (Status codes)
- ส่วนหัวของรายการชั่วคราวของข้อมูลที่ถูกส่ง (Meta information headers on any object transmitted)
- เนื้อความที่ถูกส่ง (The content of any object content transmitted)
- วิธีการเจรจา (Format negotiation algorithm)
- สิทธิการลงทะเบียน (HTTP Registration Authority)
- การรักษาความปลอดภัย (Sequrity Considerations)
- ส่วนอ้างอิง (Reference)

จะสังเกตเห็นได้ว่าลักษณะของการทำงานเป็นแบบ one to one ของรายการใด ๆ ที่ขอเข้ามาก็ค่อยส่งรายการนั้นไปให้ ลักษณะของโปรโตคอลเองจะไม่มีลักษณะของการกระจาย (broadcast) คือให้กับผู้รับหลาย ๆ คนพร้อม ๆ กัน และโปรโตคอลเองก็ไม่มีสถานะ (stateless) การสิ้นสุดของการรับส่งสามารถเกิดขึ้นได้เมื่อจบการรับส่ง หรืออาจเกิดขึ้นได้จากการที่ฝ่ายหนึ่งฝ่ายใดยกเลิกรายการขอ หรือรายการส่ง

โปรโตคอล HTTP ทำงานอยู่บนหลักการการเรียกขอและการตอบรับ (Request /Response paradigm) โดยการทำงานจะเริ่มจากส่วนของ Client สร้างขั้นตอนการเชื่อมต่อกับส่วนที่ให้บริการ Server เพื่อรับรายการ ดังรูปที่ 4.1 จากนั้นจะมีการส่งรายการที่ขอมายัง Server โดยรายการที่ขอจะอยู่รูป วิธีการขอ (request method) ที่ประกอบด้วย URI และ version ของโปรโตคอล ตามด้วยข้อความที่มีลักษณะแบบ MIME ที่บอกว่าต้องการอะไร และตามด้วยข้อความของ Client จากนั้น Server ก็จะตอบกลับโดยมี status line ที่มีทั้ง version

โปรโตคอลและโค้ดที่แจ้งว่าการส่งได้หรือว่ามีปัญหาหรือไม่ และตามด้วย MIME type หรือส่วนที่แจ้งว่าส่งอะไรมาให้ของ Server เอง ซึ่งรูปแบบของการบอกชนิดของข้อมูลเริ่มนำมาใช้กับโปรโตคอล HTTP นี้

4.1.1 การกำหนด Port

เนื่องจากโปรโตคอล HTTP ทำงานอยู่ในระดับบนที่ Application Layer ของโปรโตคอลที่ใช้สื่อสาร TCP/IP โดยพอร์ตมาตรฐานที่ใช้จะเป็น Port 80 แต่ถ้าไม่ต้องการใช้งานที่พอร์ตนี้ก็สามารถทำได้โดยที่ Server หรือเครื่องให้บริการจะต้องมีการระบุเอาไว้ว่าจะใช้ Port ไหน เช่นถ้าจะใช้ Port 81 สำหรับHTTP นั้น Client ก็สามารถ Request หรือขอรายการไปยัง Server ได้โดยระบุที่ URL ดังตัวอย่าง http://www.somewhere.com:81 แต่โดยปกติแล้วจะใช้ Port 80 เป็นหลัก ในส่วนของการทำงานจริงระหว่าง Client และ Server จะต้องสามารถรับกับปัญหาในลักษณะนี้ได้ เช่นการที่ฝ่ายใดฝ่ายหนึ่งจะขาดจากการติดต่อ เนื่องจากหมดเวลา หรือยกเลิกโดยผู้ใช้ การปิดการติดต่อจะทำโดยการที่ฝ่ายใดก็ตามยกเลิกการติดต่อโดยจะไม่ขึ้นอยู่กับสถานะในขณะนั้น

4.1.2 ไวยากรณ์และสัญลักษณ์

สำหรับข้อกำหนดหรือโปรโตคอลจะต้องมีไวยากรณ์หรือข้อกำหนดของภาษาสำหรับอธิบายการใช้งานสัญลักษณ์ที่ใช้อธิบายในข้อกำหนดของโปรโตคอล HTTP หรือโปรโตคอลอื่น ๆ โดยทั่วไปจะใช้ Backus Naur Form (BNF) ซึ่งจะประกอบด้วย

name = definition

โดยที่ชื่อของกฎที่ใช้จะไม่อยู่ภายใต้เครื่อง “ <” และ “>” และแยกจากนิยามหรือ definition โดยเครื่องหมาย “ =” กฎหรือ Rules ที่ใช้ในโปรโตคอลจะระบุเป็นตัวอักษรใหญ่ เช่น SP , TAB , CRLF , DIGIT , ALPHA

“ literal”

ข้อความที่เป็นเนื้อความจะล้อมรอบด้วยเครื่องหมาย “ ” (Quotation marks) โดยที่ข้อความจะไม่แตกต่างกันสำหรับตัวอักษรเล็กหรือใหญ่

rule 1 | rule2

ตัวที่คั่นด้วยเครื่องหมาย bar (" | ") จะหมายถึงตัวเลือก เช่น " yes | no " จะหมายถึงการรับว่า ใช่ หรือ ไม่

(rule1 rule2)

ตัวที่คั่นด้วยวงเล็บจะถูกมองเป็นรายการเดียว เช่น " (elem (foo | bar) elem)" จะมองเป็นลำดับของ token " elem foo elem" และ " elem bar elem"

***rule**

เครื่องหมายดอกจันที่อยู่หน้าหมายถึงการทำซ้ำ รูปเต็ม ๆ จะเขียนเป็น " < n >*< m >element " ซึ่งหมายถึงว่าจะต้องมีอย่างน้อย < n > และ < m > ปรากฏอยู่ ค่า Default values จะเป็น 0 และ infinity ดังนั้น " *element " ก็คือจะต้องมีอย่างน้อยหนึ่ง

[rule]

เครื่องหมายวงเล็บปีกกานี้จะใช้บอกตัวเลือก เช่น " [foo bar]" ซึ่งก็จะมีค่าเทียบเท่ากับ * l(foo bar) "

N rule

ใช้ระบุค่าซ้ำ "<n> (element)" จะหมายถึง "<n> * <n> (element)"; เช่น 2DIGIT คือเป็นตัวเลข 2 ตัวและถ้าเป็น 3 ALPHA จะเป็นตัวอักษรสามตัว

#rule

เครื่องหมายนี้ " #" ใช้เหมือนกับ " * " ใช้ระบุชุดของสมาชิก รูปเต็ม ๆ จะเขียนเป็น "< n >#< m >element" ซึ่งจะบอกว่ามีอย่างน้อย < n > และส่วนมาก < m > แต่ละตัวคั่นด้วยเครื่องหมาย "," และบรรทัดว่าง (LWS) เครื่องหมายนี้จะใช้กับกฎง่าย ๆ เช่น " (*LWS element *(*LWS " ," *LWS element))" สามารถแสดงได้เป็น " l#element" เมื่อไรก็ตามที่ใช้เครื่องหมายนี้จะยอมให้มีตัวว่างได้ (null element) แต่จะไม่นับ

; comment

ใช้ระบุรายละเอียดเพิ่มเติมโดยวางไว้ทางขวาของข้อความ มีประโยชน์อย่างมากในการอธิบายรายละเอียดกำกับในข้อกำหนดหรือ specifications

4.1.3 รายเอียดของ Header Field

ส่วนที่เป็นส่วนกำหนดรายการส่วนประกอบหรือ Header Field ของ HTTP จะประกอบด้วยส่วนที่เก็บรายละเอียดเกี่ยวกับการขอหรือ Reques - Header ส่วนของรายการตอบ Response – Header รายการทั่วไป General – Header รายการของส่วนประกอบ Object – Header และรายการของส่วนขยาย extension field รายการที่เป็นส่วนหัวแต่ละรายการนี้จะคั่นด้วยเครื่องหมาย : และอาจจะยาวได้หลายบรรทัด

```
HTTP-header     = field-name " : " [ field-value ] CRLF
field-name      = 1*<any CHAR , excluding CTLs , SP , and " : " >
field-value     = * ( field-content | comment | LWS )
field-content   = < the OCTETs making up the field-value
                  and consisting of either *text or combinations
                  of token , tspecials , and quoted-string >
```

แสดงส่วนของ header field ของโปรโตคอล HTTP

4.2 ระบบปฏิบัติการมาตรฐาน FTP

เอฟทีพี (FTP – File Transfer Protocol) หมายถึงระบบปฏิบัติการบนอินเตอร์เน็ตที่ใช้ในการ download files ข้อมูลจากเครื่องคอมพิวเตอร์อื่น ๆ บนอินเตอร์เน็ตมาเก็บไว้ที่คอมพิวเตอร์ของผู้ใช้ หรือการ upload files ข้อมูลจากเครื่องคอมพิวเตอร์ของผู้ใช้ส่งไปที่ศูนย์บริการเอฟทีพี (FTP Server) เพราะฉนั้น FTP จึงเป็นเครื่องมือปฏิบัติการบนอินเตอร์เน็ต อีกตัวหนึ่งที่มีความสำคัญและมีผู้ใช้มาก ในการ download file จากคอมพิวเตอร์เครื่องอื่น

มาเก็บที่เครื่องของผู้ใช้นั้นจะช้ากว่าการ upload file จากเครื่องของผู้ใช้ไปเก็บที่ FTP Server ซึ่งการเข้าใช้ศูนย์บริการ FTP จะมีด้วยกัน 2 วิธีคือ

1. **ใช้ในฐานะของสมาชิกประจำศูนย์บริการ FTP นั้น** ในกรณีนี้ศูนย์บริการจะกำหนด account ให้ โดยจะต้องบอกรหัสประจำตัวผู้ใช้ (user ID) และรหัสลับ (password) ให้ถูกต้อง จึงจะสามารถทำการ download หรือ upload file ข้อมูลได้

2. **ใช้ในฐานะเป็นบุคคลภายนอกที่ไม่มี account อยู่กับ FTP Server** ซึ่งสามารถที่จะ download files ได้จาก pub (public) directory โดยเข้าใช้ในนาม anonymous user. ผู้ใช้ต้องพิมพ์คำว่า **anonymous** เพื่อ login เข้าใช้บริการ เมื่อมีคำว่า password ส่งมา ให้เติม e-mail address ลงไปแทน

4.2.1 การใช้ FTP ระบบ UNIX

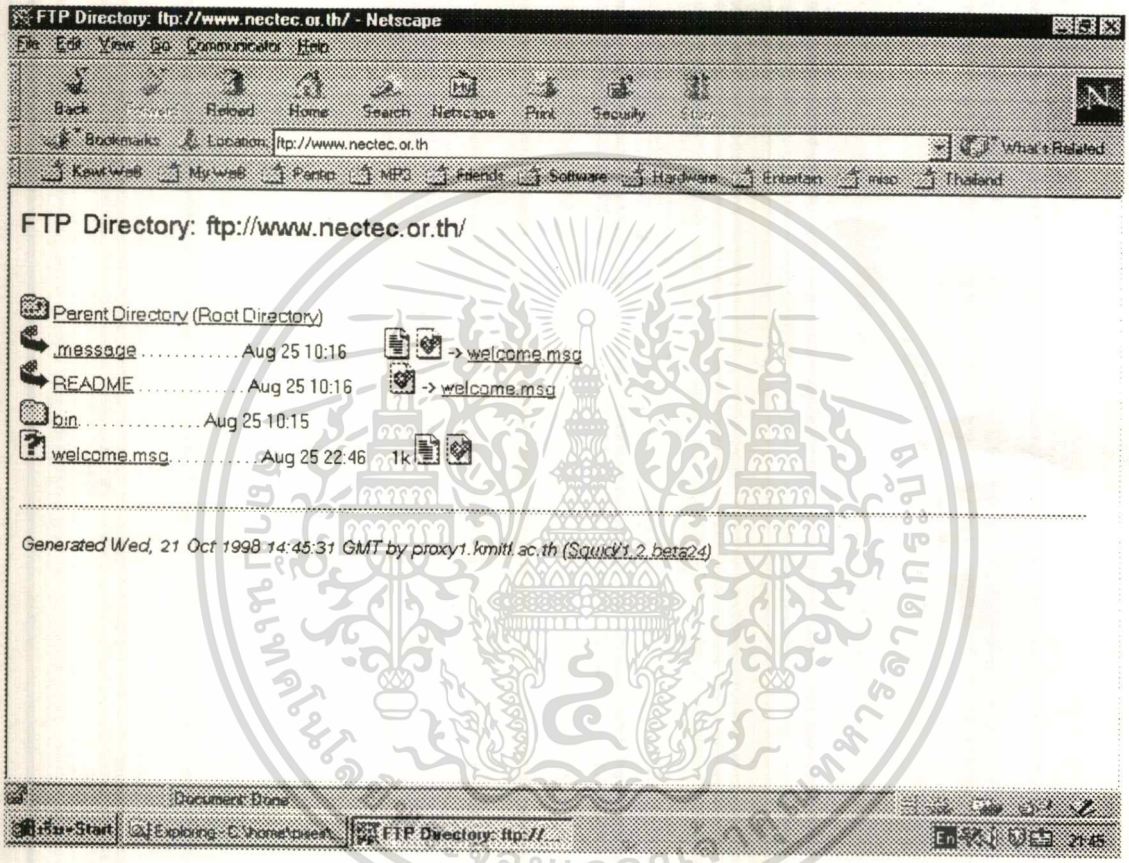
FTP ในระบบ UNIX (text based system) เป็น FTP ขั้นพื้นฐานสำหรับผู้ใช้ระบบอินเตอร์เน็ต ที่ทำการเชื่อมโยงโดยวิธีติดต่อผ่านศูนย์ให้บริการ อินเตอร์เน็ต การใช้ FTP ระบบ UNIX จะยากกว่าการใช้ FTP บนระบบ Windows เนื่องจากจะต้องทราบคำสั่งระบบ UNIX และในการ download file โดย FTP ระบบ UNIX ก็จะต้องทำการ download file มาเก็บไว้ยังที่ Home directory บนเครื่องของศูนย์บริการก่อน จากนั้นจึง download file จาก Home directory มาเก็บที่คอมพิวเตอร์ของผู้ใช้อีกทีหนึ่ง นั้นคือ จะต้องดำเนินการถึงสองขั้นตอนในการ download file ขณะที่การใช้ FTP ระบบ Windows ซึ่งสามารถ download file จาก FTP Server มาที่เครื่องคอมพิวเตอร์ผู้ใช้ได้โดยตรงทีเดียว แต่อย่างไรก็ดีการ download file โดย FTP ระบบ UNIX ก็จะเร็วกว่า FTP ในระบบ Windows

4.2.2 โปรแกรม FTP

เนื่องจากโปรแกรมที่สามารถสื่อสารแบบ FTP ได้มีอยู่จำนวนมาก แต่ในที่นี้จะขอยกตัวอย่างเฉพาะโปรแกรมเนสเคปเท่านั้น ซึ่งเป็นที่นิยมใช้กันมากในขณะนี้ โดยเนสเคปนี้เป็นโปรแกรมที่สามารถสื่อสารได้หลายภาษา ไม่ว่าจะเป็น http , mailto และ FTP ถ้าผู้อ่านต้องการสื่อสารแบบ FTP บนโปรแกรมเนสเคป ก็ให้ใช้ URL ที่มีรูปแบบดังนี้

ftp://xxx.xxx.xxx.xxx/yyy/... โดยที่ xxx.xxx.xxx.xxx เป็นชื่อของเครื่องที่เราต้องการใช้บริการ และ /yyy ... จะแทนไดเรกทอรีย่อยที่ผู้อ่านต้องการเข้าไป ข้อจำกัดของโปรแกรม

เนสเคปคือ เราสามารถใช้บริการ Anonymous Download เท่านั้น โดยที่โปรแกรมเนสเคปจะจัดการใส่ username และ password ให้โดยอัตโนมัติ สมมติว่าผู้ใช้ต้องการที่จะใช้บริการจากเครื่อง ftp.nectec.or.th เมื่อใส่ URL แล้วก็จะได้หน้าจอปรากฏดังตัวอย่างในรูปที่ 4.2 การ download ก็สามารถทำได้โดยการคลิกเมาส์ที่ LINK ที่เป็นชื่อไฟล์ตามที่ต้องการ



รูปที่ 4.2 แสดงตัวอย่างหน้าจอของ FTP

บทที่ 5
ระบบการรักษาความปลอดภัยของ Web Server

5.1 ความสำคัญของระบบการรักษาความปลอดภัย

การรักษาความปลอดภัยในการใช้งานของระบบ www นั้นเป็นเรื่องที่จำเป็นจะต้องคำนึงถึง เพราะมันเป็นส่วนหนึ่งที่ทำให้เครือข่ายอินเตอร์เน็ตและ www เป็นที่นิยมกัน โดยที่เครือข่ายที่มีความปลอดภัยในการใช้งานที่ดีพอก็จะมีผู้ที่นิยมใช้กันมากขึ้น โดยเฉพาะอย่างยิ่งในการใช้งานด้านการพาณิชย์เชิงอิเล็กทรอนิกส์นั้นก็จะต้องมีการรักษาความปลอดภัยของข้อมูลที่มีประสิทธิภาพสูง มิเช่นนั้นแล้วผลเสียก็จะต้องตกอยู่กับผู้ใช้บริการเอง เช่น การถูกลักลอบนำรหัสของบัตรเครดิตไปใช้ เป็นต้น ด้วยเหตุนี้จึงได้มีผู้ที่คิดค้นและมุ่งที่จะพัฒนาโปรโตคอลเพื่อใช้ในการรักษาความปลอดภัยของข้อมูลในการใช้งาน www ด้วยกันหลายแบบ เช่น S – HTTP (Secure Hypertext Transfer Protocol) และ SSL (Secure Sockets Layer) เป็นต้น ในที่นี้จะแนะนำถึงแบบ SSL โปรโตคอลแบบ SSL นี้เป็นของบริษัท Netscape ซึ่งเป็นโปรโตคอลตัวหนึ่งที่ใช้ในการรักษาความปลอดภัยของข้อมูลได้ โดยมีคุณสมบัติเด่นคือ ไม่ขึ้นอยู่กับ Application Protocol ใด ๆ จึงทำให้สามารถทำงานร่วมกับโปรโตคอลอื่น ๆ ได้อย่างหลากหลาย รวมทั้งยังสามารถสนับสนุนการเข้ารหัสได้หลากหลายวิธีอีกด้วย

5.2 ระบบการรักษาความปลอดภัยของโปรโตคอล SSL

การทำงานของโปรโตคอล SSL นั้นจะทำการรักษาความปลอดภัยของข้อมูลโดยจัดให้มีการเข้ารหัสข้อมูล ซึ่งสามารถใช้อัลกอริทึมได้หลายวิธี การพิสูจน์ความแท้จริงของ Server (Server Authenticate) โดยการใช้ Certificate และการตรวจสอบความถูกต้องและความสมบูรณ์ของข้อมูล โดยใช้อัลกอริทึม MAC ซึ่งตัว SSL จะทำงานเป็น state และ session โดยตัวของ SSL เองจะประกอบไปด้วย 2 ระดับชั้นคือ SSL Record Layer ซึ่งจะเป็นตัวที่ทำหน้าที่จัดการเกี่ยวกับข้อมูลที่ได้รับมาจากชั้น application ไม่ว่าจะเป็นการทำการแยกข้อมูลออกเป็นบล็อก (fragmentation) การบีบอัดข้อมูล (data compression) หรือการเข้ารหัสข้อมูล (data encryption) การตรวจสอบความถูกต้องและความสมบูรณ์ของข้อมูลโดยใช้

MAC ซึ่งวิธีการที่จะใช้กับ SSL Record Layer ทั้งหมดจะถูกกำหนดไว้ใน CipherSuite ซึ่งได้จากอีกระดับชั้นหนึ่งของ SSL คือในชั้นของ SSL Handshake Layer โดยชั้นนี้จะมีหน้าที่ในการทำความตกลง (negotiation) ระหว่าง client และ server เพื่อหาวิธีการที่จะใช้ทั้งในการเข้ารหัส การคำนวณหา MAC และการพิสูจน์ความแท้จริงของ server ด้วย

ในการเข้ารหัสข้อมูลโดยตัวของ SSL นั้นจะทำงานในระดับชั้นที่ต่ำกว่าชั้น Application Layer เช่น HTTP , SMTP , Telnet และ S- HTTP เป็นต้น แต่ก็ยังอยู่ในระดับชั้นที่เหนือกว่าชั้น Transport Layer ซึ่งได้แก่ TCP / IP นั่นเอง โดยดูได้จากรูปที่ 5.1

รูปที่ 5.1 แสดงระดับชั้นในการทำงานของ SSL

จากหลักการดังกล่าวทำให้เกิดเป็นข้อดีข้อเสียของ SSL คือ ทำให้ SSL สามารถใช้งานได้กับ Application ทุก ๆ ตัวที่ทำงานอยู่บนโปรโตคอล TCP/IP ซึ่งเป็นสื่อกลางในการสื่อสารของเครือข่ายอินเตอร์เน็ตโดยไม่ยึดติดอยู่กับตัวใดตัวหนึ่ง ปัจจุบันโปรโตคอล SSL ไม่ได้ถูกจำกัดให้เป็นโปรโตคอลเฉพาะสำหรับของ Netscape เท่านั้น แต่เป็นโปรโตคอลที่เปิดกว้างโดย SSL เริ่มถูกส่งเข้ามาให้พิจารณาใช้โดยกลุ่ม W3C (World Wide Web Consortium) ที่ต้องการให้มีมาตรฐานในการรักษาความปลอดภัยของข้อมูลสำหรับบริการบน www ทั้งในส่วนที่เป็น Client และ Server ด้วย เนื่องจากลักษณะการทำงานของ www จะทำงานในลักษณะของ เครื่อง Client และ เครื่อง Server เพราะฉะนั้น ในการใช้โปรโตคอลSSL จะใช้งานได้ก็ต่อเมื่อเครื่องของ Client และ Server จะต้องสนับสนุนในการใช้งาน SSL นั้นด้วย ถ้ามีส่วนหนึ่งส่วนใดที่ไม่สนับสนุนแล้วโปรโตคอล SSL นี้ก็จะใช้งานไม่ได้

ในการใช้งาน SSL บนโปรแกรมเว็บบราวเซอร์นั้นจะสังเกตได้ดังนี้ เมื่อมีการใช้งานกับเครื่อง Server ก็ให้ระบุที่ URL หรือที่อยู่ของเอกสารเป็น “ https “ สำหรับการเชื่อมต่อกับ

Server ในแบบ HTTP สำหรับส่วนที่รักษาความปลอดภัยของ www หรือ https นั้นจะทำงานผ่าน Port 443 ซึ่งเป็น Port มาตรฐานที่ถูกกำหนดโดยหน่วยงานที่กำหนดหมายเลขของอินเตอร์เน็ต หรือ Internet Assigned Numbers Authority (IANA) โดยทั่วไปการติดต่อกันระหว่างเครื่องโดยมีการขอเรียกใช้บริการประเภทต่าง ๆ นั้นจะต้องอาศัยกติกาและ Port ที่กำหนดไว้เพื่อการขอใช้บริการ ตัวอย่างเช่น Port การติดต่อของโปรโตคอลแบบ HTTP จะต้องต่อที่ Port 80 สำหรับการใช้งาน Proxy ก็จะต้องผ่าน Port 8080 เป็นต้น

5.3 การทำงานของ โปรโตคอล SSL

เมื่อมีความต้องการที่จะรับส่งข้อมูลที่มีความสำคัญมาก หรือต้องการเก็บเป็นความลับก็จะต้องเป็นหน้าที่ของโปรโตคอล SSL เช่น เมื่อต้องการที่จะสั่งซื้อสินค้าทางอินเตอร์เน็ต แล้วเครื่องทางด้าน Client และ Server ก็สนับสนุน SSL อยู่แล้ว เมื่อมีการกรอกข้อความแล้วในการส่งข้อมูลนั้นเครื่องทางด้าน Client และ Server จะทำการ handshake ซึ่งเป็นหน้าที่ของ SSL Handshake Protocol คือจะเริ่มเมื่อเครื่องทางด้าน Client ส่งสัญญาณ Client hello ก่อนเพื่อจะทำการตกลงเกี่ยวกับการกำหนดพารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการรักษาความปลอดภัย ไม่ว่าจะเป็นการเข้ารหัส การคำนวณ MAC และการพิสูจน์ความแท้จริงของแต่ละด้าน เมื่อดำเนินการตกลงกันเกี่ยวกับพารามิเตอร์ต่าง ๆ ได้แล้ว ก็จะมีการส่งข้อมูลในชั้นของแอพพลิเคชั่น (Application Data) ซึ่งจะต้องเป็นหน้าที่ของ SSL Record Protocol ที่จะทำหน้าที่แยกข้อมูลออกเป็นส่วน ๆ เข้ารหัสข้อมูล หรือคำนวณค่า MAC เป็นต้น ตามวิธีที่ได้ตกลงกันไว้ในส่วนของ SSL Handshake Protocol เพื่อรักษาความปลอดภัยของข้อมูลแล้วจึงส่งต่อไปยัง Server ตามต้องการ

5.4 รายละเอียดต่าง ๆ เกี่ยวกับ SSL

เป้าหมายพื้นฐานของโปรโตคอล SSL คือ การจัดเสนอความเป็นส่วนตัว (privacy) และความน่าเชื่อถือ (reliability) ระหว่างสองแอพพลิเคชั่นที่ใช้ในการติดต่อสื่อสารกัน โปรโตคอล SSL จะประกอบด้วยสองชั้นคือ

1. SSL Record Layer จะอยู่ในระดับต่ำสุด แต่อยู่บนโปรโตคอลที่ใช้ในการสื่อสาร (Transport Protocol) เช่น TCP/IP เป็นต้น SSL Record Protocol นี้จะใช้สำหรับล้อมและอธิบายรายละเอียดคร่าว ๆ เกี่ยวกับโปรโตคอลในระดับที่สูงขึ้นไป

2. SSL Handshake Protocil จะอนุญาตให้ Client และ Server สามารถที่จะพิสูจน์ซึ่งกันและกันได้ และจะทำความตกลงเกี่ยวกับอัลกอริทึมในการเข้ารหัส รวมทั้งค้นหากุญแจในการเข้ารหัส (cryptographic keys) ก่อนที่จะมีการรับหรือส่งไบท์แรกของข้อมูล

ประโยชน์อย่างหนึ่งของ SSL คือ มันไม่ขึ้นอยู่กับโปรโตคอลของแอพพลิเคชั่นใด โปรโตคอลในระดับสูงกว่าสามารถทำงานอยู่บนโปรโตคอล SSL ได้อย่างแน่นอน

โปรโตคอล SSL จะจัดเสนอ ความปลอดภัยในการเชื่อมต่อ (connection security) ซึ่งมีคุณสมบัติพื้นฐาน 3 ประการ คือ

1. การเชื่อมต่อ (connection) จะต้องเป็นส่วนตัว การเข้ารหัสจะถูกใช้หลังจากมีการเริ่มทำ handshake เพื่อจะกำหนด secret key การเข้ารหัสแบบสมมาตร (symmetric cryptography) เช่น DES , RC4 ก็จะถูกนำมาใช้ในการเข้ารหัสข้อมูล

2. เอกลักษณ์ของแต่ละ peer สามารถจะถูกพิสูจน์ได้ โดยใช้การเข้ารหัสแบบไม่สมมาตร (Asymmetric cryptography) หรือแบบกุญแจสาธารณะ (public key)

3. การเชื่อมต่อจะสามารถเชื่อถือได้ในการรับส่งข้อมูลนั้น จะมีการเช็คความครบถ้วนและความถูกต้องของข้อมูลโดยใช้กุญแจ MAC Secure hash functions เช่น SHA , MD5 จะถูกนำไปใช้ในการคำนวณหา MAC

นอกจากนี้แล้วเนื่องจากในการกระทำการเข้ารหัสนั้นจะต้องการใช้ CPU สูงมาก โดยเฉพาะอย่างยิ่งในการเข้ารหัสแบบ public key โปรโตคอล SSL จะมี option ในการทำ session caching scheme เพื่อช่วยลดจำนวนในการเชื่อมต่อที่จะต้องเริ่มตั้งแต่ต้น ซึ่งมันจะช่วยลดภาระของ network ลงได้ เนื่องจากSSL เป็นโปรโตคอลที่ทำงานเป็นชั้น (Layer) ในแต่ละชั้นข้อความอาจจะรวมส่วนที่แสดงถึง ความยาว (length) คำอธิบาย (description) และเนื้อหา (content) โปรโตคอล SSL จะนำข้อมูลมาเพื่อเตรียมพร้อมที่จะทำการส่ง แยกข้อมูลให้อยู่ในรูปของบล็อก (block) ข้อมูลที่สามารถจัดการได้ สามารถทำการบีบอัดข้อมูลได้ พร้อมทั้งใช้ MAC การเข้ารหัส แล้วส่งข้อมูลไป ข้อมูลที่จะถูกรับเข้ามาจะถูกทำการถอดรหัส ตรวจสอบความถูกต้อง คลายจากการบีบอัด และนำข้อมูลมารวมกัน เพื่อส่งข้อมูลไปสู่ Client ในระดับสูงต่อไป

5.4.1 Session and connection states

ในแต่ละ session ของ SSL จะมีการทำงานแบบเป็น state ของทั้ง client และ server เพื่อให้แต่ละส่วนทำงานได้สอดคล้องกัน แม้ว่ามันจะไม่ได้ทำงานพร้อมกันและขนานกันอย่างแท้จริงก็ตาม ด้วยเหตุนี้จึงมีการแทนถึง state สองครั้ง ครั้งหนึ่งคือ current operating state และอีกครั้งหนึ่งเป็น pending state นอกจากนี้แล้วก็ยังแยกได้เป็น read state และ write state ออกจากกันอีกด้วย เมื่อ client หรือ server ได้รับข้อความ change cipher spec มันก็จะคัดลอก pending read state ไปสู่ current read state เมื่อ client หรือ server ส่งข้อความ change cipher spec มันก็จะคัดลอก pending write state ไปสู่ current write state เมื่อการตกลงกันในการทำ handshake เสร็จสมบูรณ์ client หรือ server ก็จะทำการแลกเปลี่ยนข้อความ change cipher spec หลังจากนั้นทั้ง client และ server จะติดต่อสื่อสารกันโดยใช้ cipher spec อันใหม่ที่เพิ่งทำการตกลงกัน

SSL session อาจจะรวมการเชื่อมต่อ (connect ion) ที่ปลอดภัยหลาย ๆ อันด้วย นอกจากนี้การติดต่อสื่อสารก็อาจจะมีหลาย ๆ session ในเวลาเดียวกัน

Session state จะรวมด้วยสมาชิกต่อไปนี้

session indentifier

ลำดับของไบท์ที่ถูกเลือกโดย server เพื่อจะกำหนดถึง session state ที่ถูกเริ่มต้น

peer certificate

X509.v3 certificate ของ peer สมาชิกตัวนี้อาจจะว่างเปล่าก็ได้

Compression method

อัลกอริทึมที่ใช้ในการบีบอัดข้อมูลก่อนที่จะถูกเข้ารหัส

cipher spec

เป็นตัวกำหนดอัลกอริทึมในการเข้ารหัสข้อมูลจำนวนมาก เช่น DES , null เป็นต้น และอัลกอริทึมในการหา MAC เช่น MD5 , SHA มันยังสามารถกำหนดส่วนขยายในการทำการเข้ารหัส เช่น hash_size

master secret

48- byte secret ที่รู้ร่วมกันระหว่าง client และ server

is resumable

เป็น flag ที่ใช้บอกถึงว่า session นั้น ๆ สามารถที่จะถูกใช้ในการเริ่มต้นการเชื่อมต่อ (connection)ใหม่ได้หรือไม่

Connection state จะรวมด้วยสมาชิกดังต่อไปนี้

server and client random

ลำดับของไบท์ ที่ถูกเลือกโดย server และ client สำหรับแต่ละการเชื่อมต่อ

server write MAC secret

secret ที่ถูกใช้ใน MAC ที่กระทำบนข้อมูลที่เขียนโดย server

client write MAC secret

secret ที่ถูกใช้ใน MAC ที่กระทำบนข้อมูลที่เขียนโดย client

server write key

bulk cipher key สำหรับข้อมูลที่ถูกเข้ารหัสโดย server และถอดรหัสโดย client

client write key

bulk cipher key สำหรับข้อมูลที่ถูกเข้ารหัสโดย client และถอดรหัสโดย server

initialization vectors

เมื่อมีการใช้ block cipher ใน CBC mode initialization vectors (IV) จะถูกรักษาไว้แต่ key ในส่วนนี้จะถูกทำให้เริ่มต้นเป็นส่วนแรกโดย SSL handshake protocol หลังจากนั้น ciphertext block สุดท้ายของแต่ละ record จะถูกนำไปใช้กับ record ต่อไป

sequence numbers

ทั้ง client และ server จะรักษา sequence numbers ที่แยกกันของแต่ละส่วนไว้สำหรับการส่งหรือรับข้อความในแต่ละการเชื่อมต่อ เมื่อ client หรือ server ส่งหรือรับข้อความ change cipher spec แล้ว squence numbers ที่ถูกต้องจะถูกเซตให้เป็น 0

5.4.2 Record Layer

SSL Record Layer จะรับข้อมูลจากชั้นที่สูงกว่า ในรูปแบบของบล็อก (block) ที่ไม่ว่างเปล่าขนาดใดก็ได้

5.4.2.1 Fragmentation

Record layer จะแบ่งบล็อกข้อมูลไปสู่ SSLPlaintext records ซึ่งมีความยาว 2^{14} ไบท์ หรือน้อยกว่าขอบเขตของ client message ก็จะไม่ถูกคุ้มครองใน Record layer ยกตัวอย่าง เช่น ถ้ามี client message หลาย ๆ ข้อความเป็นของ Content type ที่เหมือนกัน อาจจะถูกรวมกันไปใน SSLPliantext records เพียงอันเดียวได้

```
Struct {
        Uint8 major , minor ;
} ProtocolVersion ;
enum {
        change_cipher_spec ( 20 ) , alert ( 21 ) , handshake ( 22 ) ,
        application_data ( 23 ) , ( 255 )
} ContentType ;
struct {
        ContentType type ;
        ProtocolVersion version ;
        Uint 16 length ;
        Opaque fragment [ SSLPlaintext.length ] ;
} SSLPlaintext ;
```

type

เป็นชื่อโปรโตคอลในระดับสูงกว่าที่ใช้ในการดำเนินการส่วนที่ถูกแยกออกมา

length

ความยาวในหน่วยไบท์ของ SSLPlaintext.fragment ที่ตามมา

fragment

ข้อมูลของแอพพลิเคชั่น ซึ่งข้อมูลนี้จะตรงไปตรงมา และจะถูกปฏิบัติในฐานะของบล็อกที่ไม่ขึ้นอยู่กับสิ่งใด (independent block) เพื่อที่จะถูกจัดการโดยโปรโตคอลในระดับที่สูงกว่าที่ถูกกำหนดไว้ในส่วนของ type

5.4.2.2 Record compression and decompression

Record ทั้งหมดจะถูกบีบอัดโดยใช้อัลกอริทึมในการบีบอัด ซึ่งถูกกำหนดไว้ใน current session state จะมีอัลกอริทึมในการบีบอัดที่ถูกใช้อยู่เสมอ แม้ว่าในตอนเริ่มแรกมันจะถูกกำหนดเป็น CompressionMethod.null อัลกอริทึมในการบีบอัดจะแปลงโครงสร้างของ SSLPliantext ไปสู่โครงสร้างของ SSLCompressed ฟังก์ชั่นในการบีบอัดจะลบข้อมูลเกี่ยวกับ state เมื่อไรก็ตามที่ cipher spec ถูกแทนที่ การบีบอัดจะต้องมีการสูญหายน้อยที่สุด และอาจจะไม่เพิ่ม content length เกินกว่า 1024 ไบท์ ถ้าฟังก์ชั่นในการคลายการบีบอัดพบ SSLCompressed.fragment ซึ่งเมื่อคลายการบีบอัดแล้วมีความยาวเกินกว่า 2^{14} ไบท์แล้ว มันก็ควรจะแสดงข้อความผิดพลาดแบบ fatal ออกมา

Decompression_failure **alert**

```
Struct {
        ContentType  type ;                    /* same as SSLPlaintext.type */
        ProtocolVersion  version ;             /* same as SSLPlaintext.version */
        uint 16  length ;
        opaque  fragment [ SSLCompressed.length ] ;
} SSLCompressed ;
```

Length

ความยาวในหน่วยไบท์ ของ SSLCompressed.fragment ที่ตามมา

fragment

รูปแบบที่ถูกบีบอัดแล้วของ SSLPlaintext.fragment

5.4.2.3 Record payload protection and the Cipher Spec

Record ทั้งหมดจะถูกป้องกันโดยใช้การเข้ารหัสและอัลกอริทึมของ MAC ที่ถูกกำหนดไว้ใน current CipherSpec ที่ถูกใช้อยู่ตลอดเวลา แม้ว่าในตอนเริ่มต้นมันจะถูกกำหนดเป็น SSL_NULL_WITH_NULL_NULL ซึ่งจะไม่มีการจัดเตรียมการรักษาความปลอดภัยให้ เมื่อการทำ handshake เสร็จสมบูรณ์ ทั้ง client และ server ก็จะรู้ถึง secret ร่วมกัน โดย sedret

นี้จะถูกใช้ในการเข้ารหัส record และคำนวณ keyed message autentication codes (MACs) บนเนื้อหาของ record เทคนิคที่ใช้ในการทำการเข้ารหัสและ การทำ MAC นั้นจะถูกกำหนดโดย Cipherspec และจะถูกบังคับโดย CipherSpec.cipher_type การเข้ารหัสและฟังก์ชั่น MAC จะแปลงจากโครงสร้างของ SSLCompressed ไปสู่โครงสร้างของ SSLCiphertext ฟังก์ชั่นในการถอดรหัสจะดำเนินการกลับกัน ในการส่งจะรวม sequence number เพื่อที่จะทำให้ข้อผิดพลาดข องข้อความ (missing) การเปลี่ยนแปลงของข้อความ (altered) และข้อความที่เพิ่มเข้ามา (extra message) สามารถถูกตรวจพบได้

```
Struct {
        ContentType type ;
        ProtocolVersion version ;
        Uint 16 length ;
        select ( CipherSpec.cipher_type ) {
                case stream : GenericStreamCipher ;
                case block  : GenericBlockCipher ;
        } fragment ;
} SSLCiphertext ;
```

type

ส่วนของ type ซึ่งจะเหมือนกับ SSLCompressed.type

version

ส่วนของ version ซึ่งจะเหมือนกับ SSLCompressed.version

length

ความยาวในหน่วยไบท์ของ SSLCiphertext.fragment ที่ตามมา

fragment

รูปแบบที่ถูกเข้ารหัสแล้วของ SSLCompressed.fragment และยังรวมไปถึง MAC อีกด้วย

5.4.2.3.1 Null or standard stream cipher

Stream cipher ซึ่งรวมไปถึง bulkcipherAlgorithm.null นี้จะแปลงไปมาระหว่างโครงสร้างของ SSLCompressed.fragment กับ stream ของโครงสร้าง SSLCiphertext.fragment

```
Stream-ciphered struct {
        Opaque content [ SSLCompressed.length ] ;
        opaque MAC [ CipherSpec.hash_size ] ;
} GenericStreamCipher ;
```

ส่วน MAC จะถูกสร้างขึ้นดังนี้

```
Hash ( MAC_write_secret + pad_2 +
        Hash ( MAC_write_secret_ + pad_1 + seq_num + length + content ) ) ;
```

เมื่อ " + " หมายถึงการนำมาต่อกัน (concatenation)

pad_1

ตัวอักษร 0 x 36 จะถูกทำซ้ำ 48 ครั้งสำหรับ MD5 และ 40 ครั้งสำหรับ SHA

pad_2

ตัวอักษร 0 x 5c จะถูกทำซ้ำในจำนวนที่เท่ากัน

seq_num

sequence number สำหรับข้อความนี้

hash

อัลกอริทึมในการ hashing ซึ่งได้รับจาก cipher suite

จะสังเกตเห็นได้ว่า เนื่องจาก MAC ถูกคำนวณก่อนที่จะทำการเข้ารหัส และ stream cipher จะเข้ารหัสบล็อกทั้งหมด ซึ่งรวมทั้ง MAC ด้วย สำหรับ stream cipher ซึ่งไม่ได้ใช้ synchronization vector (เช่น RC4) stream cipher state จากจุดสิ้นสุดของ record หนึ่งจะง่ายต่อการถูกใช้ใน packet ต่อมาถ้า CipherSuite เป็น SSL_NULL_WITH_NULL_NULL

การเข้ารหัสจะประกอบด้วยการกระทำที่เหมือนกันกับที่กำหนดไว้ (เช่น ข้อมูลที่ไม่ถูกเข้ารหัส และขนาดของ MAC เป็นศูนย์ บอกเป็นนัยว่าไม่มีการใช้ MAC) SSLCiphertext.length คือ SSLCompressed.length บวกกับ SSLCipherSpec.hash_size

5.4.2.3.2 CBC block cipher

สำหรับ block Cipher (เช่น RC2 หรือ DES) การเข้ารหัสและฟังชั่น MAC จะแปลงไปมาระหว่างโครงสร้าง SSLCompressed.fragmant กับโครงสร้าง SSLCiphertext.fra

```
Block-ciphered struct {
        opaque content [SSLCompressed.length];
        opaque MAC [CipherSpec.hash_size];
        unit8 padding [GennericBlockCipher.padding_length];
        unit8 padding_length;
} GenericBlockCipher;
```

การสร้าง MAC ได้กล่าวไว้แล้วในหัวข้อ 5.4.2.3.1

padding

padding จะถูกรวมเข้าไปเพื่อบังคับความยาวของ plaintext เพื่อให้เป็นจำนวนเท่าของความยาวของบล็อก cipher

padding_length

ความยาวของ padding จะต้องน้อยกว่าความยาวของบล็อก cipher และอาจเป็นศูนย์ได้ และความยาวของ padding ก็ควรจะเป็นจำนวนที่ทำให้ขนาดทั้งหมดของโครงสร้าง GenericBlockCipher เป็นจำนวนเท่าของความยาวบล็อก cipher

ความยาวของข้อมูลที่ถูกเข้ารหัสแล้ว (SSLCiphertext.length) จะเป็นจำนวนหนึ่งที่มากกว่าผลบวกของ SSLCompressed.length, CipherSpec.hash_size, และ padding_length

5.4.3. Change cipher spec protocal

change cipher spec protocal จะมีอยู่ในสัญญาณการส่งผ่านในการวางแผนเกี่ยวกับ cipher โปรโตคอลประกอบด้วยข้อความเพียงข้อความเดียว ซึ่งถูกเข้ารหัสแบบบีบอัดภายใต้ current CipherSpec (ไม่ใช่ pending) ข้อความประกอบด้วยไบท์เดียวซึ่งมีค่าเท่ากับ 1

```
Struct {
        enum { change_cipher_spec ( 1 ), ( 255 ) } type;
} changecipherspec;
```

ข้อความ change cipher spec จะถูกส่งโดยทั้ง client และ server เพื่อให้เป็นที่สังเกตสำหรับผู้ที่ได้รับ ว่า record ที่ตามหลังมาจากนี้นั้น จะถูกป้องกันภายใต้ CipherSpec และ keys ที่เพิ่งได้ตกลงกันไว้ การได้รับข้อความนี้ จะเป็นสาเหตุให้ผู้ได้รับต้องคัดลอก read pendding state ไปสู่ read current state โดยยังคงมีการแยก read state และ write state โดยทั้ง SSL client และ server อยู่ เมื่อ client หรือ server ได้รับข้อความ change cipher มันก็จะคัดลอก pending read state ไปสู่ current read state เมื่อ client หรือ server ได้รับข้อความ change cipher มันก็จะคัดลอก pending read state ไปสู่ current read state เมื่อ client หรือ server ได้เขียนข้อความ change cipher มันก็จะคัดลอก pending write state ไปสู่ current write state client จะส่งข้อความ change cipher spec หลังจาก handshake ข้อความ key exchange และข้อความ certificate varify (ถ้ามี) และ server ส่ง change cipher spec หลังจากมันประมวลผลข้อความ key exchange ที่มันได้รับมาจาก client ข้อความ change cipher spec ที่ไม่ถูกคาดหวังไว้ควรจะทำให้เกิด unexpected_message alert เมื่อมีการนำมาทำต่อ (resuming) ของ session ข้อความ change cipher spec จะถูกส่งหลังข้อความ hello message

5.4.4 Alert protocol

หนึ่งใน content type ที่ SSL Layer สนับสนุนคือ alert type ข้อความ alert message จะถ่ายทอดถึงความเข้มงวดของข้อความและคำอธิบายของการเตือน alert message ซึ่งมีระดับอยู่ในระดับของ fatal จะมีผลทำให้เกิดการหยุดการเชื่อมต่อโดยทันที ในกรณีนี้ การเชื่อมต่อ (connection) อื่น ๆ ที่ตรงกันกับ session นี้อาจจะดำเนินต่อไปได้ แต่ session identifier จะไม่ถูกยอมรับอีกต่อไป เพื่อป้องกัน session ที่ผิดพลาดจากการถูกนำไปใช้ในการเริ่มการเชื่อมต่อ

อันใหม่ เหมือนกับข้อความอื่น ๆ ข้อความ alert message จะถูกเข้ารหัสและบีบอัด โดยวิธีการที่ถูกกำหนดไว้ใน current connection state

```
enum { warning (1), fatal (2), (255) } Alert ever;
enum {
        close_notify (0),
        unexpected_message (10),
        bad_record_mac (20),
        decompression_failure (30),
        handshake_failure (40), no_certificate (41), bad_certificate (42),
                unsupported_certificate (43), certificate_revoked (46),
                certificate_expired (45), certificate_unknown (46),
                illegal_parameter (47),
        (255)
} AlertDescription;
struct {
        AlertLevel      level;
        AlertDescription        description;
} Alert ;
```

5.4.4.1 Closure alerts

Client และ server จะแบ่งการรับรู้เกี่ยวกับตอนจบของการเชื่อมต่อ เพื่อที่จะหลีกเลี่ยง truncation attack แต่ละส่วนอาจจะเริ่มแลกเปลี่ยน closing message

Close_notify

ข้อความนี้จะเป็นตัวบอกให้ผู้รับทราบว่า ผู้ส่งจะไม่ส่งข้อความอะไรอีก สำหรับการเชื่อมต่อนี้ session จะไม่สามารถนำกลับมาทำต่อได้ (unresumable) ถ้าการเชื่อมต่อใด ๆ ไม่ถูกหยุดลงด้วยข้อความ close_notify ที่ถูกต้องและระดับของมันจะเท่ากับการเตือน

5.4.4.2 Error alerts

การตรวจข้อผิดพลาดใน SSL Handshake protocol จะเป็นอย่างง่ายมาก เมื่อตรวจพบข้อผิดพลาดจะส่งข้อความไปสู่อีกด้านหนึ่ง แม้ว่าจะเป็นการส่งหรือการรับข้อความ fatal alert ทั้งสองด้านจะปิดการเชื่อมต่อนั้นทันที server หรือ client จะได้รับการร้องขอให้ลืม session-identifiers, keys, และ secrets ที่เกี่ยวข้องกับการเชื่อมต่อที่ล้มเหลวนั้น error alerts ต่อไปนี้จะถูกกำหนดไว้แล้ว

unexpected_message

เกิดจากการได้รับข้อความที่ไม่ถูกต้อง ข้อความเตือนนี้จะเป็นระดับ fatal เสมอและไม่ควรจะพบในการติดต่อระหว่างเครื่องมือที่ติดตั้งอย่างถูกต้อง

bad_record_mac

คำเตือนนี้จะถูกคืนกลับมา ถ้า record ที่ได้รับมา มากับ MAC ที่ไม่ถูกต้อง ข้อความนี้จะเป็นระดับ fatal เสมอ

decompression_failure

ฟังก์ชั่นในการคลายการบีบอัดได้รับอินพุทที่ไม่ถูกต้อง (เช่น ข้อมูลที่ถูกขยายเกินขอบเขตความยาวที่กำหนดไว้) ข้อความนี้จะเป็น fatal เสมอ

handshake_failure

การได้รับข้อความ handshake_failure alert เป็นตัวบ่งชี้ว่า ผู้ส่งไม่สามารถตกลงเกี่ยวกับเรื่องของพารามิเตอร์ข้อเสนอในการรักษาความปลอดภัยที่สามารถยอมรับได้ ข้อความนี้จะเป็นระดับข้อผิดพลาดของ fatal เสมอ

no_certificate

ข้อความนี้อาจจะส่งเพื่อตอบถึง certification request ถ้าไม่มี certificate ที่เหมาะสม

bad_certificate

certificate เกิดความไม่ถูกต้องขึ้น เช่น บรรจุลายเซ็นซึ่งไม่ถูกพิสูจน์อย่างถูกต้อง เป็นต้น

unsupported_certificate

certificate ไม่อยู่ในชนิดที่มีการสนับสนุน

certificate_revoked

certificate ถูกยกเลิกโดยผู้ที่เซ็นมัน

certificate_expired

certificate หมดอายุหรือไม่สามารถใช้การได้อย่างถูกต้องในตอนนี้

certificate_unknown

บางสิ่งปรากฏขึ้นในการประมวลผลเกี่ยวกับ certificate ซึ่งทำให้การเสนอมันนั้นไม่สามารถยอมรับได้

illegal_parameter

field ในการ handshake ซึ่งไม่อยู่ในขอบเขต หรือไม่สอดคล้องกับ field อื่น ข้อความนี้จะเป็นระดับ fatal เสมอ

5.4.5 Handshake protocol overview

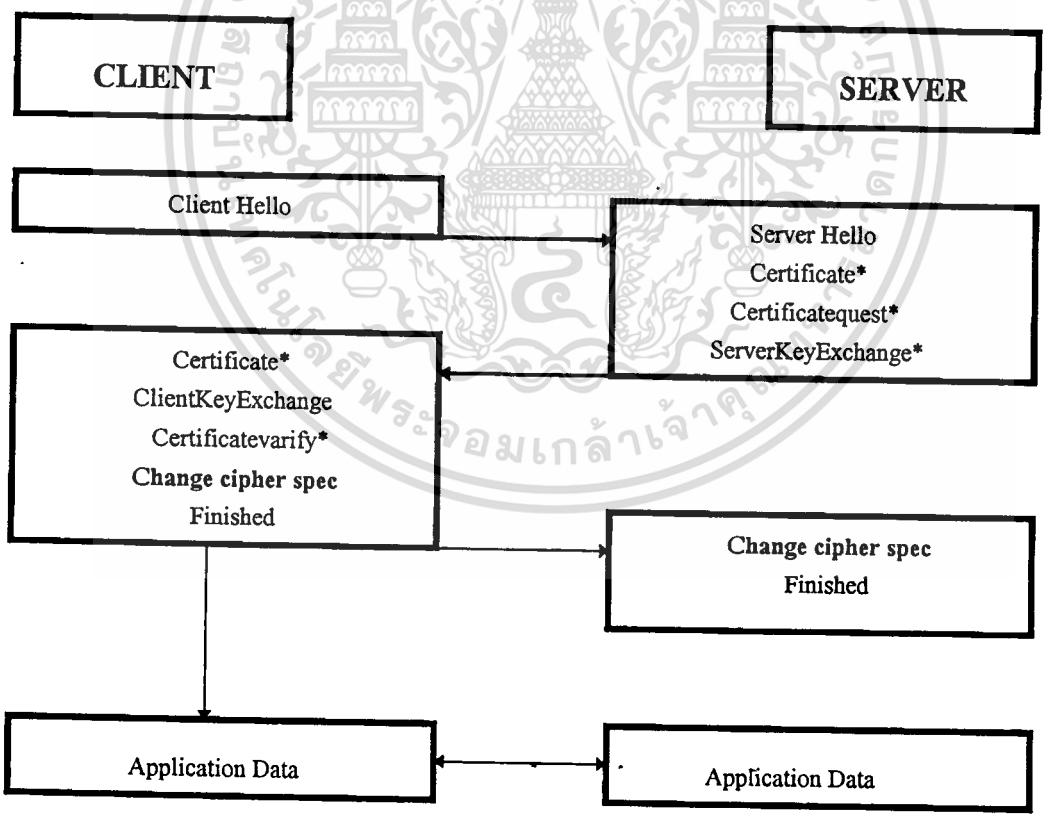
พารามิเตอร์ในการเข้ารหัสของ session state จะถูกสร้างโดย SSL Handshake Protocol ซึ่งจะทำงานอยู่บน SSL Record Layer เมื่อ SSL client และ server เริ่มการติดต่อสื่อสาร พวกมันจะต้องเห็นด้วยใน protocol version เลือกอัลกอริทึมในการเข้ารหัส พิสูจน์ความจริง (authenticate) ซึ่งกันและกัน แล้วใช้เทคนิคการเข้ารหัสแบบ public key เพื่อสร้าง secrets ที่ต้องถือไว้ร่วมกัน กระบวนการเหล่านี้จะถูกกระทำใน handshake protocol ซึ่งสามารถสรุปได้ดังต่อไปนี้

เมื่อ client ส่งข้อความ client hello ไป ซึ่ง server จะต้องตอบด้วยข้อความ server hello มิเช่นนั้นข้อผิดพลาดระดับ fatal จะเกิดขึ้นและการเชื่อมต่อนั้นจะล้มเหลว client hello และ server hello จะถูกใช้เพื่อเริ่มความสามารถในการปรับปรุงการรักษาความปลอดภัยระหว่าง client และ server client hello และ server hello เริ่มด้วย attribute ต่อไปนี้ protocol version, session ID , cipher suite และ compression method ค่าที่เกิดจากการ random 2 ค่า จะถูกสร้างขึ้นและแลกเปลี่ยนซึ่งกันและกัน คือ ClientHello.random และ ServerHello.random

ตามหลัง hello message ถ้า server ต้องถูกพิสูจน์ (authenticate) server จะส่ง certificate ของมันไป นอกจากนี้ ถ้าข้อความ server key exchange อาจจะถูกส่งไป ถ้ามันเป็นที่ต้องการ (เช่น ถ้า server ไม่มี certificate หรือถ้า certificate ของมันเป็นแบบสำหรับ signing เพียงอย่างเดียว) ถ้า server ถูกพิสูจน์ (authenticate) แล้ว มันอาจจะส่งสัญญาณเพื่อขอ certificate จาก client ถ้ามันเป็นสิ่งที่ถูกต้องสำหรับ cipher suite ที่ถูกเลือกมา

ถึงขั้นนี้ server จะส่งข้อความ certificate request client จะต้องส่ง certificate message หรือ ข้อความเตือน no certificate alert อย่างใดอย่างหนึ่ง ถึงตอนนี้ถ้าข้อความ client key exchange ถูกส่งและเนื้อหา (content) ของข้อความจะขึ้นอยู่กับอัลกอริทึมของ public key ที่ถูกเลือกระหว่าง client hello และ server hello ถ้า client ได้ส่ง certificate ซึ่งสามารถ signing ได้ digitally-signed certificate verify message จะถูกส่งเพื่อแสดงถึงการพิสูจน์ certificatd อย่างชัดแจ้ง

ถึงจุดนี้ ข้อความ change cipher spec จะถูกส่งโดย client และ client จะคัดลอก pending Cipher Spec ไปสู่ current Cipher Spec แล้ว client จะส่งข้อความ finished โดยทันที ภายใต้การใช้ algorithm, keys และ secrets อันใหม่ของมัน ณ จุดนี้ การ handshake จะเสร็จสมบูรณ์ แล้วทั้ง client และ server จะเริ่มแลกเปลี่ยนข้อมูลในชั้นแอพพลิเคชั่น (application layer data)

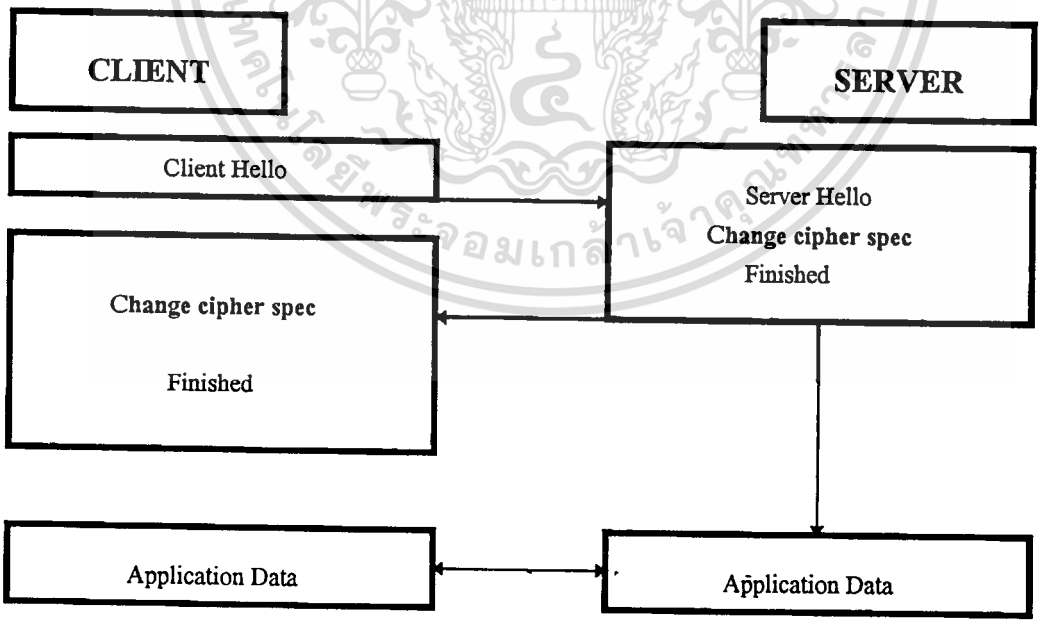


รูปที่ 5.2 แสดงถึงการทำงานของ client และ server ระหว่างการ handshake

จะพบว่า เพื่อหลีกเลี่ยงการหยุดกลางคันของ pipeline ChangeCipherSpec เป็น content type ของ SSL Protocol ที่เป็นอิสระ และไม่ใช่ข้อความของ SSL handshake อย่างแท้จริงนัก

เมื่อ client และ server ได้ตัดสินใจที่จะทำต่อ (resume) session ที่เคยทำแล้ว หรือคัดลอก session ที่มีอยู่แล้ว (แทนที่จะตกลงเกี่ยวกับพารามิเตอร์ในการรักษาความปลอดภัยใหม่) จะมีการไหลของข้อความต่อไปนี้

เมื่อ client ส่ง client hello โดยที่ใช้ Session ID ของ session ที่ต้องการจะทำต่อไปนั้น (resume) server จะตรวจสอบ session cache ของมันว่าตรงกับที่ต้องการหรือไม่ ถ้าตรงกัน และ server เต็มใจที่จะเริ่มการเชื่อมต่อใหม่อีกครั้งภายใต้ session state ที่ได้ระบุไว้ มันจะส่ง server hello ที่มีค่า session ID เดียวกัน ณ จุดนี้ ทั้ง client และ server จะต้องส่งข้อความ change cipher Spec และปฏิบัติตรงไปสู่ข้อความ finished เมื่อการเริ่มต้นใหม่เสร็จสมบูรณ์ ทั้ง client และ server อาจจะเริ่มแลกเปลี่ยนข้อมูลในระดับชั้นแอพพลิเคชั่น (application layer data) ถ้า session ID ไม่ตรงกันกับที่อยู่ใน cache server จะสร้าง session ID ค่าใหม่ และ SSL client และ server จะทำการ handshake แบบเต็มต่อไป



รูปที่ 5.3 แสดงการทำ handshake แบบ resume session

5.4.6 Handshake Protocol

SSL Handshake Protocol เป็นหนึ่งใน client ระดับสูงกว่า (higher level clients) ที่ถูกกำหนดไว้แล้วใน **SSL Record Protocol** โปรโตคอลนี้ จะถูกใช้ในการตกลงเกี่ยวกับ attributes เกี่ยวกับความปลอดภัยของ session ข้อความ Handshake จะถูกเสนอให้จาก SSL Record layer ที่พวกมันถูกครอบด้วยโครงสร้างของ SSL Plaintext หนึ่งหรือมากกว่านั้น ซึ่งจะถูกดำเนินการและขนส่งดังที่ได้กำหนดไว้แล้วโดย current active session state

```
                    Enum {
        Hello_request (0), client hello (1), serverhello (2),
        Certificate (11), server_key_exchange (12), certidicate_request (13),
        Server_hello_done (14), certificate_verify (15), client_key_exchange (16),
        Finished (20), (255)
} HandshakeType;
struct {
        HandshakeType       msg_type;
        Uint24              lentth
        Select              ( HandshakeType )
                Case    hello_request:      HelloRequest;
                Case    client_hello:       CleientHello;
                Case    server_hello:       ServerHello;
                Case    certificate:        Certificate;
                Case    server_key_exchange:    ServerKeyExchange;
                Case    certificate_request:    CertificateRequest;
                Case    server_hello_done:      ServerHelloDone;
                Case    certificate_varify:     CertificateVerify;
                Case    client_key_exchange:    ClientKeyExchange;
                Case    finished:               Finished;
        } body;
```

} Handshake;

ข้อความ handshake protocol จะถูกนำเสนอในลำดับที่พวกมันจะต้องส่งตามนั้น การส่งข้อความ handshake ในลำดับที่ไม่คาดหวังจะเป็นผลทำให้เกิด fatal error

5.4.6.1 Hello message

ข้อความในช่วงของการ hello จะถูกใช้สำหรับการแลกเปลี่ยนความสามารถในการปรับปรุงรักษาความปลอดภัยระหว่าง client และ server เมื่อ session เริ่มขึ้น CipherSpec encryption, hash และ compression algorithms จะเริ่มต้นเป็น null current CipherSpec จะถูกใช้เป็นข้อความสำหรับการตกลงกันใหม่

5.4.6.1.1 Hello request

ข้อความ Hello request อาจจะถูกส่งโดย server ณ เวลาใดก็ได้ แต่อาจจะถูกเพิกเฉยโดย client ถ้า handshake protocol กำลังทำอยู่แล้ว มันสังเกตได้โดยง่ายว่า เมื่อ client ต้องการที่จะเริ่มกระบวนการในการทำความตกลงใหม่อีกครั้งหนึ่ง โดยการที่มันส่งข้อความ hello message เมื่อสะดวก

จะพบว่า เพราะว่าข้อความ handshake จะถูกส่งก่อนข้อมูลของแอพพลิเคชั่น มันจึงเป็นสิ่งที่คาดหวังว่าการทำความตกลงไม่ควรจะเริ่มเกิน 1 หรือ 2 ครั้งของเวลาในการส่งข้อความของข้อมูลที่ยาวที่สุดของแอพพลิเคชั่น

หลังจากส่ง hello request server ไม่ควรจะทำการขอซ้ำ จนกระทั่งการทำความตกลงในการ handshake เสร็จสมบูรณ์ client ที่ได้รับ hello request ในขณะที่กำลังทำความตกลงในการ handshake state ควรจะเพิกเฉยต่อข้อความนั้น

โครงสร้างของข้อความ hello request คือ

```
Struct { } HelloRequest ;
```

5.4.6.1.2 Client hello

เมื่อ client ทำการติดต่อครั้งแรกถึง server มันจะถูกต้องการให้ส่งข้อความ client hello เป็นข้อความแรก client ยังสามารถส่งข้อความ client hello เพื่อตอบข้อความ hello request หรือปีนความคิดริเริ่มของมันเองเพื่อที่จะทำความตกลงกันใหม่เกี่ยวกับค่าพารามิเตอร์ของการรักษาความปลอดภัยในการเชื่อมต่อที่มีอยู่แล้ว

ข้อความ client hello ยังรวมถึงโครงสร้างที่เป็นการ random ซึ่งจะมีการนำไปใช้ภายหลังในโปรโตคอลนั้น

```
Struct {
    Uint32      gmt_unix_time;
    Opaque      random_bytes [28];
} Random;
```

Gmt_unix_time

เป็นเวลาและวันที่ในปัจจุบันซึ่งจะเกิดจาก clock ภายในของผู้ส่ง และอยู่ในรูปของมาตรฐานรูปแบบ 32-bit ของ Unix โดย SSL จะไม่จำเป็นที่จะต้องเซตค่า clock ให้ถูกต้องขึ้นอยู่กับโปรโตคอลในระดับสูงกว่า หรือแอพพลิเคชั่นโปรโตคอลที่จะกำหนดค่าเพิ่มเติมเอง

random_bytes

28 ไบต์ ที่สร้างโดย secure random number generator

ข้อความ client hello จะรวมค่า session identifier ซึ่งมีค่าความยาวที่ไม่คงที่ ถ้าไม่ว่างเปล่า ค่าที่เป็นตัวกำหนด (identify) session ระหว่าง client และ server เดียวกัน จะเป็นค่าที่พารามิเตอร์ในการรักษาความปลอดภัยที่ client ต้องการนำกลับมาใช้ใหม่ session identifier อาจจะได้จากการเชื่อมต่อที่เคยผ่านมาแล้ว การเชื่อมต่อในครั้งนี้ หรือ การเชื่อมต่ออันอื่นที่กำลังเชื่อมต่ออยู่ในปัจจุบัน ข้อเสนออันที่สองจะเป็นประโยชน์ ถ้า client ต้องการเพียงแค่ update โครงสร้างที่เป็นการ random และรับค่ามาจากการเชื่อมต่อนั้น ขณะที่ข้อเสนออันที่สาม ทำให้มีความเป็นไปได้ที่จะเริ่มการเชื่อมต่ออย่างปลอดภัยหลาย ๆ การเชื่อมต่อในเวลา

โครงสร้างทั้งหมดของ client hello จะเป็นดังต่อไปนี้

```
                    Struct {
    ProtocolVersion     client_version;
    Random              random;
    SessionID           session_id;
    CipherSuite         cipher_suites<2..2^16-1>;
    ComproessionMethod  compression_methods<1..2^8-1>;
} ClientHello;
```

Client_version

เป็นเวอร์ชั่นของโปรโตคอล SSL ที่ client ต้องการจะใช้ในการติดต่อสื่อสารระหว่าง session นี้

random

โครงสร้างที่เป็นการ random ซึ่งสร้างโดย client โดยจะนำไปใช้กับ ServerHello.random เพื่อสร้าง master secret (มีขนาด 48 บิต) สำหรับใช้ในทั้ง client และ server นั้น โดยค่า master secret นี้ จะถูกนำไปใช้ในการเข้ารหัสและคำนวณหา MAC ต่อไป

session_id

ID ของ session ที่ client ต้องการใช้สำหรับการติดต่อสื่อสารในการเชื่อมต่อนี้ ส่วนนี้ควรจะว่างเปล่า ถ้าไม่มี session_id อยู่ หรือ client ต้องการที่จะสร้างพารามิเตอร์ในการรักษาความปลอดภัยอันใหม่

Cipher_suites

เป็นรายชื่อของข้อเสนอในการจัดการการเข้ารหัส ซึ่งสนับสนุนโดย client เรียงจากรายชื่อที่ client ที่ต้องการให้ใช้มากที่สุดก่อน ถ้าส่วนของ session_id ไม่ว่างเปล่า (ซึ่งเป็นการบอกให้ทราบว่า มันเป็นการขอร้องขอที่จะนำ session ที่มีอยู่มาทำต่อ (resume session) field นี้จะต้องรวม cipher_suite จาก session นั้นด้วยเป็นอย่างน้อย

compression_methods

เป็นรายชื่อของวิธีการในการบีบอัดซึ่งสนับสนุนโดย client เรียงตามลำดับความต้องการก่อนหลังเช่นกัน ถ้าส่วนของ session_id ไม่ว่างเปล่า (เป็นการบอกให้ทราบว่า มันเป็นการร้องขอที่จะนำ session ที่มีอยู่มาทำต่อ (resume session) filed นี้จะต้องรวมcipher_suite จาก session นั้นด้วยเป็นอย่างน้อย

หลังจากส่งข้อความ client hello นั้นแล้ว client จะรอสำหรับข้อความ server hello ถ้ามีการส่งข้อความในการทำ handshake ใด ๆ นอกจาก hello request แล้ว จะมีผลทำให้เกิดข้อผิดพลาดระดับ fatal และข้อมูลของแอพพลิเคชั่นไม่ควรจะถูกส่งก่อนข้อความ finished ได้ส่งไปแล้ว ข้อมูลของแอพพลิเคชั่นที่ถูกส่งมาก่อนได้รับข้อความ finished ที่ถูกต้อง จะเป็นที่รู้กันว่ายังไม่มีความปลอดภัยเพียงพอ

5.4.6.1.3 Server hello

Server จะประมวลผลข้อความ client hello และตอบกลับไปด้วยข้อความ handshake_failure alert หรือ server hello อย่างใดอย่างหนึ่ง

Field นี้ จะบรรจุเวอร์ชั่นที่ต่ำที่สุดที่เสนอโดย client ใน client hello และเวอร์ชั่นที่สูงที่สุดที่สนับสนุนโดย server

```
Struct {
        ProtocolVersion     server_version;
        Random              random;
        SessionID           session_id;
        CipherSuite         cipher_suite;
        CompressionMethod   compression_method;
} ServerHello;
server_version
```

Random

โครงสร้างนี้จะถูกสร้างโดย server และต้องแตกต่างจาก (หรือเป็นอิสระจาก) ClientHello.random เพื่อนำไปสร้าง master secret

session_id

จะเป็นเอกลักษณ์ของ session ที่สอดคล้องกับการเชื่อมต่อนั้น ถ้า ClientHello.session_id ไม่ว่างเปล่า server จะค้นหาใน session cache เพื่อดูว่า ถ้าค่าตรงกัน และ server เต็มใจที่จะเริ่มการเชื่อมต่อใหม่ด้วยการใช้ session state ที่ได้กำหนดไว้แล้ว server จะตอบด้วยค่าเดียวกับที่ client เสนอให้ ซึ่งจะเป็นการบ่งชี้ว่าเกิดการ resume session และจะสั่งให้ทั้งสองด้านกระทำตรงไปสู่ข้อความ finished ถ้าไม่พบ filed นี้จะบรรจุค่าที่แตกต่างออกไป เพื่อเป็นการชี้ว่ามันเป็น new session server จะคืนค่า empty session_id เพื่อบ่งชี้ว่า session นั้นจะไม่ถูก cache ไว้ ซึ่งจะไม่สามารถ resume ได้

cipher_suite

cipher suite เพียงอันเดียวที่ถูกเลือกโดย server จากรายชื่อใน ClientHello.cipher_suites สำหรับ resumed session field นี้จะเป็นค่าจาก state ของ session ที่ถูก resume

compression_method

อัลกอริทึมที่ใช้ในการบีบอัดเพียงอันเดียวที่ถูกเลือกโดย server จากรายชื่อใน ClientHello.compression_methods สำหรับ resumed session fieldนี้จะเป็นค่าจาก state ของ session ที่ถูก resume

5.4.6.2 Server certificate

Server ที่จะต้องถูกพิสูจน์ความแท้จริง (authenticate) จะส่ง certificate ของมันตามหลังข้อความ server hello อย่างทันที ชนิดของ certificate จะต้องเหมาะสมกับอัลกอริทึมในการแลกเปลี่ยน cipher suite's key ที่ถูกเลือกไว้ ซึ่งโดยทั่วไปจะเป็น X.509.v3 certificate ชนิดของข้อความที่เหมือนกันจะถูกใช้สำหรับการตอบของ client ต่อข้อความ server certificate request

```
Opaque ASN.1Cert<1..2^24-1>;
Struct {
        ASN.1Cert certificate_list<1..2^24-1>;
} Certificate
```

Certificate_list จะเป็นลำดับ (chain) ของ X.509.v3 certificate ซึ่งเรียงลำดับจาก certificate ของผู้ส่งก่อน ไปจนถึง root certificate authority ในท้ายที่สุด

5.4.6.3 Server key exchange message

ข้อความ server key exchange จะถูกส่งโดย server ถ้า server ไม่มี certificate หรือมี certificate ที่ใช้สำหรับ signing เพียงอย่างเดียว

5.4.6.4 Certificate request

Server ที่มีชื่อเสียง (non anonymous server) สามารถที่จะขอ certificate จาก client ได้ ถ้าไม่ขัดกับ cipher suite ที่ได้เลือกไว้

Certificate_types

```
Opaque CertificateAuthority<0..2^24-1>;
Enum {
        Ras_sign (1), dss_sign (2), ras_fixed_dh (3), ss_fixed_dh (4),\
        Rsa_ephermeral_dh (5), dss_ephermeral_dh (6), fortezza_dms (20), (255)
} ClientCertificateType;
opaque DistinguishedName<1..2^16-1>;
struct {
        ClientCertificateType certificate_type<1..2^8-1>;
        DistinguishedName certificate_authorities<3..2^16-1>;
} CertificateRequest;
```

Field นี้เป็นรายชื่อของชนิดของ certificate ที่ถูกต้องการ เรียงลำดับจากรายชื่อที่ server ต้องการมากที่สุดก่อน

Certificate authorities

รายชื่อของ certificate authority ที่มีชื่อเสียง ซึ่งสามารถยอมรับได้ โดยรายชื่อเหล่านี้ได้รับจาก [X509]

ถ้า anonymous server ต้องการพิสูจน์เอกลักษณ์ของ client จะเกิดข้อความเตือน fatal handshake_failure ขึ้น

5.4.6.5 Server hello done

ข้อความ server hello done จะถูกส่งโดย server เพื่อชี้ว่าจบข้อความ server hello และข้อความที่เกี่ยวข้อง หลังจากส่งข้อความนี้แล้ว server จะรอสำหรับการตอบจาก client

```
Struct { } ServerHellodone;
```

เมื่อได้รับข้อความ server hello done client ควรตรวจสอบว่า server ได้เสนอ certificate ที่ใช่ได้หรือไม่ และควรตรวจสอบว่า พารามิเตอร์ใน server hello เป็นที่ยอมรับได้หรือไม่

5.4.6.6 Client certificate

จะเป็นข้อความแรกที่ client สามารถส่งไปได้หลังจากได้รับข้อความ server hello done ข้อความนี้จะถูกส่งเมื่อ server ต้องการ certificate เท่านั้น ถ้าไม่มี certificate ที่เหมาะสม client ควรส่งข้อความเตือน no certificate แทน ซึ่งข้อความนี้เป็นข้อผิดพลาดในระดับ warning เท่านั้น อย่างไรก็ตาม server อาจจะตอบด้วย fatal handshake failure ก็ได้ ถ้าการพิสูจน์ client เป็นที่ต้องการอย่างยิ่ง

5.4.6.7 Client key exchange message

ตัวเลือกของข้อความขึ้นอยู่กับการเลือกอัลกอริทึมของ public key ซึ่งอยู่ใน KeyExchangeAlgorithm

```
Struct {
    Select (KeyexchanbeAlgorithm) {
        Case rsa : EncryptedPreMasterSecret;
        Case diffie_helloman : ClientDiffieHellmanPublic;
        Case fortezza_dms : FortezzaKeys;
```

ข้อมูลเกี่ยวกับการเลือกโครงสร้าง record ที่ถูกต้องอยู่ใน pending session state

5.4.6.8 Certificate verify

ข้อความนี้จะถูกใช้เพื่อเสนอให้เป็นการพิสูจน์ของ client certificate อย่างชัดเจน ข้อความนี้จะถูกส่งตามหลัง client certificate ที่มีความสามารถในการ signing เท่านั้น

```
Struct {
        Signature signature;
} CertificateVerify;
```

CertificateVerify.signature.md5_hash

MD5 (master_secret + pad2 + MD5 (handshake_message +

master_secret + pad1));

Certificate.signature.sha_hash

SHA (master_secret + pad2 + SHA (handshake_messages +

Master_secret + pad1));

Handshake_message ในที่นี้ หมายถึง ข้อความ handshake ทั้งหมดเริ่มตั้งแต่ client hello จนถึงข้อความนี้แต่ไม่รวมข้อความนี้

5.4.6.9 Finished

ข้อความ finished จะถูกส่งทันทีหลังจากข้อความ change cipher specs เพื่อเป็นการพิสูจน์ว่ากระบวนการในการแลกเปลี่ยน key และการพิสูจน์ความแท้จริง (authentication) ได้ทำสำเร็จเรียบร้อยแล้ว ข้อความ finished นี้ จะเป็นข้อความแรกที่ถูกป้องกันโดยอัลกอริทึม , keys และ secretsที่เพิ่งได้ทำการตกลงกัน ข้อความ finished นี้จะไม่มีข้อมูลที่ถูกต้องการอะไรเลย ผู้ติดต่ออาจจะเริ่มการส่งข้อมูลที่เป็นความลับโดยทันทีหลังจากส่งข้อความ finished แล้ว
ผู้รับข้อความ finished ต้องตรวจสอบว่าเนื้อหาที่ได้รับ ถูกต้องหรือไม่

```
enum {  client(0x434C4E54), server (0x53525625) } Sender;
struct {
        opaque   md5_hash [16];
        opaque   sha_hash [20];
} Finished;
```

Md5_hash

MDS (master_secret + pad2 + MDS (handshaake_messages +
sender + master_secret + pad1));

sha_hash

SHA (master_secret + pad2 + SHA (handshaake_messages +
sender + master_secret + pad1));

ฟังก์ชัน hash ที่อยู่ในข้อความ finished ที่ถูกส่งโดย sever จะทำงานร่วมกับ Sender.server และทั้งหมดที่ถูกส่งโดย client จะทำงานร่วมกับ Sender.client ค่าของ handshake_messages จะรวมข้อความ handshake ทั้งหมด เริ่มตั้งแต่ client hello จนถึง (แต่ไม่รวม) ข้อความ finished ซึ่งอาจจะแตกต่างจาก handshake_massages ในหัวข้อ 6.8 เพราะว่ามันอาจจะรวมข้อความ certificate verify (ถ้ามีการส่ง) ข้อความ change cipher spec ไม่ใช่ข้อความ handshake และไม่ถูกรวมอยู่ในการคำนวณค่า hash

5.4.7 Application data protocal

ข้อความ Application data จะถูกส่งโดย Record Layer และจะถูกแยกส่วน บีบอัด และเข้ารหัส โดยขึ้นอยู่กับ current connection state ข้อความจะถูกปฏิบัติดังเช่น transparent data สำหรับ **Record Layer**

สรุป

ในการรักษาความปลอดภัยในการใช้งาน WWW นั้น จะมีโปรโตคอลหลายตัวที่นิยมใช้งานกันอย่างกว้างขวาง เช่น S-HTTP เป็นต้น หรือว่ามาตรฐานที่จะเกิดขึ้นในอนาคต เช่น SET เป็นต้น โปรโตคอล SSL ก็น่าจะมีการใช้กันอย่างกว้างขวาง ด้วยคุณสมบัติที่มันไม่ขึ้นกับโปรโตคอล ในระดับชั้นสูงกว่า เช่น ในชั้น Application ซึ่งทำให้มันเกิดเป็นมาตราฐานที่ยอมรับการรับส่งข้อมูลที่ต้องการความปลอดภัยในปัจจุบัน

อย่างไร ก็ตามในการใช้งานของ SSL นั้น จะต้องมีการสนับสนุนโปรโตคอล SSL ทั้งทางด้าน client และ server ซึ่งอาจจะถูกมองว่าเป็นการลงทุนที่เพิ่มขึ้น ถ้ามีการรักษาความปลอดภัยโดยโปรโตคอลตัวอื่นอยู่แล้ว อย่างไรก็ตามด้วยคุณสมบัติเด่นของมันหลายประการทำให้คาดเดาว่า น่าจะมีการใช้งานโปรโตคอล SSL อย่างแพร่หลายต่อไป

บทที่ 6
โครงสร้างและการเขียน Home page ด้วยภาษา HTML และภาษา JAVA

6.1. แนะนำภาษา HTML

HTML (HyperText Markup Language) เป็นภาษาที่มีพื้นฐานมาจากภาษา SGML (Standard Generalized Markup Language) โดยมีความสามารถในการเชื่อมต่อกับเอกสารอื่น ๆ ในลักษณะไฮเปอรเท็กซ์ (Hypertext) ที่ทำให้การเข้าถึงเอกสารอื่น ๆ ในเครื่องเดียวกัน หรือเอกสารบนเครื่องอื่น ๆ ในสถานที่ต่าง ๆ จากภายในเอกสารนั้นทำได้อย่างรวดเร็ว และเอกสารจะแสดงออกมาเป็นหน้า เอกสารในหน้าแรกซึ่งแสดงทุกครั้งที่เริ่มเข้าสู่โปรแกรม จะถูกเรียกว่า Home page ซึ่งเอกสารที่เขียนขึ้นมาตามรูปแบบของ HTML นั้น สามารถนำเสนอได้ในลักษณะของ ข้อความ ภาพนิ่ง ภาพเคลื่อนไหว หรือเสียงได้อีกด้วย แต่ในการที่จะแสดงเอกสารออกมาได้นั้นจำเป็นต้องอาศัย Application Program ที่สามารถ Run อยู่บนระบบ www ของอินเตอร์เน็ตได้ ที่เรียกว่าบราวเซอร์ ซึ่งโปรแกรมบราวเซอร์นี้เป็นส่วนของเครื่อง Client เพื่อเข้าไปเรียกดูข้อมูลจาก www Server ต่าง ๆ โดยการใช้ URL เพื่อกำหนดส่วนที่อยู่ของข้อมูล

6.2 โครงสร้างของ HTML

ลักษณะทั่วไปของ HTML ก็เช่นเดียวกับการเขียนโปรแกรมที่มีรูปแบบของคำสั่งหรือฟังก์ชั่นที่ใช้โดยทั่วไป ซึ่งคำสั่งที่ใช้ใน HTML จะอยู่ในรูป

<Tag_Name> affected text </ Tag_Name>

Tag แต่ละตัวประกอบขึ้นจาก เครื่องหมายน้อยกว่า (<) ตามด้วยชื่อของ Tag นั้นและปิดท้ายด้วยเครื่องหมายมากกว่า (>) การใช้ Tag จะใช้เป็นคู่ ๆ โดย Tag ตัวแรกจะเรียกว่า Tag เริ่มต้น (Starting Tag) ส่วนที่เหลือจะเรียกว่า Tag จบ (Ending Tag) โดย Tag จบจะต้องมีเครื่องหมาย "/" อยู่หน้าชื่อของ Tag จบ ต่อจากเครื่องหมายน้อยกว่า เช่น

< title> My Useful Document </title>

ซึ่งเป็นการบอกแก่ Browser ให้รู้ว่าข้อความ My Useful Document นี้เป็นชื่อเรื่องของเอกสารนี้เท่านั้น

หมายเหตุ

มีบาง Tag ที่ไม่ต้องใช้เป็นคู่ คือไม่ต้องมี Tag จบ เช่น Tag ซึ่งเป็นการขึ้นบรรทัดใหม่โดยที่ไม่ต้องใช้ Tag

6.3 รูปแบบและหน้าที่ของ Tag

Title

รูปแบบ : <Title> Name of Document </Title>

หน้าที่ : ใช้ในการกำหนดชื่อของเอกสาร

หมายเหตุ

ชื่อของ Title จะมีความยาวจำกัดประมาณ 6 คำ และจะถูกแสดงแยกต่างหากจากส่วนอื่น ๆ ของ Document โดยจะถูกแสดงไว้ในส่วน Windows title bar ของ Browser นั้น

Headings

รูปแบบ : <Hy> Text of Heading </Hy>

โดยที่ y เป็นตัวเลขตั้งแต่ตัวเลข 1 ถึง 6

หน้าที่ : , ทำให้ข้อความที่เป็นหัวข้อ แตกต่างจากข้อความธรรมดา ซึ่งหัวข้อของ HTML Document มีทั้งหมด 6 ระดับ เริ่มจากระดับที่ 1 จนถึงระดับที่ 6 โดยในส่วนของ Tag Heading นี้จะแสดง text ออกมาในรูปแบบของอักขระขนาดใหญ่และหนา โดยที่ระดับที่ 1 จะมีขนาดใหญ่สุด และลดขนาดลงไปเรื่อย ๆ จนถึงระดับที่ 6 ซึ่งเป็นระดับที่เล็กที่สุด

ตัวอย่าง

<H1> Heading1 </H1>

<H2> Heading2 </H2>

ผลลัพธ์

Heading1

Heading2

Paragraph

HTML Document จะแตกต่างจาก Document ของ Word Processor ทั่วไปคือ HTML Document จะไม่ถือว่า Carriage return สำคัญ ดังนั้นการทำ Word Wrapping หรือการตัดคำเพื่อขึ้นบรรทัดใหม่โดยอัตโนมัติ จะสามารถเกิดขึ้นได้ทุกจุดใน HTML Document

รูปแบบ : <p>

หน้าที่ : ใช้ในการขึ้นบรรทัดใหม่ โดยเว้นบรรทัด 1 บรรทัด

หมายเหตุ

ช่องว่าง (space) หลาย ๆ ช่องว่างจะถูก collapsed ให้เหลือเพียง 1 spece เท่านั้น

การเชื่อมโยงกับ HTML Document อื่น ๆ

การทำให้ข้อความที่เราต้องการสามารถเชื่อมโยงไปหาเอกสารที่เกี่ยวข้องได้นั้น สามารถทำได้โดยใช้ Tag <a> ซึ่งข้อความเหล่านี้จะแตกต่างจากข้อความทั่วไปคือ เป็นข้อความสีฟ้า ซึ่งถ้าลากเมาส์ไปในบริเวณข้อความนี้จะทำให้สัญญลักษณ์ลูกศรของเมาส์เปลี่ยนเป็นสัญลักษณ์รูปมือแทน ซึ่งมีวิธีการใช้ดังนี้

1. พิมพ์ <a ตามด้วยช่องว่าง หนึ่งช่อง
2. พิมพ์ชื่อของ Doccment ที่ต้องการเชื่อมโยงไประหว่าง “ ” ดังนั้น HERF=“File_Name และตามด้วย >
3. พิมพ์ข้อความที่ต้องการ
4. พิมพ์ Tag จบ

รูปแบบ : <a HERF=”File_Name”> Text

ตัวอย่าง : <a HERF=”Mainestats.html”>Maine

จากตัวอย่า'นี้ คำว่า Maine จะสามารถเชื่อมโยงไปยังแฟ้ม Mainestats.html ได้ ซึ่งแฟ้มนี้จะต้องอยู่ใน directory ปัจจุบัน แต่ถ้าแฟ้มนี้อยู่ใน directory อื่นๆ เราจะต้องบอกถึง path ที่ถูกต้องของแฟ้มนี้ด้วย เพราะฉะนั้นเพื่อความถูกต้องไม่ว่าเราจะอยู่ path ไหนก็ตามก็ควรบอก path แบบสมบูรณ์ของแฟ้มนั้นลงไป เช่น ถ้าแฟ้ม mainstats.html อยู่ใน directory AtlantioStates จะสามารถเขียนได้ดังนี้

<a HERF= "/AtlanticStates/Mainestats.html">Maine

Address

รูปแบบ : <address> Name of Address </address>

หน้าที่ : ใช้ในการกำหนดว่าข้อความในระหว่าง Tag address นี้เป็นที่อยู่ ซึ่งอาจจะเป็นที่อยู่ของผู้แต่ง ซึ่งเราสามารถติดต่อได้ โดย browser จะแสดงออกมาในลักษณะตัวอักขระเอียง (Italio)

ตัวอย่าง

<address> A beginner's Guide to HTML / NCSA / pubs@ncsa.uiuc.edu </address>

ผลลัพธ์

A beginner's Guide to HTML HTML / NCSA / pubs@ncsa.uiuc.edu

Character Formatting

หน้าที่ : เป็นการกำหนดรูปแบบของตัวอักขระที่จะให้ browser แสดงออกมา ซึ่งมี Tag ที่ใช้กำหนดอยู่ด้วยกันหลายแบบ ดังนี้

	แสดงข้อความในลักษณะตัวเอียงและหนา
	แสดงข้อความในลักษณะตัวหนา
<b>	แสดงข้อความในลักษณะเช่นเดียวกับ Tag
<I>	แสดงข้อความในลักษณะตัวเอียง
<blink>	แสดงข้อความในลักษณะที่สามารถกระพริบได้

ตัวอย่าง

 This is the first sentence.

 This is the second sentence.

<b> This is the third sentence. <b>

<I> This is the fourth sentence. <I>

ผลลัพธ์

This is the first sentence.

This is the second sentence.

This is the third sentence.

This is the fourth sentence.

Inline Image

คือ การแสดงภาพที่เราต้องการ โดย Browser ต่าง ๆ จะแสดงภาพที่มีส่วนขยายเป็น .bmp (X bitmap) และ .gif ซึ่งมีรูปแบบในการใช้ ดังนี้ <IMG SRC = "image_URL"> และถ้าต้องการกำหนดตำแหน่งของภาพว่าจะให้อยู่ส่วนไหนของ page ก็สามารถที่จะทำได้ โดยเพิ่มข้อความ ALIGN = Left หรือ ALIGN = Right ลงไปใน Tag แต่ถ้าต้องการให้ข้อความที่อยู่ต่อจากรูปภาพอยู่ในระดับใหนของรูปภาพ ก็ให้เพิ่ม option top , option middle หรือ option bottom เข้าไปในข้อความ ALIGN และข้อความนี้จะมี default เป็น bottom

เส้นใต้

รูปแบบ : <hr>

หน้าที่ : ใช้ในการขีดเส้นตรง

ตัวอย่าง

This is the line. <hr>

ผลลัพธ์

This is the line.

การทำให้ข้อความหรือรูปภาพอยู่ตรงกลาง Page

รูปแบบ : Text

หน้าที่ : ทำให้ข้อความหรือรูปภาพอยู่กลางหน้าเอกสาร

การสร้าง Black Ground ของเอกสาร

รูปแบบ : <body background = "name"> </body>

โดยที่ name เป็นชื่อของแฟ้มรูปภาพที่นำมาทำเป็น black ground

การขึ้นบรรทัดใหม่โดยไม่เว้นบรรทัด

รูปแบบ :

หน้าที่ : ใช้ในการขึ้นบรรทัดใหม่โดยที่ไม่ต้องการเว้นบรรทัด

ตัวอย่าง

Welcome to HTML Document.

This is the first paragraph.

This is the second paragraph.

ผลลัพธ์

Welcome to HTML Document. This is the first paragraph.

This is the second paragraph.

Image Map.

ตามปกติสามารถทำให้รูปภาพเชื่อมไปยังเอกสารอื่น ๆ ที่ต้องการได้ โดยใช้ Tag <IMG> ใน Tag <a> ซึ่งสามารถที่จะคลิกเมาส์ในทุก ๆ จุดของรูปภาพ แต่ถ้าต้องการให้รูปภาพเชื่อมต่อไปยังเอกสารได้หลาย ๆ เอกสาร โดยขึ้นอยู่กับตำแหน่งที่คลิกนั้น เราจะต้องใช้วิธี Image Map ซึ่งการใช้ Image Map นั้น ก็จะต้องสร้างแผนที่ขึ้นมา แล้วบอกพิกัดของพื้นที่แต่ละส่วนที่จะใช้ในการคลิกเมาส์

หมายเหตุ

ในโปรแกรม Browser ที่เป็น Text อย่างเดียวเช่น Lynx ซึ่ง browser เหล่านี้ไม่สามารถแสดงในแบบ Graphic ได้ ดังนั้นจึงไม่สามารถใช้ Image Map ใน browser เหล่านี้ได้

แต่ก็จะสามารถแก้ไขได้โดย ทำ Text Anchor ที่สอดคล้องกับ Image Map ขึ้นเพื่อให้ browser เหล่านี้สามารถเชื่อมโยงไปยังเอกสารตามที่ต้องการได้

การสร้าง Image Map มี 3 ขั้นตอน คือ

1. สร้าง หรือ เลือกรูปที่จะนำมาทำ Image Maps
2. สร้าง Map file ซึ่งเป็น Text file ที่แสดงพิกัด (Coordinate) ของรูปภาพที่จะนำมาทำ Image Maps พร้อมกับ URL ที่ต้องการจะเชื่อมโยงไปถึง
3. เชื่อมต่อ Image และ Map file เข้าด้วยกันโดย program ที่เรียกว่า โปรแกรม gateway soript ใน HTML

6.4 แนะนำภาษา JAVA

Sun Microsystem ได้แนะนำ Java ซึ่งมีลักษณะเป็นภาษา Object Oriented ซึ่งในปัจจุบันเป็นที่นิยมอย่างมาก Java ได้ถือกำเนิดขึ้นมาเพื่อเป็นภาษาสำหรับโปรแกรมที่ใช้งานได้จริงทางธุรกิจ

ประเภทของการ program ใน Java Programming

มีอยู่ 2 ประเภท

1. Application
2. Applet

Java Application เป็น standalone ที่สามารถ execute ได้ด้วยตัวเอง

Java Applet มีลักษณะคล้าย Java Application เพียงแต่ว่าไม่สามารถ run standalone ได้ ต้อง run ภายใต้ Virtual machine ของ Java โดยส่วนใหญ่โปรแกรม Java มักถูกเรียกใช้อยู่ใน homepage

6.4.1 ขั้นตอนพัฒนาโปรแกรมด้วย Java

หลังจากที่เขียน Program ด้วย Java เสร็จสิ้นแล้ว ทำการ compile ภายหลัง compile เสร็จจะใช้ชื่อ file class โดยจะอยู่ใน directory เดียวกับ Java source file (นามสกุลของ java source file คือ java) ใน class file นี้ เป็น java byte code ไปใช้ได้โดยแล้วแต่ละประเภท

6.4.2 ลักษณะเด่นของ Java

มีขนาดเล็กปลอดภัย Object oriented programming ทำให้สามารถแบ่งการโปรแกรม

ย่อย ๆ ได้สะดวกเป็นลักษณะ byte coded มีความสามารถ Multithreaded programming language ทำงานได้ไม่ขึ้นกับ platform

6.4.3 มุมมอง Java กับการโปรแกรม

Syntax ของ Java ถูกพัฒนามาจาก C++ ทำให้ programmer ที่คุ้นเคยกับภาษานี้ (ซึ่งมีอยู่เป็นจำนวนมาก) สามารถเรียนรู้ Java ได้อย่างง่ายดาย

หากแต่ว่าบางส่วนของ C++ ก็ไม่ปรากฏใน Java เนื่องจากต้องการแก้ปัญหาในส่วนที่ยุ่งยาก และซับซ้อนของ C++ เช่น ในส่วนของ pointer และการจัดการ memory โดยส่วนเหล่านี้จะมีความสลับซับซ้อนในการใช้ และอาจใช้ผิดได้โดยง่าย การหาความผิดพลาดของ pointer ใน program ขนาดใหญ่ เป็นการหาที่ ยุ่งยากมาก ดังนั้นในการจัดการ memory Java จะจัดการให้อย่างอัตโนมัติ programmer ไม่ต้องมานั่งเขียน garbage collection เอาเอง

การออกแบบในส่วนของ data types และ object ใน Java นั้นง่ายมาก โดยจะเข้มงวดมากในเรื่องของตัวแปร นั่นคือ สามารถใช้ตัวแปรใน data type ที่กำหนดไว้แล้วเท่านั้น ซึ่งหากมีการใช้ตัวแปรผิดพลาดในตอน compile program ไม่มีการปล่อยให้ข้อผิดพลาดนั้นหลุดรอดไปถึงขั้น run program ซึ่งทำให้เป็นการยากยิ่งขึ้นที่จะหา

programmer ที่มีประสบการณ์ในการเขียน C++ แล้วเปลี่ยนมาเขียน Java จะมีปัญหาเกี่ยวกับการเปลี่ยนการลดของ C++ แต่ก็จะได้รับความสะดวกในการเขียนที่ง่ายขึ้น

6.4.4 ความเป็น Object oriented programming ของ Java

ภาษา Java นับเป็นภาษาแบบ Object oriented programming (OOP) ซึ่งลักษณะของภาษาเช่นนี้มีประโยชน์มากในการพัฒนา software OOP จะ organize program เป็น set ของ component ที่เรียกว่า objects โดย objectsเหล่านี้จะมีลักษณะเป็นอิสระต่อกัน และจะมีกฎในการติดต่อสื่อสารต่อกัน

Java จะได้รับ concept object oriented มาจาก C++ และภาษาอื่นอีกเช่น Smalltalk ในข้อดีของภาษาในแนว object oriented ก็คือว่า จะสามารถเข้าใจง่าย หาข้อผิดพลาดได้ง่าย และนำไปปรับปรุงใช้ใน project ได้ง่าย

6.4.5 ความปลอดภัยใน Java

ในการประสบความสำเร็จของ Java อีกประการหนึ่งก็คือ เป็นภาษาซึ่งมีความปลอดภัย ซึ่งการเน้นทางด้านความปลอดภัยเป็นสิ่งที่เหมาะสมสำหรับการพัฒนา Java ในโลกของ World Wide Web

Java จะมีความปลอดภัยในหลายระดับ อย่างแรกก็คือว่าเป็นภาษาที่ถูกออกแบบมาให้ยากที่จะ execute จาก code ที่ทำอันตราย program ได้ เช่น การไม่มี pointer ถึงแม้ว่า pointer จะมีประโยชน์อย่างมาก แต่ก็มีโทษมหันต์ เนื่องจาก pointer สามารถถูกใช้เพื่อเข้าถึงในส่วนพื้นที่ที่อาจทำลาย program ได้ และยังสามารถเข้าถึงพื้นที่ใน memory ได้ ดังนั้นการไม่มี pointer จึงเป็นความปลอดภัยอย่างหนึ่งของ Java ที่มีมากกว่าภาษาอื่น

ความปลอดภัยในระดับ byte code โปรแกรม Java ที่ทำการ compile แล้วจะอยู่ในรูปแบบของ byte codeก่อนที่จะมีการ run program Java จะมีการตรวจสอบในแต่ละ byte code ก่อนว่ามีสิ่งผิดปกติเกิดขึ้นหรือไม่

ความปลอดภัยในการใช้ Java applet ก็คือ Java applet จะไม่สามารถ เปิด อ่าน เขียน file บนระบบของuser เพื่อป้องกัน program จากการกระทำผิดของ disk dive ของ user

6.4.6 Multithread ของ Java

ในลักษณะของ Multithread ทำให้สามารถทำงานได้หลาย ๆ งานในเวลาเดียวกัน ซึ่งเหมาะกับ operating system ที่มีลักษณะเป็นแบบ multitasking เช่น Window 95 โดย Java ได้จัดเตรียม tool ในการเขียน program แบบ multithread และทำให้ program เหล่านี้มีการ execute ได้อย่างน่าเชื่อถือ

6.4.7 การทำงานของ Java ที่ไม่ขึ้นกับ platform

Software ของ computer ส่วนมากจะถูกพัฒนาเฉพาะสำหรับ operating system บางประเภทเท่านั้น การที่ไม่ขึ้นกับ platform ทำให้ program เดียวกันทำงานได้บน operating system ที่แตกต่างกัน ซึ่งช่วยลดความยุ่งยากในการพัฒนา program หลายรอบสำหรับแต่ละ operating system

ชนิดตัวแปรของ Java มีขนาดเดียวกันในทุก ๆ platform รวมถึง applet ที่พบบน Web byte code สามารถexecute ได้ในทุก ๆ platform โดยไม่มีการเปลี่ยนขนาด

6.5 จาวาสคริปต์เบื้องต้น

วิธีการเพิ่มโค้ดจาวาสคริปต์ในเว็บเพจ

จะเริ่มกันที่วิธีการแทรกโค้ดจาวาสคริปต์ลงไปในไฟล์ HTML ซึ่งแท็ก (tag) ที่ใช้บ่งบอกถึงจุดเริ่มต้นและจุดสิ้นสุดของโค้ดจาวาสคริปต์ก็คือ <SCRIPT> และ </SCRIPT> ซึ่งแท็กเริ่มต้นจาวาสคริปต์ ควรระบุ ว่าเป็นภาษาจาวาสคริปต์ด้วย ซึ่งจะเขียนได้ดังนี้

```
<SCRIPT language="JavaScript">
```

ส่วนที่เขียนว่า language="JavaScript" ก็เพื่อให้โปรแกรมบราวเซอร์รับรู้ว่าเป็นภาษาจาวาสคริปต์ ไม่ใช่ภาษาสคริปท์อื่นๆเช่น VBScriptจะใส่โค้ดจาวาสคริปท์ตามแท็กดังกล่าวและสิ้นสุดโค้ดจาวาสคริปท์ด้วยแท็ก </SCRIPT> ซึ่งจะเขียนได้ดังนี้

```
<SCRIPT language="JavaScript">

.....โค้ดจาวาสคริปท์.....

</SCRIPT>
```

ในไฟล์ HTML ไฟล์หนึ่งสามารถจะมีแท็ก <SCRIPT> ได้หลายชุดตามที่ต้องการ ซึ่งก็เหมือนกับว่า มันเป็นHTML แท็กอย่างหนึ่งนั้นเอง แต่อย่าลืมแท็กสำหรับปิด </SCRIPT> และถ้าหากต้องการใช้ ฟังก์ชัน (ซึ่งจะอธิบายในบทต่อ ๆ ไป) จะต้องใส่ฟังก์ชันจาวาสคริปท์ไว้ในส่วน <HEAD> และ </HEAD> ของไฟล์ HTML ทั้งนี้ก็เพื่อให้ฟังก์ชันถูกเรียกขึ้นมาก่อนที่เว็บเพจนั้นจะปรากฏให้เห็นและจะได้ไม่ต้องพบกับปัญหาการเกิดความผิดพลาด (error) ตัวอย่างต่อไปนี้แสดงการเขียนฟังก์ชันจาวาสคริปท์

```
<HEAD>
<TITLE>My World</TITLE>
```

```
<SCRIPT language="JavaScript">
  function cool0 {
  .....ตัวโปรแกรม.....
}
</SCRIPT>
</HEAD>
```

ตอนนี้ ยังมีอีกสิ่งหนึ่งที่ควรจะทราบก่อนที่จะเริ่มต้นการเขียนจาวาสคริปท์กันก็คือ เนื่องจากในโลกนี้ ยังมีผู้ใช้อินเทอร์เนตอีกเป็นจำนวนมากที่ยังใช้โปรแกรมบราวเซอร์รุ่นเก่า ซึ่งยังไม่มีความสามารถในการแสดงจาวาสคริปท์ นั่นคือ ไม่รู้จักแท็ก <SCRIPT> ซึ่งมีผลทำให้โปรแกรมเหล่านั้นแสดงจาวาสคริปท์ออกมาเป็นตัวอักษรธรรมดา วิธีแก้ไขก็คือต้องทำให้โปรแกรมบราวเซอร์รุ่นเก่า ๆ ไม่สนใจ

สิ่งที่อยู่ในแท็ก <SCRIPT> โดยจะต้องเขียนเป็นคอมเม้นท์ (comment) เหมือนกับการเขียนคอมเม้นท์ในไฟล์ HTML ซึ่งทำได้ดังนี้

```
<SCRIPT language="JavaScript">
<!--hide from old browsers

.....โค้ดจาวาสคริปท์.....

//-->
</SCRIPT>
```

6.5.1 สร้างอเลิท (Alerts) ด้วยจาวาสคริปท์

เพิ่มลูกเล่นจาวาสคริปท์อเลิทเพื่อสร้างความแปลกใจ จาวาสคริปท์อเลิท (JavaScript alert) ก็คือ ป๊อปอัพเล็ก ๆ ที่สามารถแสดงข้อความให้ผู้ใช้อ่าน สามารถสร้างอเลิทไว้ในส่วนหนึ่งของเว็บเพจก็ได้เพื่อบอกหรือทักทายผู้ใช้ เริ่มโดยใช้สคริปท์ต่อไปนี้


```
alert('ข้อความที่ต้องการแสดง')
```

อเลิทต้องใช้ร่วมกับคำสั่งอื่น นั่นคือ เมื่อผู้ใช้ดำเนินการบางอย่างแล้วจะทำให้เกิดอเลิทขึ้น เช่น คลิกบัตตอน หรือเลื่อนเม้าส์ผ่านมาบนไฮเปอร์ลิ้งค์ ในตัวอย่างนี้จะใช้กับ onMouseover ขึ้นเมื่อผู้ใช้จะพยายามคลิกบนไฮเปอร์ลิ้งโดยการเลื่อนเม้าส์มาอยู่ที่ไฮเปอร์ลิ้งค์

```
<A HREF="noplace" onMouseOver="alert('Hey! I said not to try clicking this link!')">
Don't click this link!
</A>
```

onMouseOver=" " ใช้เพื่อบอกโปรแกรมบราวเซอร์ให้ให้ทำตามคำสั่งอเลิทที่กำหนดไว้เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลื่อนเม้าส์มาบนไฮเปอร์ลิ้งค์

alert('Hey! I said not to try clicking this link!') เป็นการบอกให้โปรแกรมบราวเซอร์แสดงข้อความที่อยู่ในวงเล็บ (ต้องมีเครื่องหมายคำพูดเดี่ยวด้วย) บนป๊อปอัพ

อเลิทยังสามารถทำได้มากกว่านั้น เช่น ทักทายผู้ใช้เมื่อจะเริ่มทำการโหลดเว็บเพจ หรือใช้แจ้งสิ่งต่าง ๆ

```
<HEAD>
<TITLE>Cool JavaScript</TITLE>
<SCRIPT language="JavaScript">
<!-- hide from old browsers
    alert('Welcome to my Web Site!');
//-->
</SCRIPT>
</HEAD>
```

โค้ดข้างบนจะแสดงอเลิทให้ผู้ใช้คลิก OK เพื่อโหลดข้อมูลในเว็บเพจนั้นต่อไป ซึ่งสามารถอธิบายได้ดังนี้

<SCRIPT language="JavaScript"> บอกให้โปรแกรมบราวเซอร์รับรู้ว่าสคริปท์ที่ใช้คือจาวาสริปท์

<!--hide script from old browsers ทำให้โปรแกรมรุ่นเก่าไม่สนใจโค้ดจาวาสคริปท์

alert('Welcome to my Web Site!'); ข้อความที่จะแสดงในอเลิท ซึ่งต้องอยู่ในระหว่างเครื่องหมายคำพูดเดี่ยว และปิดท้ายบรรทัดด้วยเซมิโคลอน (semi colon)

//--> บอกจุดสิ้นสุดการซ่อนโค้ดจาวาสคริปท์จากบราวเซอร์รุ่นเก่า

</SCRIPT> บอกจุดสิ้นสุดของโค้ดจาวาสคริปท์

ลองคลิกไฮเปอร์ลิ้งค์ข้างล่างเพื่อไปยังเพจตัวอย่าง

Example 1

ถ้าต้องการสร้างให้มีการอเลิทหลาย ๆ ครั้ง ก็ให้คั่นด้วยเครื่องหมายเซมิโคลอน

```
<HEAD>
<TITLE>JavaScript Example 2</TITLE>
<SCRIPT language="JavaScript">
<!--hide from old browsers
    alert('Please Sign My Guestbook,NOW!');
    alert('I mean it, NOW!!!');
    alert('Did I mention I had a guestbook? Well SIGN IT!');
    alert('Oh,remember....THE GUESTBOOK! O.K.?!!');
//-->
```

```
</SCRIPT>
</HEAD>
```

อเลิทยังสามารถใช้ร่วมกับบัตตอน นั่นคือ คลิกบัตตอนแล้วมีอเลิทปรากฏขึ้น เพื่อความแปลกใจ การใช้อเลิทกับบัตตอนไม่ต้องใช้แท็ก<SCRIPT> ให้สร้างบัตตอนและกำหนดให้เกิดอเลิทเมื่อบัตตอน ถูกคลิกเท่านั้น ดังตัวอย่างนี้

```
<FORM>
<INPUT type="button" value="Click here to see what I think of YOU!"
onClick="alert('You are the greatest person I have ever met!')">
</FORM>
```

ถ้าเกิดการผิดพลาดก็ให้กลับมาตรวจสอบให้ดีว่าพิมพ์ทุกอย่างถูกต้องหรือไม่

6.5.2 การสร้างตัวแปร (Variable)

การสร้างตัวแปรและชนิดต่าง ๆ ของตัวแปรเพื่อจะได้นำไปใช้กับฟังก์ชันจาวาสคริปท์ เริ่มจากการประกาศ (declare) ตัวแปร ซึ่งจะต้องอยู่ในส่วน HEAD ของไฟล์ HTML โดยการประกาศตัวแปรลงไประหว่างแท็ก SCRIPT ดังนี้

```
<HEAD>
<SCRIPT language="JavaScript">
<--hide from old browsers
   var name=value;
//-->
</SCRIPT>
</HEAD>
```

โดยที่สามารถอธิบายความหมายได้ดังนี้

var คือการบ่งชี้ว่าเป็นการประกาศตัวแปร

name ชื่อของตัวแปร จะกำหนดให้ตัวแปรที่ชื่อว่าอะไรก็ได้ ที่ไม่ซ้ำกับคำสงวนของการเขียนโปรแกรม เช่นให้ตัวแปรมีชื่อว่า function ไม่ได้

value ค่าเริ่มต้นของตัวแปรนั้น ซึ่งสามารถจะเป็นตัวเลข (number) ข้อความ (words) ค่าบูลีน
(boolean(true , false)) หรือว่าง (null) ก็ได้ ซึ่งจะแยกอธิบายในส่วนต่อไปนี้

การใช้ตัวเลข (Numbers)

สามารถกำหนดค่าที่เป็นตัวเลขให้กับตัวแปรโดยใช้เครื่องหมายเท่ากับ ดังนี้

```
var cars=3;
```

ตัวเลขดังกล่าวไม่เป็นจำนวนเต็ม เช่น เป็นทศนิยมก็ได้ เช่น

```
var cost=9.95;
```

การใช้สตริง (Strings)

สตริงคือกลุ่มของตัวอักษร เช่น คำหรือประโยคการกำหนดค่าตัวแปรเป็นสตริงต้องใช้เครื่องหมายคำพูดคู่ล้อมสตริงนั้นไว้ ดังนี้

```
var movie="The Lost World";
```

ถ้าใส่ตัวเลขไว้ระหว่างเครื่องหมายคำพูด ตัวเลขนั้นจะถูกใช้เหมือนกับว่ามันเป็นสตริง

การใช้ค่าบูลีน (Boolean Value)

เป็นการกำหนดให้ตัวแปรมีค่าเป็นจริง (true) หรือ เท็จ (false) ดังนี้

```
var story=true;
```

ค่าว่าง (Null Value)

ถ้ากำหนดให้ตัวแปรมีค่าเป็น null นั่นคือ เป็นค่าว่าง ไม่ใช้แม้แต่ 0 ซึ่งมีรูปแบบการใช้ดังนี้

```
var mymoney=null;
```

ความแตกต่างของตัวอักษรตัวเล็กหรือตัวใหญ่(Case Sensitivity)

ไม่เหมือนกับภาษา HTML จาวาสคริปท์ให้ความสำคัญกับความแตกต่างของตัวอักษรตัวเล็กหรือตัวใหญ่ จะใช้สลับกันไม่ได้ เช่น joHn กับ JOHN จะไม่ได้หมายถึงสิ่งเดียวกัน ดังนั้นต้องใช้ความรอบคอบ หากพิมพ์ผิด อาจทำให้โปรแกรมทำงานผิดพลาดได้

6.5.3 การสร้างฟังก์ชัน

ฟังก์ชันคือชุดของจาวาสคริปท์สเตทเม้นท์เพื่อใช้ทำงานอย่างใดอย่างหนึ่งฟังก์ชันต้องสร้างขึ้นภายในแท็ก SCRIPT ซึ่งอยู่ในส่วน HAED ของไฟล์ HTML การประกาศฟังก์ชันทำได้ดังนี้

```
<HEAD>
<SCRIPT langusge="JavaScript">
<!--hide from old browsers
    function name (parameter1,parameter2) {
.....จาวาสคริปท์สเตทเม้นท์ต่าง ๆ .....
}
```

```
//-->
</SCRIPT>
</HEAD>
```

โค้ดข้างบนสามารถอธิบายได้ดังนี้

function เป็นส่วนบ่งชี้ว่ากำลังประกาศฟังก์ชัน

name ชื่อของฟังก์ชัน จะใช้ชื่ออะไรก็ได้ที่ไม่ใช่คำสงวนของการเขียนโปรแกรม

(parameter1,parameter2) พารามิเตอร์ (parameters) ก็คือตัวแปรที่ส่งไปยังฟังก์ชันเมื่อฟังก์ชันถูกเรียก อาจจะใช้พารามิเตอร์กี่ตัวหรือไม่มีพารามิเตอร์เลยก็ได้

{ เครื่องหมายปีกกาเปิดแสดงให้เห็นการเริ่มต้นฟังก์ชัน หรือสเตทเม้นท์ต่าง ๆ

} เครื่องหมายปีกกาปิดแสดงให้เห็นจุดสิ้นสุดของฟังก์ชัน หรือสเตทเม้นท์

การที่จะสามารถใช้ฟังก์ชันได้ จะต้องทำการเรียกฟังก์ชันนั้นๆ เสียก่อน การเรียกฟังก์ชันสามารถทำได้ดังนี้

```
function name (parameter1,parameter2);
```

นั่นคือ ต้องระบุชื่อของฟังก์ชันที่ต้องการใช้ พร้อมทั้งพารามิเตอร์ (ถ้ามี) คราวนี้จะลองเขียนข้อความลงบนสเตตัสบาร์โดยใช้ฟังก์ชัน และตัวแปร ฟังก์ชันดังกล่าวที่กำลังสร้างขึ้นจะทำการเขียนข้อความลงบน สเตตัสบาร์เมื่อเลื่อนเม้าส์มาบนไฮเปอร์ลิ้งและลบข้อความออกเมื่อเลื่อนเม้าส์ไปยังตำแหน่งอื่น ๆ

Place your mouse here!

สคริปท์จะเป็นดังนี้

```
<HEAD>
<SCRIPT language="JavaScript">
<!-----hide
   var text=" ";
    function overlink (ten) {
    window .status=text;
    }
    function offlink (text) {
    window .status=text;
    }
//-->
</SCRIPT>
</HEAD>
<BODY>
<A HREF="jvar.htm" onMouseover="('Function Rule!');return true"
onMouseout="offlink(' ');return true"> Place your mouse here!</A>
</BODY>
```

คำอธิบายสคริปท์ข้างบนเป็นดังนี้

ในส่วน HEAD

var text=" "; เป็นการประกาศตัวแปรชื่อ text และกำหนดค่าเริ่มต้นให้กับมันเป็นคำว่างเปล่า

function overlink(text) เป็นการประกาศฟังก์ชันชื่อ overlink ฟังก์ชันนี้ใช้ตัวแปร text ในการทำงาน

{ จุดเริ่มจาวาสคริปท์สเตทเม้นท์ของฟังก์ชัน overlink

window .status=text; เป็นการกำหนดให้สเตตัสบาร์ถูกเขียนด้วยข้อความที่อยู่ในตัวแปร text โดยค่าของตัวแปร text จะถูกส่งไปยังฟังก์ชัน เมื่อฟังก์ชันถูกเรียก (ขณะที่เม้าส์เลื่อนมาอยู่เหนือไฮเปอร์ลิ้งค์) ซึ่งในตอนนี้ ตัวแปร text จะมีค่าเป็นคำว่า "Functions Rule!"

} จุดสิ้นสุดของฟังก์ชัน overlink.

function offlink (text) เป็นการประกาศฟังก์ชันชื่อ offlink ซึ่งฟังก์ชันนี้ใช้ตัวแปร text ในการทำงานเช่นกัน

window .status=text; เป็นการกำหนดให้สเตตัสบาร์ถูกเขียนข้อความที่อยู่ในตัวแปร text โดยค่าของตัวแปร text จะถูกส่งไปยังฟังก์ชันเมื่อฟังก์ชันนั้นถูกเรียกในขณะที่เม้าส์เคลื่อนออกจากไฮเปอร์ลิงค์ นั่นคือ ค่าของตัวแปร text ที่ถูกส่งไปแสดงบนสเตตัสบาร์ในตอนนี้ก็คือคำว่างเปล่า " "

ในส่วน BODY

<A HREF="jvar.htm"onMouseover="overlink('Function Rule!');return true"
onMouseOut="offlink(' ');return true"> Place your mouse here!</A>

แท็กดังกล่าวเรียกฟังก์ชันทั้งสอง และผ่านค่าสตริง ซึ่งกำหนดให้เป็นค่าของตัวแปร text ฟังก์ชันแรกคือ overlink จะถูกเรียกโดยเมธอด onMouseOver หมายความว่า เมื่อเลื่อนเม้าส์มาอยู่บนไฮเปอร์ลิ้งค์ จะทำให้สเตทเมนที่อยู่ในฟังก์ชันนี้ถูกดำเนินการ จะเห็นได้ว่าฟังก์ชัน overlink ถูกเรียกขึ้นมาโดยมีค่าตัวแปร สตริงซึ่งอยู่ในวงเล็บ ( ) สังเกตว่าจะใช้เครื่องหมายคำพูดเดี่ยวในการกำหนดค่าสตริง เพื่อไม่ให้ซ้ำซ้อนกับเครื่องหมายคำพูดคู่ซึ่งใช้กับ onMouseOver ค่าสตริงนี้คือค่าส่งไปยังฟังก์ชัน overlink ดังนั้น สเตตัสบาร์จะถูกเขียนโดยข้อความ "Functions Rule!" ในขณะที่เลื่อนเม้าส์ไปอยู่บนไฮเปอร์ลิ้งค์นั่นเอง หลักการทำงาน

ของฟังก์ชันofflink ก็จะเหมือนกับฟังก์ชัน overlink แต่จะเป็นในทางตรงกันข้าม คือฟังก์ชัน offlink จะถูกเรียกทันทีที่เลื่อนเม้าส์ออกจากไฮเปอร์ลิงค์ อย่าลืมใส่คำว่า returning true เพื่อให้สคริปท์ ทำงานอย่างถูกต้อง

6.5.4 Forward และ Back โดยใช้บัตตอน

การสร้างบัตตอนที่ใช้ในการเดินหน้าและถอยหลังระหว่างเว็บเพจที่ผู้ใช้เพิ่งเข้าไปหรือกลับ ขึ้นหน้าไปยังเว็บเพจล่าสุดโดยใช้บัตตอน โดยบัตตอนชนิดนี้จะทำงานคล้าย ๆ กับปุ่มBack และ Forward ซึ่งสามารถสร้างได้จากสคริปท์ต่อไปนี้

```
<FORM>
<INPUT type="button" value="Click here to go back" onClick=history.back()">
</FORM>
```

โค้ดข้างบนสร้างบัตตอนซึ่งเมื่อคลิกแล้วจะนำกลับไปยังที่อยู่ในรายการฮิสทรี่ ลองทดสอบการสร้างโดยคลิกไฮเปอร์ลิ้งค์ข้างนี้
Example Page

สามารถอธิบายสคริปท์ได้ ดังต่อไปนี้

<FORM> แท็กฟอร์มใช้เพื่อให้สามารถสร้างบัตตอน

<INPUT type="button" value="Click here to go back"..... สร้างบัตตอนเพื่อนำมาใช้กับสคริปท์

....onClick ="history.back()"> เมธอด onClick สั่งให้โปรแกรมบราวเซอร์ให้ดำเนินการตามคำสั่งที่อยู่ภายในเครื่องหมายคำพูด นั่นคือในที่นี้เป็น history.back() ซึ่งหมายความว่าให้ถอยหลังกลับไปยังเพจที่ผ่านมาในฮิสทรี่ลิสท์

ในทางตรงกันข้าม ถ้าจะไปข้างหน้า 1 เพจ ในฮิสทรี่ลิสท์ ก็ทำเช่นเดียวกัน แต่เปลี่ยน

history.back() เป็น history.forword() เท่านั้น

ส่วน history.go(-1) or history.go(1) ทำให้สามารถเดินหน้าหรือถอยหลังไปในฮิสทรี่ลิสท์ได้เป็นจำนวนกี่เพจตามที่ต้องการ เช่น กลับไป 2 หน้า หรือไปข้างหน้า 5 หน้า ฯลฯ

6.6 JDBC

JDBC (Java Database Connectivity) คือ Java API ที่ใช้สำหรับตีความคำสั่ง SQL โดยประกอบด้วยชุดของ classes และ interfaces ที่เขียนด้วยโปรแกรมภาษา Java ตัว JDBC นี้เป็น API มาตรฐาน สำหรับการพัฒนาระบบฐานข้อมูล และทำให้มีความเป็นได้ในการเขียนโปรแกรมเกี่ยวกับฐานข้อมูลด้วย Java API เพียงอย่างเดียว

การใช้ตัว JDBC นี้ทำให้ง่ายในการส่งคำสั่ง SQL ให้กับระบบฐานข้อมูลชนิดต่าง ๆ ในรูปแบบเดียวกันนั่นคือ JDBC API จะไม่บังคับให้เขียนโปรแกรมตัวหนึ่งเพื่อใช้กับฐานข้อมูลของ Sybase และอีกโปรแกรมหนึ่ง สำหรับฐานข้อมูลของ Oracle และอีกหนึ่งโปรแกรมสำหรับ ฐานข้อมูล Informix และอีกหลายโปรแกรมสำหรับฐานข้อมูลชนิดต่าง ๆ สามารถเขียนโปรแกรมโดยใช้ JDBC API เพียงโปรแกรมเดียว และโปรแกรมนั้นจะสามารถส่งคำสั่ง SQL ไปยังฐานข้อมูลชนิดต่าง ๆ นั้นได้ และการเขียนโปรแกรมโดยใช้ภาษา Java นี้ไม่ต้องกังวลว่าจะต้องเขียนโปรแกรมอีกหลายตัว เพื่อนำไปใช้กับคอมพิวเตอร์ที่อยู่บน platform ชนิดต่าง ๆ เลย ในการรวมตัวกันระหว่าง Java และ JDBC นี้ ทำให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมเพียงหนึ่งโปรแกรม แล้วสามารถนำไปใช้ได้ทุก ๆ แห่งที่ต้องการ Java ที่กำลังแข็งขัน มีความปลอดภัย ง่ายต่อการใช้ ง่ายต่อการเข้าใจ และถ่ายโอนข้อมูลมาได้อย่างอัตโนมัติบนระบบเครือข่าย เป็นโปรแกรมภาษาที่ฉลาดมากที่จะ ใช้เขียนโปรแกรมที่ใช้จัดการฐานข้อมูล และตัว JDBC ก็เป็นทางเลือกที่ต้องการเพื่อจะให้โปรแกรมตัวนั้นสามารถติดต่อกับฐานข้อมูลไม่ว่าฐานข้อมูลชนิดนั้นจะเป็นชนิดใดก็ตาม

JDBC เป็นส่วนขยายของความสามารถที่ Java ทำได้ นั่นคือ เมื่อใช้ Java และ JDBC API สามารถเผยแพร่ web page ที่มี java applet ที่ใช้ข้อมูลจากฐานข้อมูลที่อยู่แห่งอื่นได้ หรือในบริษัทต่าง ๆ ก็สามารถใช้JDBC เพื่อติดต่อพนักงานเข้ากับฐานข้อมูลต่าง ๆ ไม่ว่าพวกเขาจะใช้เครื่อง Windows, Macintosh หรือ UNIX ก็ตามบนระบบอินทราเน็ต และในขณะนี้ความต้องการของการเข้าถึงฐานข้อมูลที่ง่ายจาก Java กำลังมีเพิ่มสูงขึ้นเรื่อย ๆ กับผู้เขียนโปรแกรม Java จำนวนมาก

ผู้ที่จัดการ MIS ชอบที่จะรวม Java และ JDBC เข้าด้วยกัน เพราะมันทำให้การเผยแพร่ข้อมูลทำได้ง่ายและประหยัด ในบริษัทสามารถใช้ฐานข้อมูลที่ได้ติดตั้งไว้ก่อนแล้ว และการเข้าถึงข้อมูลก็ทำได้ง่าย แม้ว่ามันจะอยู่ในระบบจัดการฐานข้อมูลต่างชนิดกันก็ตาม ในการพิมพ์โปรแกรมตัวใหม่จะใช้เวลาน้อยลง การติดตั้งและการดูแลเรื่อง version ก็ง่ายมาก โดยผู้เขียนโปรแกรมสามารถเขียนหรือแก้ไขโปแกรม เพียงโปรแกรมเดียวแล้วนำไปไว้ที่เครื่อง server จากนั้นทุก ๆ คน ก็จะสามารถเข้าถึงข้อมูลแบบ version ล่าสุดได้ทันที และสำหรับธุรกิจบริการข้อมูลการขาย Java และ JDBC เป็นหนทางที่ดีที่สุดที่จะนำเสนอข้อมูลที่แก้ไขล่าสุดให้กับลูกค้าข้างนอกของตน

6.6.1 JDBC ทำอะไรได้บ้าง

ในการใช้ทั่วไป JDBC สามารถทำได้ 3 อย่าง ดังนี้

สร้างการเชื่อมต่อกับฐานข้อมูล

ส่งคำสั่ง SQL

จัดการกับผลลัพธ์ที่ได้มา

ตัวอย่างข้างล่างนี้เป็นโปรแกรมที่ตัดมา เพื่อแสดงการทำงานทั้ง 3 อย่างข้างต้น ดังนี้

```
Connection con = DriverManager.getConnection(
        "jdbc:odbc:wombat", "login","password");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executQuery("SELECT a,b,c FROM Table1");
while (rs.next()) {
  int x = getInt("a");
  String s = getString("b");
  float f = getFloat("c");
}
```

6.6.2 การรักษาความปลอดภัยของ JDBC

ผู้พัฒนา JDBC นั้นควรจะพิจารณา

ความน่าเชื่อถือของโปรแกรม

ความน่าเชื่อถือของ Java program เป็นไปตาม Java application และ Java applet จากแหล่งข้อมูลที่เกี่ยวข้อง แหล่งข้อมูลที่เกี่ยวข้องนั้นจะเป็น server ที่เก็บรายละเอียดของการให้บริการของบริษัทหรือ Server หลัก Applet สามารถที่จะถูกกำหนดด้วยรหัสลับที่แน่ใจได้ว่าสามารถจะ download ได้อย่างสมบูรณ์

ความไม่น่าเชื่อถือของโปรแกรม

ความไม่น่าเชื่อถือของโปรแกรมนั้นคือ Java Applet ที่เข้าได้ทาง internet ความไม่น่าเชื่อถือของ Applet คือ จะต้องไม่ยอมให้ Client เข้าถึงเช่น file system ได้เป็นต้น ในอนาคตความไม่น่าเชื่อถือของ Appletเป็นเพียงแค่การยอมรับการเชื่อมโยงได้กับ network กลับไปยัง Server จากการ download

การใช้งาน JDBC การตั้งชื่อฐานข้อมูลบน network

Javasoft นั้นออกแบบการใช้งานมาตรฐานการตั้งชื่อไว้เป็นแบบ URL (Uniform Resource Locator) syntaxพื้นฐานที่จะอธิบายถึง ฐานข้อมูลก็คือ jdbc:: ตัวอย่างฐานข้อมูล mydatabase บน host cc2.cpe.ku.ac.th ,port 1234, โดยใช้ mSQL สำหรับการเข้าถึงทาง Network

jdbc:mSQL://cc2.cpe.ku.ac.th:1234/mydatabase

การประมวลผลการค้นหาข้อมูลและการได้มาซึ่งผลลัพธ์ เมื่อได้ผลของการค้นหาข้อมูล โดย SQL จากการทำงานของ java.sql.Statement class แล้วหลังจากการสถาปนาการเชื่อมโยงฐานข้อมูลจึงสร้าง statementและประมวลผล

```
java.sql.Statement  stmt = conn.createStatement();
ResultSet r = stmt.executeQuery("SELECT a,b,c, FROM Table");
```

ResultSet object r จะได้รับข้อมูลจากการค้นหาไว้ การกำหนดค่าให้จาก field ใน r โดยใช้ JDBC getxxx


```
while (r,next)
```

```
{ int I = r getInt("a");
String s = r,getString("b");
byte b[ ] = r,getByte("c");
System.out.println("ROW = "+ I + " " + s + " " + b[0]);}
```

การใช้งานแอพพลิเคชั่นเมื่อทำงานในระบบ ไคลเอนท์เซอฟเวอร์ นั้นสามารถเลือกใช้ได้หลายรูปแบบ ขึ้นอยู่กับลักษณะเฉพาะของงานแต่ละชนิด และขนาดของงานที่เลือกใช้งานบางชนิดต้องติดต่อฐานข้อมูลขนาดใหญ่ แต่บางชนิดไม่จำเป็นต้องใช้ฐานข้อมูลขนาดใหญ่มากนัก องค์ประกอบเหล่านี้มีส่วนเกี่ยวข้องกับประสิทธิภาพของแอพพลิเคชั่น และผลลัพธ์ของงานที่ได้มา มีวิธีการแก้ไขปัญหาโดยแบ่ง แอพพลิเคชั่นออกเป็นโมดูลย่อยให้ทำงานร่วมกัน แต่ละโมดูลทำงานตามหน้าที่ของตนเป็นอิสระจากโมดูลอื่น ทำให้ง่ายต่อการเปลี่ยนแปลงแก้ไขและบำรุงรักษา เรียกการทำงานแบบนี้ว่า " มัลติเทียร์ "ซึ่งพัฒนาจาก " ทูเลียร์ "และใช้วิธีการเข้าถึงฐานข้อมูลโดยใช้ JDBC ไดรเวอร์ (Java Database Connectivity Driver) ซึ่งแบ่งลักษณะการทำงานออกเป็น 4 แบบ ขึ้นอยู่กับผู้ใช้จะเลือกใช้งานอย่างไร เพื่อประสิทธิภาพที่ดีที่สุด

6.6.3 การทำงานแบบ Client/Server

สถาปัตยกรรมของแอพพลิเคชั่นแบบ Client/Server จะแบ่งการประมวลผลออกเป็นสองโปรแกรม โดยทั่วไปจะทำงานบนเครื่องสองเครื่องขึ้นไป แอพพลิเคชั่นที่ทำงานกับฐานข้อมูลClient/Server จะรับผิดชอบการเก็บข้อมูล , การประมวลผลข้อมูล , การโอนย้ายข้อมูล และไปแสดงผลที่อื่น เครื่อง server จะเก็บรวบรวมข้อมูลไว้ ส่วนเครื่องไคลเอนต์จะประมวลผลข้อมูลที่ได้มา หรือสร้างเป็นข้อมูลใหม่ วิธีการทำงานโดยใช้สถาปัตยกรรมแบบ Client/Server นี้ทำให้สามารถติดต่อใช้งานข้อมูลได้จากผู้ใช้หลายแห่ง

<u>การทำงาน Client/Server แบบทูเทียร์ (Two - Tier Application)</u>

รูปแบบธรรมดาทั่วไปของสถาปัตยกรรม Client / Server เป็น ทูเทียร์ (two - tier) ซึ่งมาจากการแบ่งการทำงาน ของแอพพลิเคชั่นออกเป็นส่วน Client / Server ยอมรับการติดต่อจากหลาย ๆ ที่เข้าคู่ส่วนให้บริการซึ่งเก็บข้อมูลไว้ ส่วนแสดงผลจะอยู่ที่ Client และส่วนเก็บ

รวบรวมข้อมูล จะอยู่ที่ Server แอพพลิเคชั่นทั่วไป ส่วนใหญ่บนอินเตอร์เน็ต เช่น email , telnet , ftp , gopher หรือ web เป็น แอพพลิเคชั่นแบบ 2 ระดับซึ่งทำงานโดยไม่ต้องประมวลข้อมูลขนาดใหญ่ โดยทั่วไปจะทำงานติดต่อใช้ข้อมูลภายในอินเตอร์เน็ต

ข้อดี ของแอพพลิเคชั่นแบบ ทูเทียร์

คือเป็น แอพพลิเคชั่นง่าย ๆ ธรรมดา ที่ไม่ต้องการการดูแลบำรุงรักษามากนัก สามารถพิจารณาเลือกใช้ว่าเหมาะกับ แอพพลิเคชั่นแบบ ทูเทียร์ หรือไม่ควรขึ้นอยู่กับเงื่อนไขดังนี้

- เป็นแอพพลิเคชั่นที่ใช้ฐานข้อมูลเดี่ยว
-ฐานข้อมูลบรรจุอยู่ภายใน CPU เครื่องเดียว
-ฐานข้อมูลมีขนาดเดิมไม่เปลี่ยนแปลงบ่อย ๆ
-user base ไม่มีการเปลี่ยนแปลงบ่อย
-requirement ไม่มีการเปลี่ยนแปลงหรือมีการเปลี่ยนแปลงน้อยมาก
-แอพพลิเคชั่น ที่สมบูรณ์แล้ว ไม่จำเป็นต้องดูแลบำรุงรักษามาก

ข้อเสียของ ทูเทียร์

ความต้องการของผู้ใช้เพิ่มมากขึ้นดังนั้นความซับซ้อนของแอพพลิเคชั่นจึงต้องมากตามไปด้วย จากการที่ไคลเอนท์มีประสิทธิภาพ และมีความซับซ้อนขึ้นเรื่อย ๆ ในขณะที่เซิร์ฟเวอร์มีขนาดเล็กลงเพื่อให้ราคาถูกและความสามารถในการจัดการฐานข้อมูลที่ซับซ้อนต่ำลง เช่น ในปัจจุบันเครื่องคอมพิวเตอร์เมนเฟรมได้ถูกเปลี่ยนมาใช้เครื่องคอมพิวเตอร์ขนาดเล็กจำนวนมาก มาทำงานแทน และงานบางส่วนจะถูกผลักภาระไปที่เครื่องไคลเอนท์ เพื่อเป็นการลดค่าใช้จ่าย แต่การทำเช่นนี้ทำให้เกิดปัญหา " fat client "

ไคลเอนท์ที่มีปัญหา fat client นี้เกิดจากการที่ไคลเอนท์ไม่สามารถรองรับ ขนาดของข้อมูล และงานของผู้ใช้ที่มีจำนวนมากขึ้นได้ เพราะว่า งานของไคลเอนท์ไม่ได้มีแค่แสดงข้อมูลให้เห็นเท่านั้น แต่ยังมีการดึงข้อมูลอื่น ๆ จำนวนมากที่ไม่เกี่ยวข้องเลยกับงานนั้น ๆ มาด้วย และในกรณีที่มีการเปลี่ยนแปลง ฟังก์ชันการทำงานบางส่วน ผู้ใช้จำเป็นต้องมีการเปลี่ยนแปลง, ทดสอบ และแจกจ่ายโปรแกรมในส่วนของไคลเอนท์ที่ปรับปรุงแล้วไปยังไคลเอนท์ทุกเรื่อง

การทำงาน client /server แบบ ทรีเทียร์ (Three - Tier Application)

เพื่อแก้ไขปัญหาของทูเทียร์ จึงเพิ่มจากสองเทียร์เป็นสามเทียร์ โดยในแบบทูเทียร์เดิม ไคลเอนท์จะติดต่อโดยตรงกับฐานข้อมูล หากมีการเปลี่ยนแปลงใด ๆ เกิดขึ้นในฐานข้อมูล การแสดงผลทางด้านไคลเอนท์จำเป็นต้องเปลี่ยนแปลงตามไปด้วย ในการแก้ปัญหานี้ จะเพิ่มเทียร์ใหม่เข้ามาขั้นระหว่าง ไคลเอนท์และเซิร์ฟเวอร์ โดยไคลเอนท์จะติดต่อกับเซิร์ฟเวอร์โดยผ่านทางออบเจคที่อยู่บนมิดเดิลเทียร์ จากนั้นมิดเดิลเทียร์จะติดต่อกับเซิร์ฟเวอร์ โดยไคลเอนท์จะเห็นเฉพาะออบเจคในมิดเดิลเทียร์เท่านั้น การเปลี่ยนแปลงใด ๆ จะต้องทำผ่านมิดเดิลเทียร์เท่านั้น

การทำงาน client/server แบบ มัลติเทียร์ (Multi - Tiered Application)

โปรแกรมแอพพลิเคชั่น โดยทั่วไปที่ใช้งานอยู่ จะประกอบด้วยส่วนที่ติดต่อกับผู้ใช้ (userInterface)สำหรับแสดงผลและเก็บรวบรวมข้อมูล เข้ามา กลุ่มของฟังก์ชันต่าง ๆ ที่ทำหน้าที่ประมวลผลข้อมูลและแบ่งงานต่าง ๆ รวมถึง วิธีการเก็บรักษาข้อมูล ถึงแม้ว่าฟังก์ชันที่ใช้ในการ เก็บรักษาข้อมูล โดยทั่วไปจะทำงานอยู่ภายใต้เซิร์ฟเวอร์ (server) ของฐานข้อมูลส่วนกลาง บางครั้งเรียกรูปแบบลักษณะการทำงานแบบนี้ว่า เป็นโมเดล แอพพลิเคชั่นแบบ 2 ระดับ (two - tier application model), ซึ่งโปรแกรมแอพพลิเคชั่นแบบเก่า จะเป็นโปรแกรมเดียวซึ่งทำงานบนเครื่องของผู้ใช้ เนื่องจากโปรแกรมแอพลิเคชั่นที่ทำงานเดี่ยว ๆนั้นมีขนาดใหญ่มาก จึงพัฒนาได้ช้า , และบำรุงรักษามากอีกทั้งยังใช้เนื้อที่ฮาร์ดดิสก์ สูง มาก เพียงแค่มีการเปลี่ยนแปลงเล็กน้อย ก็จะต้องมีการเขียนโปรแกรมไปใหม่ , คอมไพล์ใหม่ และเนื่องจากโปรแกรมแอพพลิเคชั่นเหล่านี้ เขียนขึ้นมาเพื่อใช้งานกับระบบที่มีลักษณะต่างกัน จึงไม่สามารถที่จะเปลี่ยนไปใช้งานบนระบบที่แตกต่างไปได้ วิธีการแก้ไขปัญหาดังกล่าวมาทำได้โดยการแบ่ง โปรแกรมแอพพลิเคชั่นเดี่ยว ๆ นี้ ออกเป็นโมดูลย่อย ๆที่ทำงานร่วมกัน การแยกส่วนที่ติดต่อกับผู้ใช้ออกมาจากฟังก์ชันอื่น ๆ ในโปรแกรมแอพพลิเคชั่น ทำให้สามารถสร้างโปรแกรมไคลเอนท์ (client) เล็ก ๆ ซึ่งไม่ซับซ้อนและไม่ต้องทำงานมากเกินไป บนเครื่อง
ของผู้ใช้ โดยในโมดูลนี้จะใช้ฮาร์ดดิสส์ บนเครื่องของผู้ใช้น้อยกว่าและสามารถพัฒนา และบำรุงรักษาได้ง่ายกว่า ตัวอย่างเช่น ส่วนที่ติดต่อกับผู้ใช้สามารถเปลี่ยนแปลงได้โดยไม่ต้องเขียนโปรแกรมแอพพลิเคชั่นใหม่ทั้งหมดในทางทฤษฎีแล้ว ส่วนโมดูลที่ติดต่อกับผู้ใช้นี้

สามารถเขียนได้โดยใช้ ภาษาที่ต่างกัน เช่น จาวา(Java)จึงทำให้ใช้ได้บนเครื่องที่แตกต่างกัน ไคลเอนท์โมดูลออกแบบโดยใช้ จาวาแอพเพลต(Java applet) จึงไม่ต้องการเนื้อที่บนฮาร์ดดิสก์เพื่อการติดตั้ง

java IDL สนับสนุนซอฟต์แวร์ที่ออกแบบเป็นโมดูล แต่ละโมดูลออกแบบให้รองรับแอพพลิเคชั่นได้หลายแบบ แบ่งลักษณะของโมดูลดังกล่าวได้ 3 ประเภทใหญ่ ๆ คือ

User - Interface (Client) Tier

ส่วนที่ติดต่อกับผู้ใช้โดยโมเดล แอพพลิเคชั่นแบบหลายระดับ จะรวมไปถึงส่วนที่ติดต่อกับผู้ใช้แบบกราฟฟิก (GUI - Graphics User interface) สำหรับแอพพลิเคชั่น ทั้งแบบดั้งเดิมและแบบพื้นฐาน สร้างมาเพื่อให้ทำงาน กับผู้ใช้ได้เร็วและได้ผลลัพธ์ที่ถูกต้อง โดยเหมาฟังก์ชันที่ทำงานเกี่ยวกับ GUI ออกจากส่วนที่ให้บริการทางเครือข่าย การทำงานได้สอดคล้องกันของ ส่วนที่ติดต่อกับผู้ใช้จะช่วยลดปัญหาในการเรียนรู้ เพื่อใช้งานแอพพลิเคชั่นใหม่ ๆ , ทำงานร่วมกันกับแอพพลิเคชั่นได้ดีและให้ผลลัพธ์ที่มีคุณภาพสูงขึ้น แอพพลิเคชั่นแบบGUI นี้ สามารถใช้ได้สำหรับงานทั่วไปของผู้ใช้ เช่น บนเครื่องในระบบเครือข่าย , หรือบนอินเตอร์เน็ต

Server (server) Tier

ส่วนที่ให้บริการหรือส่วน เซอร์เวอร์ นี้เป็นส่วนสำคัญ ของแอพพลิเคชั่น เป็นส่วนกลางซึ่งคอยให้บริการการใช้แอพพลิเคชั่น และการสร้างแอพพลิเคชั่น ซึ่งการให้บริการนี้มีอยู่ในเครือข่าย และสามารถเข้าใช้ได้จากแอพพลิเคชั่น ทุกระดับ

Data Store (Database)Tier

แอพพลิเคชั่นแบบหลายระดับ (Multi - tier) นี้จะแยกการติดต่อ เข้าใช้ข้อมูลออกจากส่วนเซอร์เวอร์เรียกส่วนที่แยกออกมานี้ว่า "data store tier" มีออพชั่น หลายแบบที่ใช้เก็บและติดต่อ ใช้ข้อมูลเพื่อช่วยให้ พิจารณา

ผู้พิจารณาสามารถใช้กลุ่มข้อมูล ทีมีความสำคัญที่สุดเป็น อันดับแรกสุดออบเจคที่สร้างขึ้นมาจะใช้เพื่อสนองความต้องการในการใช้ข้อมูลต่าง ๆ ประกอบด้วย ความสามารถในการเลือกใช้ข้อมูลใน RDBMS (RelationDatabase Management System) หรือ OODMS (Object Oriented Database Management System)

6.6.4 ชนิดของ JDBC

JDBC drivers ที่ทราบในเวลานี้มีอยู่ 4 แบบด้วยกันคือ

JDBC/ODBC bridge.

Native - API ,partly Java driver.

Network-protocol , all - Java driver.

Native Protocol , all - Java driver.

JDBC / ODBC bridge (ชนิดที่ 1)

JDBC/ODBC bridge ได้ถูกพัฒนาขึ้นโดยการร่วมมือกันระหว่าง JavaSoft และ Intersolv เพื่อที่จะเพิ่มความสามารถให้กับฐานข้อมูลจำนวนมาก ที่ใช้ ODBC ในส่วนของ client Java applet หรือโปรแกรมอื่นจะถูกเขียนขึ้นโดย JDBC API โดย bridge ตัวนี้จะทำการแปลงคำสั่งจาก JDBC ไปเป็นคำสั่งของ ODBC แล้วส่งคำสั่งนั้นไปยังตัว ODBC driver เพื่อจัดการกับฐานข้อมูล ข้อได้เปรียบหลักของ bridge ตัวนี้คือ โปรแกรมที่เขียนขึ้นนั้นจะง่ายต่อการเข้าถึงข้อมูล จากระบบฐานข้อมูลของผู้ผลิตต่าง ๆ โดยการเลือก ODBC driver ที่เหมาะสม อย่างไรก็ตาม ตัวติดต่อกับฐานข้อมูลชนิดนี้ทำให้ต้องนึกถึงผลข้างเคียงและความยุ่งยากที่อาจเกิดขึ้น เพราะว่าคำสั่งต้องส่งจาก JDBC ไปยังbridge ที่เชื่อมไปยัง ODBC driver และสุดท้ายจาก ODBC ก็ส่งไปยัง native client - API เพื่อไปยังฐานข้อมูล ตัว driver ชนิดนี้ทำให้ Java applets ส่งข้อมูลไปไม่ทันทีทันใด กับที่เรียกไปเพราะ code หลัก ๆ ต้องถูกติดตั้งไว้ก่อนแล้ว บนทุก ๆ เครื่องของ client ที่ต้องการใช้การเชื่อมต่อด้วย JDBC/ODBC bridge เพื่อช่วยเหลือคำสั่งของตัว API จากการที่ต้องการติดตั้งโปแกรมก่อน ทำให้การจัดการด้านการติดต่อของ client - server เป็นภาระที่หนักมาก ดังนั้น JDBC/ODBC Bridge จึงไม่ได้แก้ปัญหาการเปลี่ยนแปลงของ client program

Native - API , Partly Java Driver(ชนิดที่ 2)

เป็นแบบ two - tier นั่นคือ JDBC driver ต้องการ library เพื่อแปลงฟังก์ชันของ JDBC ไปเป็น querylanguage ต่าง ๆ ของ DBMS (เช่น library สำหรับ Sybase หรือ dbilb , สำหรับ Oracle คือ ocilib และอื่น ๆ) driver เหล่านี้โดยปกติจะเขียนขึ้นโดยภาษา Java และ C/C++

เนื่องจาก driver ต้องใช้ layer ของ C ในการเรียกไปยัง library druver ชนิดที่ 2 เช่น JDBC/ODBC Bridge ต้องการให้ code (ซึ่งคือ library ของแต่ละclient ดังนั้นจึงมีปัญหา ทางด้านการการดูแลซอฟต์แวร์ เช่นเดียวกับ bridge อย่างไรก็ตาม driver ชนิดที่ 2 นี้จะเร็วกว่าชนิดที่ 1 เพราะ layer พิเศษของการแปลงเป็น ODBC ถูกเอาออกไป

Network - Protocol, All - Java Driver (ชนิดที่ 3)

driver ชนิดที่ 3 นี้แปลงการเรียก JDBC ไปเป็นโปรโตคอลเครือข่ายฐานข้อมูลอิสระ ซึ่งจะแปลงไปเป็นการเรียก database - specific API โดย middle-tier server (middle-tier server อาจใช้ driver ชนิดที่ 1 หรือชนิดที่ 2 ถ้าเขียนโดย Java) สถาปัตยกรรมโดยรวมประกอบ ด้วย 3 tiers คือ JDBC client และ driver , middleware, และฐานข้อมูลที่ถูกเข้าถึง JDBC driver ขนาดเล็ก (200 KB หรือน้อยกว่า) ทำงานบน client และมีการใช้ login ในการส่งผ่านคำสั่ง SQL ในเครือข่ายไปยัง JDBC server , รับข้อมูลกลับจาก sever, และจัดการติดต่อโดย driver ชนิดที่ 3 นี้จะมีลักษณะเป็น just-in-time client deployment

JDBC server จัดการติดต่อหลายอย่างกับฐานข้อมูล รวมทั้งการยกเว้นและสถานะของเหตุการณ์ที่มีผลมาจากการทำคำสั่ง SQL JDBC Server ยังจัดรูปแบบข้อมูลสำหรับการส่งในเครือข่ายไปยัง JDBC client

middleware server สามารถ implement เป็นส่วนประกอบดั้งเดิมหรือเขียนโดย Java การใช้แบบดั้งเดิมติดต่อกับ server ฐานข้อมูล ใช้ client library ของผู้ผลิตหรือ ODBC เช่น dbAnywhere ของ Symantec และ SequeLink ของ Intersoft ถึงแม้ว่า SequeLink ไม่ต้องการ client library ของฐานข้อมูลติดตั้งบน server แต่จะใช้ library ของตัวเอง server จะต้องตั้งค่าสำหรับฐานข้อมูลที่จะเข้าถึง ซึ่งอาจเกี่ยวข้องกับการตั้งเลขพอร์ต, ตัวแปรสภาพแวดล้อมต่าง ๆ ของฐานข้อมูล (เช่น DSQuery กับ Sybase) , พารามิเตอร์เฉพาะของฐานข้อมูล (การ log , การแปลง) และพารามิเตอร์อื่น ๆ ที่ server ต้องการ ถ้า middleware เขียนโดย Java , จะสามารถใช้ JDBC - compliant server ในการสื่อสารกับ DBMS โดยผ่านโปรโตคอลฐานข้อมูลของผู้ผลิต

Type III driver เหมาะสำหรับแอพพลิเคชั่นที่มีหลายผู้ใช้บนอินเตอร์เน็ต/อินทราเนต มากที่สุด ที่ซึ่งการกระทำของข้อมูลต่อเนื่องจำนวนมาก เช่น queries, searches และอื่น ๆ ถูกคาดหมายประสิทธิภาพเป็นสิ่งสำคัญServer สามารถจัดการฐานข้อมูลจำนวนมากรวมกันได้ ,

สามารถให้การตรวจสอบและดูแลข้อมูล, สามารถทำload balancing และสนับสนุน catalog และ query caches และอย่างที่ได้กล่าวไปแล้ว แอพพลิเคชั่นบน web แบบ three-tier เกี่ยวข้องกับความปลอดภัย, firewalls และ proxies ซึ่ง Type III driver สนับสนุนสิ่งเหล่านี้

สิ่งที่เป็นข้อเสียของ network - centric driver ก็คือส่วนประกอบของ server เป็นแบบ middleware ผู้ผลิตแต่ละรายใช้ middleware ของตนเอง สำหรับการติดต่อในเครือข่าย

Native-Protocol, All-Java Driver(ชนิดที่ 4)

จะแปลงการเรียก JDBC ไปเป็นโปรโตคอลเครือข่ายโดยตรง โดยใช้ไดรฟ์เวอร์ซึ่งเขียนขึ้นโดยเฉพาะโดย ไดรฟ์เวอร์เหล่านี้สามารถเขียนโดย Java ทั้งหมด และสามารถให้การส่งข้อมูลแบบ just-in-time ของแอพเพลท(เหมือน Type III) เนื่องจากไดรฟ์เวอร์เหล่านี้แปลง JDBC . ไปเป็นโปรโตคอลโดยตรงไม่มีการใช้ODBC หรือ API ดั้งเดิม จึงสามารถให้การเข้าถึงฐานข้อมูลที่มีประสิทธิภาพสูง ไดรฟ์เวอร์เหล่านี้ทำขึ้นจากผู้ผลิต DBMS เท่านั้น จากความจริงที่ว่าความรู้ในเรื่องโปรโตคอลเป็นของผู้ผลิต ในปัจจุบันยังมี Type IV ใช้อยู่น้อย แต่จำนวนน่าจะมากขึ้นในเดือนต่อ ๆมา

บทที่ 7
การพัฒนาโครงงาน

การปฏิบัติงานในโครงงานนี้มีลำดับขั้นการพัฒนาตามหัวข้อดังต่อไปนี้

1. จัดเตรียมเครื่องคอมพิวเตอร์ที่จะนำมาใช้เป็นเครื่อง Server โดยได้จัดเตรียมเป็นเครื่องคอมพิวเตอร์ที่มีคุณสมบัติเบื้องต้นดังต่อไปนี้
   - Computer Compaq
   - CPU Pentium II 233 Mhz
   - RAM 128 Mbyte
   - Harddisk 4 Gbyte

2. ติดตั้งระบบปฏิบัติการ (OS) โดยในโครงงานนี้เลือกใช้ Solaris 2.6 for X86 มาเป็นระบบปฏิบัติการบนเครื่อง Server

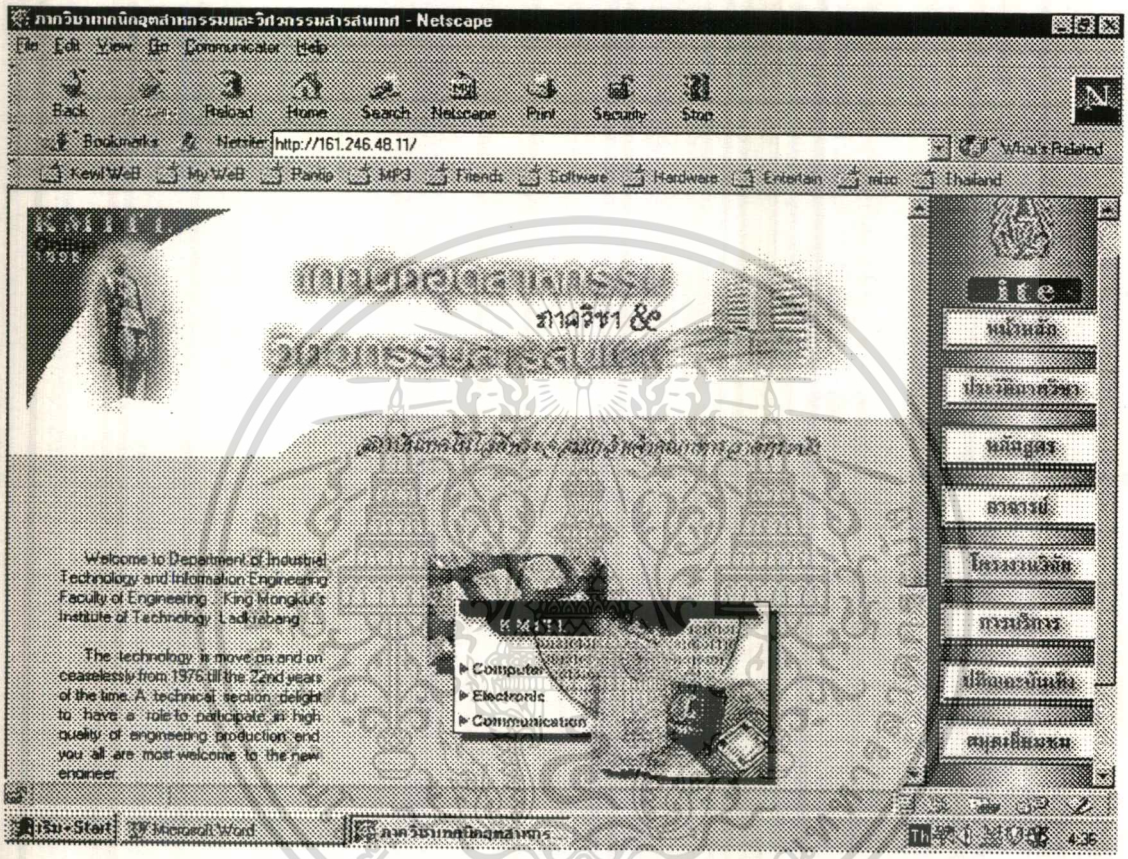
3. ติดตั้งโปรแกรม Web server โดยใช้โปรแกรมของ Sun Web Server มาเป็นโปรแกรมของระบบให้บริการ

4. เซตค่า Configuration ของระบบ FTP Server ให้ Anonymous user สามารถใช้งานได้ เพื่อให้นักศึกษาสามารถ Download เอกสารประกอบการศึกษาในรายวิชาต่างๆ จากอาจารย์ได้

5. การศึกษาการเขียน Home page โดยในโครงงานนี้ได้ทำการเขียนเป็น Home page ของภาควิชาเทคนิคอุตสาหกรรม และวิศวกรรมสารสนเทศ เพื่อเป็นการประชาสัมพันธ์ เผยแพร่ข้อมูลข่าวสาร ให้การบริการและเพื่อความบันเทิงในด้านต่าง ๆ แก่อาจารย์ นักศึกษาภายในภาควิชา และบุคคลที่สนใจอื่น ๆ ทั่วโลก

โดยเนื้อหาสาระของHome page นี้จะประกอบไปด้วยหัวข้อและรายละเอียดต่าง ๆ ดังต่อไปนี้

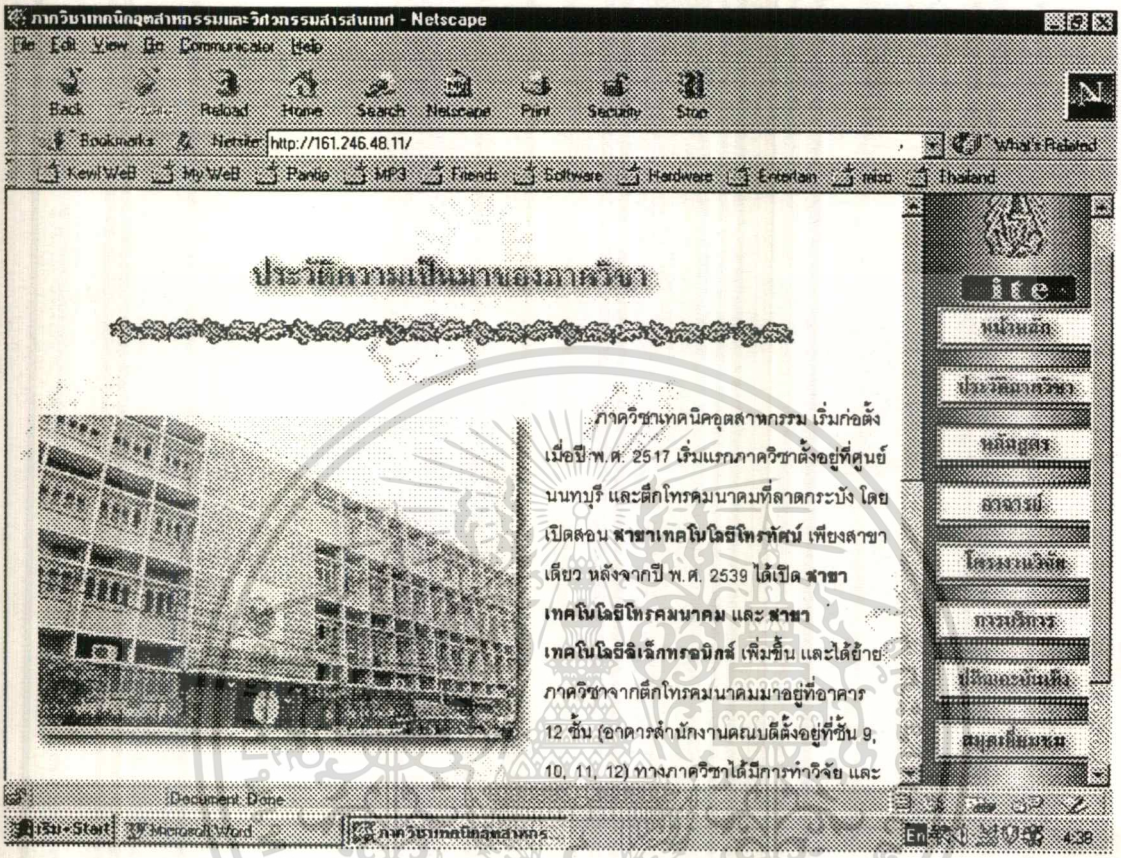
1. หน้าหลัก



รูปที่ 7.1 แสดงรูปหน้าจอในส่วนของหน้าหลัก

หน้าหลัก เป็นหน้าจอแรกที่จะบอกถึงเจ้าของ Home page ซึ่งโดยทั่วไปในการเขียน Home page นั้น หน้าจอแรกก็ควรจะประกอบด้วย โลโก้ หรือชื่อของหน่วยงานเจ้าของ Home page นั้น นอกจากนั้นยังจะต้องประกอบด้วย ส่วนที่หัวข้อของรายละเอียดเนื้อหาในหน้าต่อไปด้วย โดยอาจจะทำเป็นแถบ Menu ชื่อหัวข้อเรื่อง ซึ่งเมื่อเอาเมาส์ไปคลิ้กบนหัวข้อนั้นแล้วก็จะปรากฏเป็นรายละเอียดต่าง ๆ ของหัวข้อนั้น เป็นต้น

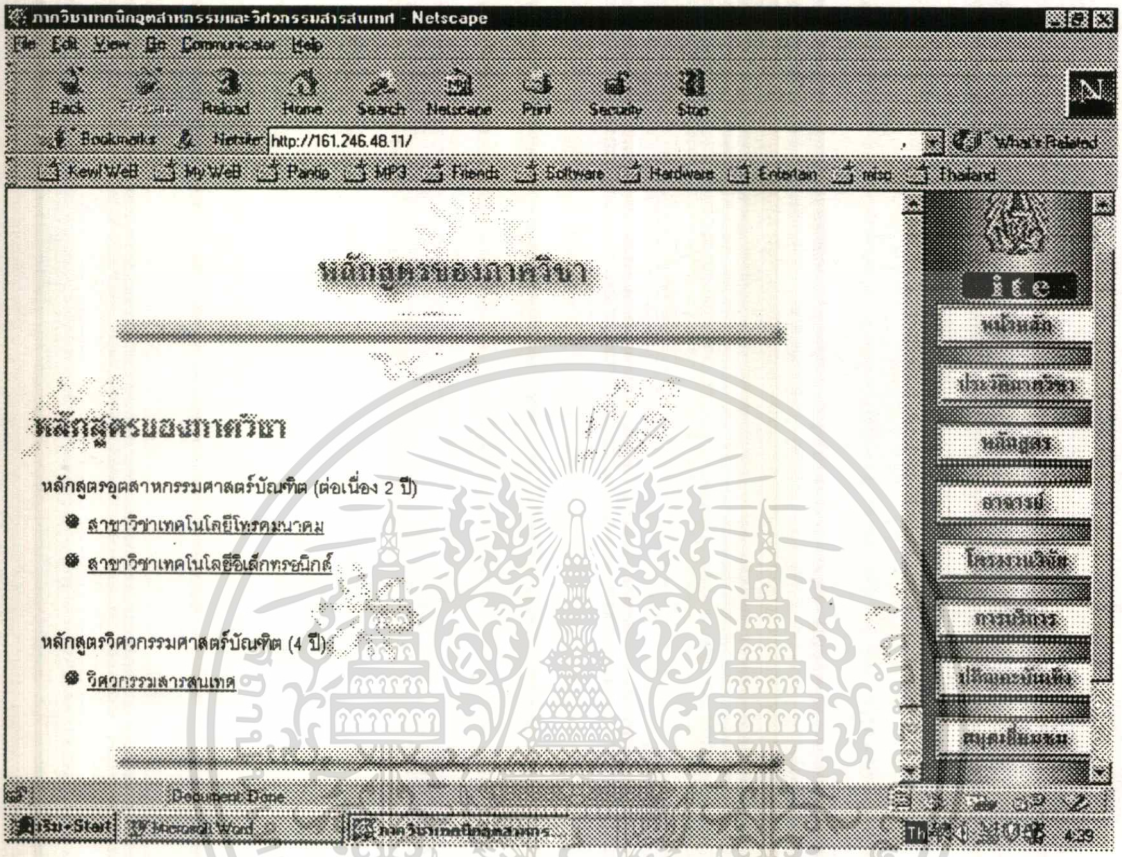
2. ประวัติภาควิชา



รูปที่ 7.2 แสดงรูปหน้าจอในส่วนของประวัติภาควิชา

ประวัติภาควิชา ในหัวข้อนี้จะเป็นการนำเสนอถึงประวัติความเป็นมาต่าง ๆ ตั้งแต่สมัยเริ่มก่อตั้งภาควิชาเทคนิคอุตสาหกรรมตลอดเรื่อยมาจนถึงปัจจุบันได้ขยายหลักสูตรเพิ่มเป็นวิศวกรรมสารสนเทศ และในหน้านี้ยังได้แสดงถึงรายนามของอาจารย์หัวหน้าภาควิชาตั้งแต่ต้นมาจนถึงท่านอาจารย์ที่กำลังดำรงตำแหน่งค่นปัจจุบัน

3. หลักสูตร



รูปที่ 7.3 แสดงรูปหน้าจอในส่วนของหลักสูตร

หลักสูตร ในหัวข้อนี้เป็นส่วนของหลักสูตรการศึกษา ซึ่งได้แบ่งเป็น 3 หลักสูตรดังนี้

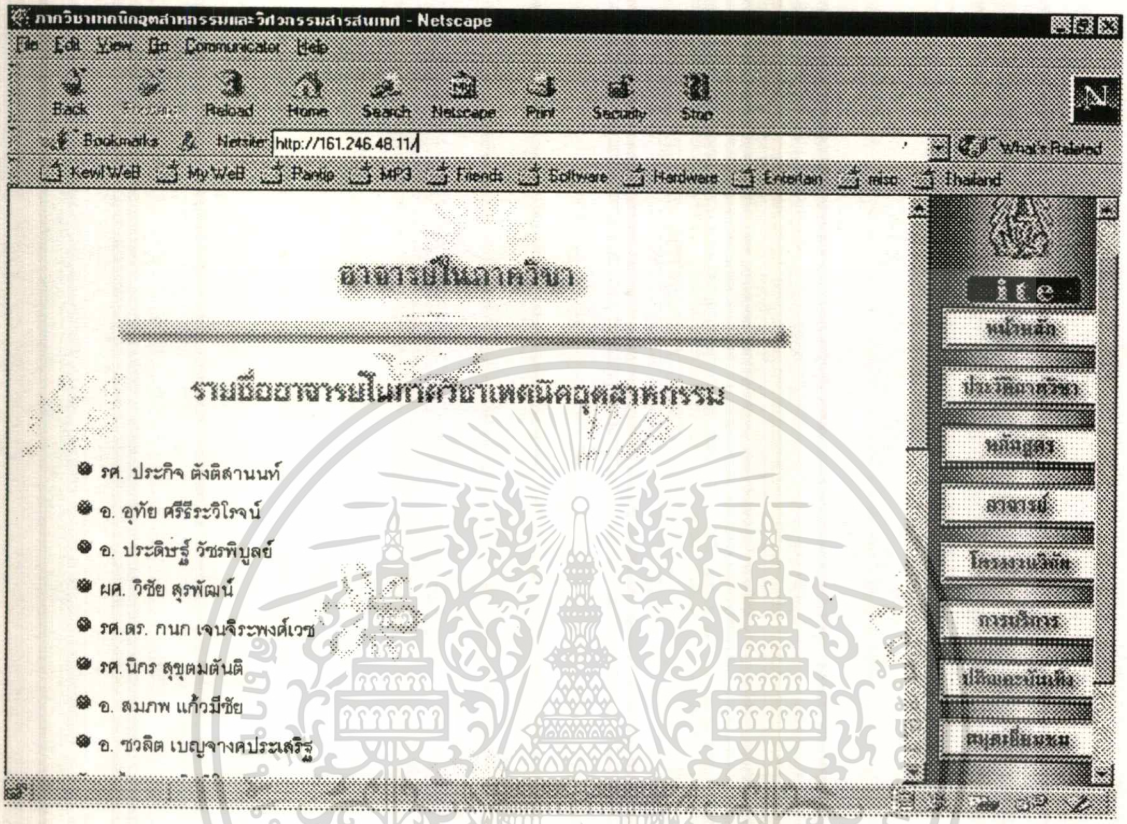
- สาขาวิชาเทคโนโลยีโทรคมนาคม
- สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์
- วิศวกรรมสารสนเทศ

ซึ่งในแต่ละหัวข้อสามารถที่จะ Link ไปยังรายละเอียดของแต่ละหลักสูตรได้โดยจะประกอบด้วยรายละเอียดต่าง ๆ มากมาย เช่น

วัตถุประสงค์ของหลักสูตร

รายวิชาของแต่ละหลักสูตร เป็นต้น

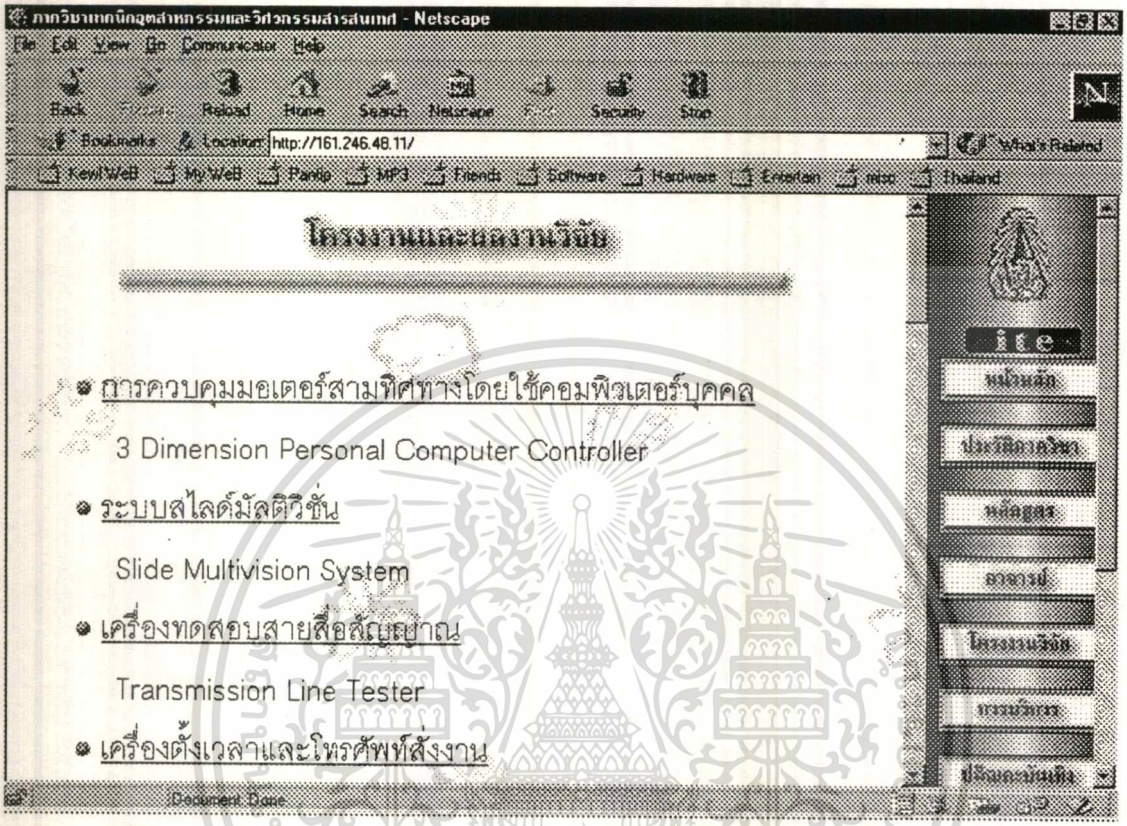
4. อาจารย์



รูปที่ 7.4 แสดงรูปหน้าจอในส่วนอาจารย์ในภาควิชา

อาจารย์ หน้านี้จะแสดงรายชื่อทั้งหมดของอาจารย์ในภาควิชาเทคนิคอุตสาหกรรม ซึ่งในหน้านี้ได้ทำให้มีการ Link ไปยัง Home page ย่อยของอาจารย์แต่ละท่านได้ เพื่อที่จะเป็นการลงข้อความนัดหมายนักศึกษา หรือให้โจทย์การบ้านแก่นักศึกษา รวมถึงการที่จะให้นักศึกษา ทำการ Download เอกสารที่ใช้ในการประกอบการศึกษาจากอาจารย์ได้อีกด้วย

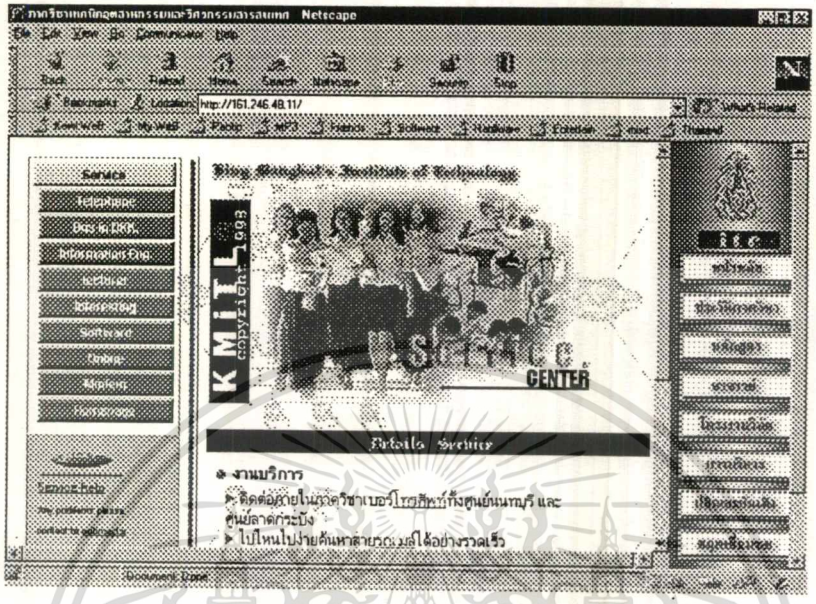
5. โครงงานวิจัย



รูปที่ 7.5 แสดงรูปหน้าจอในส่วนโครงงานวิจัย

โครงงานวิจัย หัวข้อนี้จะเป็นการเสนอความสามารถของนักศึกษาในภาควิชาเทคนิคอุตสาหกรรมให้บุคคลทั่วไปได้รับรู้ โดยได้รวบรวมเอาหัวข้อโครงงานของนักศึกษาในภาคที่น่าสนใจมาเสนอ โดยสามารถคลิ้กเมาส์ลงบนหัวข้อโครงงานที่สนใจแล้วก็จะได้พบกับรายละเอียดซึ่งประกอบด้วย บทคัดย่อ ชื่อนักศึกษาเจ้าของโครงงาน และชื่อท่านอาจารย์ที่ปรึกษาโครงงานนั้น

6. การบริการ



รูปที่ 7.6 แสดงรูปหน้าจอในส่วนของการบริการ

การบริการ ในหน้านี้จะเป็น Link ที่จะนำไปสู่บริการการอำนวยความสะดวกในด้านต่าง ๆ ซึ่งประกอบด้วยหัวข้อดังต่อไปนี้

บริการเบอร์โทรศัพท์ติดต่อของภาควิชา

บริการค้นหาสายรถเมล์

บริการข่าวสาร Engineering Lab

บริการติดต่ออาจารย์

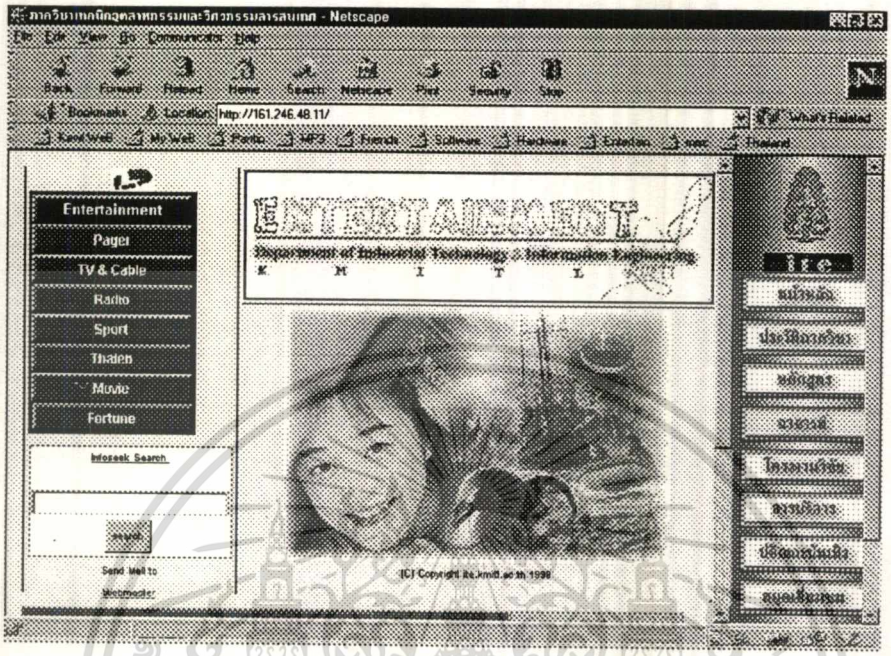
บริการสาระน่ารู้น่าสนใจ

บริการโปรแกรมที่จำเป็น

บริการแหล่งข้อมูลการค้นคว้า

บริการติดต่อสื่อสารผ่านโมเด็ม

7. ปกิณกะบันเทิง



รูปที่ 7.7 แสดงรูปหน้าจอในส่วนของปกิณกะบันเทิง

ปกิณกะบันเทิง หน้านี้เป็นแหล่งรวบรวม Link ที่จะไปสู่ความบันเทิงในรูปแบบต่างๆ ดังนี้

ส่งข้อความทางเพจเจอร์

สถานีโทรทัศน์และเคเบิ้ลทีวี

สถานีวิทยุ

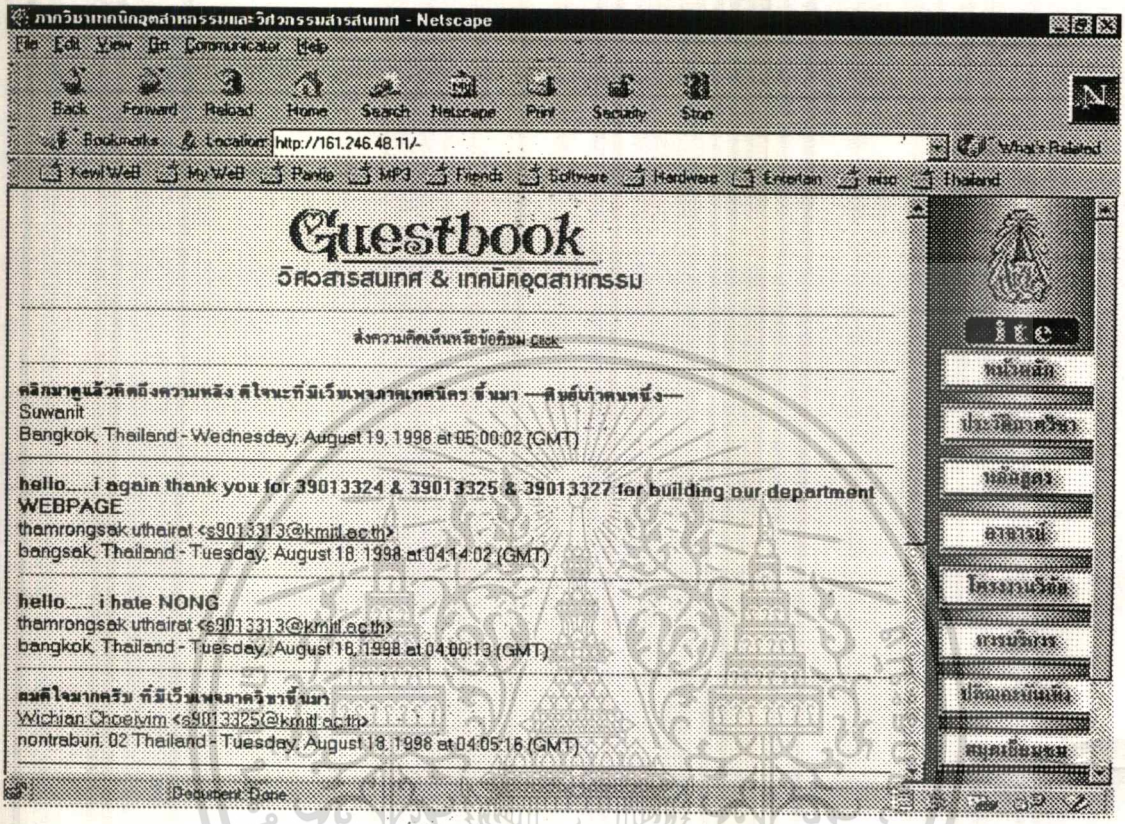
Web site เกี่ยวกับกีฬา

ผู้ผลิตสื่อความบันเทิง

Web site เกี่ยวกับภาพยนต์

สนุกกับการดูดวง

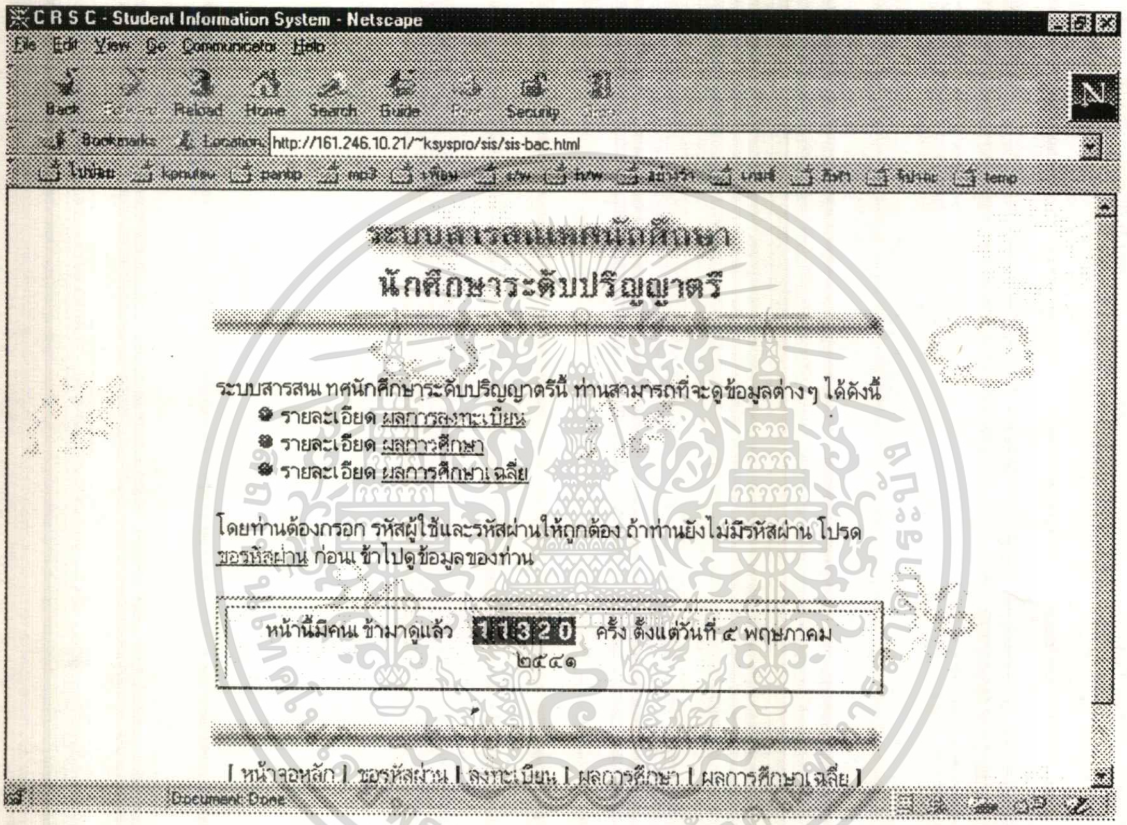
8. สมุดเยี่ยมชม



รูปที่ 7.8 แสดงรูปหน้าจอในส่วนของสมุดเยี่ยมชม

สมุดเยี่ยมชม เป็นประตูรับความคิดเห็นข้อเสนอแนะข้อติชม ต่าง ๆ จากผู้ที่เข้ามาใช้บริการ ซึ่งมีประโยชน์ในการที่จะนำเอาข้อมูลเหล่านี้มาพัฒนาปรับปรุง Home page ให้ดีขึ้น ให้เป็นที่สนใจมากยิ่งขึ้น

6. ด้านการติดต่อกับฐานข้อมูล Oracle โดยมีการใช้ ภาษา JAVA มาช่วยในการที่จะทำให้ Home page มีคุณสมบัติในการ ขอดูข้อมูลระบบสารสนเทศของนักศึกษา เช่น ระบบลงทะเบียน ระบบประมวลผลการศึกษา โดยใช้ JDBC เป็นตัวเชื่อมต่อระหว่าง Web Server กับ Data base Server



รูปที่ 7.9 แสดงรูปหน้าจอในส่วนของระบบสารสนเทศนักศึกษาระดับปริญญาตรี

บทที่ 8
บทวิจารณ์และสรุป

8.1 สรุปและวิจารณ์โครงงาน

โครงงานนี้ได้จัดทำขึ้นเพื่อต้องการให้มีระบบ Web Server ที่จะเป็นเครื่องให้บริการถ่ายโอนข้อมูลเอกสารแบบ HTML ที่ได้รับการร้องขอจากเครื่อง Client หรือ Browser โดยข้อมูลที่ทำการถ่ายโอนนี้จะเป็นได้ทั้งรูปแบบตัวอักษร ภาพ เสียง และข้อมูลในรูปแบบมัลติมีเดียอื่น ๆ โดยที่ Browser และ Server จะสามารถติดต่อกันได้ก็โดยผ่านโปรโตคอล HTTP ดังนั้นในการติดตั้ง Web Server จึงเป็นการติดตั้งเพื่อให้บริการ HTTP นั่นเอง โดยในโครงงานนี้ได้เลือกใช้ Solaris 2.6 for X86 เป็นระบบปฏิบัติการ และใช้ Sun Web Server เป็นระบบให้บริการ การติดต่อสื่อสารข้อมูลในระบบเครือข่าย www นี้ ยังมีสิ่งสำคัญอยู่ประการหนึ่งที่จะละเลยเสียไม่ได้ ก็คือระบบการรักษาความปลอดภัยของข้อมูล เพราะมันเป็นส่วนหนึ่งที่จะทำให้เครือข่ายอินเตอร์เน็ตและ www เป็นที่นิยมกัน โดยที่เครือข่ายใดที่มีความปลอดภัยในการใช้งานที่ดีพอก็จะมีผู้ที่นิยมใช้กันอย่างมากขึ้น โดยหลักในการทำงานของระบบการรักษาความปลอดภัยของข้อมูลนั้นก็จะประกอบไปด้วย การเข้ารหัสข้อมูล การพิสูจน์ความแท้จริงของ Server และการตรวจสอบความถูกต้องและความสมบูรณ์ของข้อมูล เป็นต้น

ในโครงงานนี้เมื่อทำการติดตั้งระบบ Web Server เสร็จแล้วก็ยังได้มีการเขียน Web Page ขึ้นมาด้วย เพื่อเป็นการประชาสัมพันธ์ เผยแพร่ข้อมูลข่าวสาร และเพื่อความบันเทิงในด้านต่าง ๆ ให้กับภาควิชาเทคนิคอุตสาหกรรม และวิศวกรรมสารสนเทศ โดยจะประกอบด้วยเนื้อหาดังต่อไปนี้

1. หน้าหลัก
2. ประวัติภาควิชา
3. หลักสูตร
4. อาจารย์
5. โครงงานวิจัย
6. การบริการ
7. ปกิณกะบันเทิง
8. สมุดเยี่ยมชม

การเก็บข้อมูลโดยใช้ Web Page เป็นวิธีการที่มีประโยชน์อย่างมาก โดยสามารถสรุปเป็นหัวข้อซึ่งได้จากการทดลองทำ Web Page ของภาควิชาได้ดังนี้

1. การเก็บข้อมูลโดยใช้ Web Page จะทำให้สามารถค้นหาข้อมูลได้โดยง่าย เพราะมีการเชื่อมโยงข้อมูลที่เกี่ยวข้องเข้าไว้ด้วยกัน จึงทำให้ง่ายต่อการค้นหา

2. การเก็บข้อมูลด้วยวิธีนี้สามารถที่จะกระจายความรู้ได้อย่างกว้างขวาง ทำให้มีผู้เข้ามาอ่านข่าวสารข้อมูลได้ทั่วโลก (World Wide) เป็นการเผยแพร่ความรู้ที่มีประสิทธิภาพมากที่สุดในปัจจุบัน

3. การออกแบบหน้าจอที่หน้าสนใจจะทำให้ผู้ที่ใช้บริการข้อมูลข่าวสารมีความอยากจะอ่าน Web Page นั้น ๆ มากยิ่งขึ้น ดังนั้นในการเขียน Web Page หน้าแรก (Home Page) จึงจำเป็นต้องออกแบบให้ผู้ที่พบเห็นเกิดความสนใจที่จะเข้าไปอ่านในหน้าต่อ ๆ ไป

4. การมีรูปภาพเคลื่อนไหวประกอบในหน้า Web Page ก็จะทำให้ Web Page เป็นที่น่าสนใจมากยิ่งขึ้น

5. การที่มีระบบเก็บข้อมูลของผู้ที่เข้ามาใช้บริการข่าวสารข้อมูลนั้นจะทำให้สามารถทราบถึงจำนวนผู้ที่เข้ามาใช้บริการข้อมูล และความต้องการในข้อมูลของผู้ใช้บริการ ข้อมูลเหล่านี้มีประโยชน์เพื่อที่จะได้นำมาพิจารณาเพื่อการพัฒนาปรับปรุงขยายข้อมูลให้ตรงตามความต้องการ หรือเป้าหมายของผู้ใช้บริการข้อมูลได้

8.2 ปัญหาที่พบ

เนื่องจากเครื่องคอมพิวเตอร์ที่ใช้ทำหน้าที่เป็นเครื่อง Server อยู่ในขณะนี้มีระดับการประมวลผลที่ยังช้าอยู่มาก จึงเป็นสาเหตุทำให้การติดต่อสื่อสารข้อมูลสามารถทำได้ช้า ต้องเสียเวลามากในการโหลดข้อมูลแต่ละครั้ง

8.3 ข้อเสนอแนะ

จากการทดลองหัวข้อเรื่อง www ในการติดตั้ง Web Server และการเขียน Home page ให้กับภาควิชานั้นเมื่อนำผลจากการทดลองลงมือปฏิบัติแล้วทางกลุ่มผู้จัดทำได้เกิดมีแนวความคิดที่เป็นข้อเสนอแนะดังต่อไปนี้

ทางด้านการเขียน Home page

ก่อนที่จะลงมือสร้าง Home page นั้นควรที่จะมีการวางแผนการสร้างให้ดีก่อน เพราะว่าการสร้าง Home page ที่ดีได้นั้นนอกจากจะคำนึงถึงความสวยงามแล้วยังควรจะต้องมีคุณสมบัติอื่น ๆ มาประกอบด้วยโดยจะข้อแนะนำเป็นหัวข้อดังนี้

1. มีรายการสารบัญแสดงรายละเอียดของเว็บเพจนั้น
2. เชื่อมโยงข้อมูลไปยังเป้าหมายได้ตรงกับความต้องการมากที่สุด
3. มีเนื้อหาที่สั้น กระชับ และทันสมัย
4. สามารถโต้ตอบกับผู้ใช้ได้
5. มีรูปภาพประกอบการนำเสนอที่ดี แต่ไม่ควรมีรูปภาพมากเกินไป
6. เข้าสู่กลุ่มเป้าหมายได้อย่างถูกต้อง
7. ใช้งานง่าย
8. เป็นมาตรฐานเดียวกัน

8.4 แนวทางในการพัฒนา

เนื่องจากว่าโครงงานนี้ในส่วนของการเขียน Home page ได้ประกอบด้วยส่วนที่เป็นการโต้ตอบกับผู้ใช้ในหัวข้อ สมุดเยี่ยมชม ซึ่งเป็นระบบเก็บบันทึกข้อมูลข้อคิดเห็น ข้อเสนอแนะ ตลอดจนข้อความติชมต่าง ๆ เกี่ยวกับโครงงานนี้จากผู้ใช้ ซึ่งทางกลุ่มผู้จัดทำมีความคิดเห็นว่าแนวทางในการพัฒนาสำหรับโครงงานนี้ก็ควรจะเป็นการที่ นำเอาข้อวิจารณ์ ข้อคิดเห็น ข้อเสนอแนะต่าง ๆ เหล่านี้มาเป็นแนวทางในการพัฒนาจะดีที่สุด เพื่อที่จะทำให้โครงงานนี้บรรลุเป้าหมายก็โดยที่มีผู้ที่ต้องการเข้ามาเยี่ยมชม มาศึกษาข้อมูล และมาเสนอข้อคิดเห็นบนเว็บเพจนี้ให้ได้มากที่สุด

หนังสืออ้างอิง

วารสาร

1. เอกรัฐ คงมาลัย และ สุรศักดิ์ สงวนพงษ์ , " Internet Server ", Internet magazine ฉบับที่ 22 , เมษายน 2541
2. ยุทธนา สวนสุข , " วิธีการติดตั้งเว็บเซิร์ฟเวอร์ " , Internet – Intranet , ฉบับที่ 3 , ธันวาคม 2539
3. ยุทธนา สวนสุข ," การเข้ารหัส SSLและ S- HTTP ", Internet – Intranet , ฉบับที่ 5 , เมษายน 2540

หนังสือ

1. Deborah S. Ray & Eric J. Ray , HTML FOR DUMMIES , IDG Books World Wide , Inc , 1996
2. ศรันย์ ไมตรีเวช , " FTP Explorer ", ครบเครื่องเรื่องอินเตอร์เน็ต , โปรวิชั่น , 2540
3. จิตเกษม พัฒนาศิริ , " การเขียน เอชทีเอ็มแอล " , เริ่มสร้างโฮมเพจด้วย HTML , วิตตี้ กรุ๊ป , 2539
4. ณัฐวัฒน์ ลิ้มล้อมวงศ์ , สร้างสรรค์เว็บเพจสวย ๆ ด้วย JAVA , ซีเอ็ด ยูเคชั่น , 2540
5. ศ.ดร. ศรีศักดิ์ จามรมาน , ดร. กนกวรรณ ว่องวัฒนะสิน , สยาม สงวนรัมย์ และ สหรัฐ สงวนรัมย์ , Starter Kit Internet All – in – One , ออนไลน์ แอดเวอร์ไทซิ่ง , 2537

ภาคผนวก

สรุปคำศัพท์

(Glossary)

- **เว็บ (Web)** เป็นคำเรียกโดยย่อของคำเต็มว่า World Wide Web หรือ ระบบบริการเชื่อมต่อข้อมูลเครือข่ายคอมพิวเตอร์ระดับโลก ทำหน้าที่ในการให้บริการถ่ายโอนข้อมูลข่าวสารในรูปของแฟ้มเอกสารแบบ HTML บนระบบเครือข่ายอินเตอร์เน็ต

- **เว็บเพจ หรือ โฮมเพจ (Web Page or Home Page)** คือข้อมูลที่ปรากฏบนโปรแกรมเว็บบราวเซอร์ หรือเป็นแฟ้มเอกสารแบบ HTML ที่ได้จากการเชื่อมโยงและโอนย้ายจากเว็บเซิร์ฟเวอร์แหล่งอื่น ๆ โดยทั่วไปแล้วคำว่าโฮมเพจจะเน้นถึงการเป็นข้อมูลในหน้าแรกของเวิล์ดไวด์เว็บเซิร์ฟเวอร์นั้น ๆ

- **ไฮเปอร์เท็กซ์ (Hypertext)** เป็นคำหรือวลีพิเศษของเว็บเพจ หรือ โฮมเพจ ซึ่งเกิดจากแฟ้มเอกสารแบบ HTML โดยเป็นจุดเชื่อมโยงไปยังแหล่งข้อมูลได้ด้วยการใช้เมาส์คลิกไปยังคำหรือวลีพิเศษนั้น

- **เอชทีเอ็มแอล (HTML)** เป็นภาษาสำหรับเขียนไฟล์ข้อมูลแบบไฮเปอร์เท็กซ์ ซึ่งเป็นข้อมูลที่ใช้ในระบบ เวิล์ดไวด์เว็บ หรือเป็นไฟล์ที่ใช้แสดงหน้าโฮมเพจ โดยไฟล์ข้อมูล HTML จะถูกกำหนดให้มีชื่อขยายเป็น .html ภายใต้ระบบปฏิบัติการยูนิกซ์ หรือเป็น .htm บนระบบ Windows

- **ไฮเปอร์ลิงก์ (Hyperlink)** เป็นการเชื่อมโยงเพื่อการโอนย้ายไฟล์ข้อมูลจากเว็บเซิร์ฟเวอร์มายังคอมพิวเตอร์ของผู้ใช้คอมพิวเตอร์นั้น ๆ ซึ่งเป็นไฟล์ข้อมูล HTML การเชื่อมโยงแบบไฮเปอร์ลิงก์ก็ทำได้โดยการใช้เมาส์คลิกไปยังข้อความที่ถูกกำหนดไว้

- **เอชทีทีพี (HTTP)** เป็นระบบสื่อสารเชื่อมโยงเพื่อโอนย้ายไฟล์ข้อมูล HTML ซึ่งเป็นข้อมูลที่ใช้ในระบบเวิล์ดไวด์เว็บ ดังนั้นการเชื่อมโยงเพื่อโอนย้ายไฟล์ในระบบเวิล์ดไวด์เว็บจึงต้องระบุรูปแบบของการสื่อสารเพื่อการเชื่อมโยงโดยรหัสสืบค้นยูอาร์แอล
- **ไคลเอนท์ (Client)** คือ ผู้ขอใช้บริการในระบบเครือข่ายคอมพิวเตอร์
- **เซิร์ฟเวอร์ (Server)** คือ ผู้ให้บริการในระบบเครือข่ายคอมพิวเตอร์
- **บราวเซอร์ (Browser)** เป็นโปรแกรมในการเลือกดูเอกสารในระบบอินเตอร์เน็ตที่เป็นเวิล์ดไวด์เว็บ ซึ่งบราวเซอร์นั้นจะต้องเชื่อมต่ออยู่กับเซิร์ฟเวอร์เพื่อที่จะขอดูไฟล์ข้อมูลข่าวสารต่าง ๆ ในเครือข่าย
- **ทีซีพี / ไอพี (TCP / IP – Transmission Control Protocol / Internet Protocol)** เป็นกลุ่มของโปรโตคอลในการสื่อสาร เพื่อจัดเตรียมเคลื่อนย้ายไฟล์ข้อมูล
- **โปรโตคอล (Protocol)** กฎซึ่งตกลงกันเกี่ยวกับกลวิธีในการจัดรูปแบบ จัดเวลา จัดลำดับ และควบคุมความผิดพลาดของข้อมูลที่จะส่งผ่านระบบเครือข่าย
- **ล็อกอิน (Login)** การที่ผู้ใช้ขอเข้าในระบบหรือเครือข่ายคอมพิวเตอร์
- **เอฟทีพี (FTP – File Transfer Protocol)** โปรแกรมบริการ ใช้ในการเคลื่อนย้ายและคัดลอกแฟ้มข้อมูลระหว่างคอมพิวเตอร์ กับคอมพิวเตอร์อื่น ๆ ที่อยู่ห่างไกลในระบบเครือข่ายอินเตอร์เน็ต
- **ไอพีแอ็ดเดรส (IP address)** หมายเลขประจำเครื่องคอมพิวเตอร์ในระบบอินเตอร์เน็ตที่ใช้สำหรับอ้างอิงถึง

- **ยูอาร์แอล (URL – Uniform Resource Locator)** ระบบอ้างตำแหน่งสำหรับ www โดยจะมีข้อมูลเกี่ยวกับวิธีเข้าถึงข้อมูลเซิร์ฟเวอร์ที่ต้องการใช้และไดเรกทอรี และไฟล์ที่ต้องการจะใช้บนเซิร์ฟเวอร์

- **ยูนิกซ์ (UNIX)** ระบบปฏิบัติการของคอมพิวเตอร์ ที่เป็นแบบทำงานได้หลายงาน (Multi – tasking) และหลายผู้ใช้ (Multi – user) ในเวลาเดียวกัน