

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งาน คอมมอน เกทเวย์ อินเตอร์เฟส(ซีจีไอ)

Common Gateway Interface (CGI) Application



โดย

นาย บรรพต เติศรารังสรรค์

นาย บุญญวัฒน์ บัวเล็ก

อาจารย์ที่ปรึกษา

ผศ.ดร. สมศักดิ์ ชุมช่วย

เลขหม.....
เลขทะเบียน..... 34051
วัน, เดือน, ปี - 1 ต.ค. 2542

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

กิตติกรรมประกาศ

โครงการชิ้นนี้เป็นเพียงส่วนหนึ่งของโครงการทั้งหมดและสำเร็จลงได้ก็เพราะความช่วยเหลือของหลายๆฝ่ายที่ได้ให้แนวคิดแนวทางในการออกแบบตลอดจนแนวทางในการเขียน โปรแกรมและเทคนิคต่าง ๆ ซึ่งก็ต้องขอกราบขอบพระคุณบุคคลดังต่อไปนี้

ดร. สมศักดิ์ ชุมช่วย อาจารย์ที่ปรึกษาที่คอยช่วยแนะนำในแนวคิดกว้าง ๆ และอุปสรรคในการทำโครงการนี้

www.microsoft.com WebSiteที่เป็นแหล่งศึกษา และศึกษาถึงแนวทางในการเขียน โปรแกรม และท้ายสุด ขอขอบคุณเพื่อน ๆ ทุกคนที่คอยถามถึงสารทุกข์สุขดิบและความเป็นไปของโปรเจค และอีกหลาย ๆ คนที่ไม่ได้กล่าวถึงในที่นี้

บรรพต เลิศธำรงศักดิ์ ๑๖

(นายบรรพต เลิศธำรงศักดิ์)

บุญญวัฒน์ บัวเล็ก

(นายบุญญวัฒน์ บัวเล็ก)

ผู้จัดทำ

การประยุกต์ใช้งาน คอมมอน เกทเวย์ อินเทอร์เน็ต(ซีจีไอ)

นายบรรพต เลิศธำรงศักดิ์

นายบุญญวัฒน์ บัวเล็ก

ดร. สมศักดิ์ ชุมช่วย (อาจารย์ที่ปรึกษา)

ภาคการศึกษาที่ 1 ปีการศึกษา 2541

บทคัดย่อ

โครงการนี้เป็นการสร้างระบบ ผู้ให้บริการเว็บ (Web Server) ซึ่งจะจำลองการทำงานของ อินเทอร์เน็ต โดยให้เครื่องคอมพิวเตอร์ 2 เครื่องทำหน้าที่ เป็นผู้ใช้บริการและผู้ให้บริการ (Client/Server) สามารถที่จะติดต่อกันได้โดยใช้กฎเกณฑ์แบบ ทีซีพี/ไอพี (TCP/IP Protocol) จากนั้น จะใช้ภาษา เซชทีเอ็มแอล (HTML) เขียนโฮมเพจ (Home Page) ขึ้นมาแสดงข้อมูลให้ผู้ ใช้บริการ สามารถที่จะดูได้ อีกทั้ง มีการใช้ภาษาสคริปต์ (Script) และ โปรแกรม (program) เขียนเป็น ซีจีไอ (CGI) ขึ้นมาเพื่อใช้ในการติดต่อ ทำงานระหว่างแหล่งข้อมูลกับผู้ใช้บริการ ให้สามารถที่จะติดต่อกันได้ (Interactive) ตามรูปแบบที่ต้องการ สำหรับการทำงานจะเป็นการศึกษารายละเอียดต่างๆที่เป็นส่วนประกอบของ เว็บ เพื่อที่จะสามารถสร้างการทำงานที่ต้องการได้

Common Gateway Interface (CGI) Application

Mr. Bunphote Learntumrongsak

Mr. Bunyawat Bualek

Asst.Prof.Dr. Somsak Chumchuy

2nd Semetor, Education Year 1998

Abstract

This project is the simulatanous Internet by one computer is client and another one is server. It connected as Client/Server. It was communicated by TCP/IP Protocol . At The Server wrote HTML Language for making Home Pages. CGI was used to Interactive Between Client and Server. In all heading must be studies in each details for each heading.

สารบัญ

	หน้าที่
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของภาคการศึกษานี้	1
1.4 การพัฒนา	2
บทที่ 2 ความรู้เบื้องต้น	3
2.1 ระบบเครือข่าย และการ โปรแกรมบนระบบเครือข่าย	3
2.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์ และอินเทอร์เน็ตเวิร์กกิง (Internetworking)	3
2.1.2 รูปแบบของ โปรแกรมบนระบบเครือข่าย (Network Programming Models)	4
2.1.3 โมเดล OSI	4
2.1.4 เอ็นแคปซูลชัน (Encapsulation)	5
2.1.5 ลักษณะของการติดต่อ	6
2.1.6 แอดเดรส (Address)	7
2.2 ความรู้เบื้องต้นเกี่ยวกับ โปรโตคอล TCP/IP	7
2.2.1 มาตรฐานระบบการสื่อสาร	7
2.2.2 เปรียบเทียบระหว่าง โปรโตคอล TCP/ IP และ OSI โมเดล	11
2.2.3 ชุด โปรโตคอล TCP/IP (TCP/IP Protocol Suite)	12

2.3 ผู้ให้บริการเว็บ (WEB SERVER)	14
2.4 วินโดวส์เอ็นที	17
2.4.1 ประวัติการพัฒนากาของ NT	17
2.4.2 สถาปัตยกรรมของ Windows NT	18
2.4.3 ประสิทธิภาพของ Windows NT	19
2.4.4 การรักษาความปลอดภัยภายใต้ Windows NT	20
2.4.5 รัน Windows NT กับระบบปฏิบัติการเครือข่ายอื่น	21
2.5 Internet Information Server (IIS)	22
2.5.1 คุณสมบัติสำคัญ ๆ ของ IIS	22
2.5.2 การรักษาความปลอดภัย	23
2.5.3 ตอนในการจัดทำเว็บไซต์สำหรับอินเทอร์เน็ตและอินทราเน็ต	24
2.6 ตัวกลางเชื่อมต่อใช้งาน โปรแกรมร่วม (COMMON GATEWAY INTERFACE , CGI)	25
2.6.1 หน้าที่ของ ซีจีไอสคริปต์ (CGI Application Script)	26
2.6.2 ตัวแปรรอบด้าน (Environment Variable)	26
2.6.3 การทำงานของซีจีไอ กับการรับและแสดงผล (Input/Output CGI)	27
2.6.4 การส่งข้อมูลด้วยวิธีการ POST และ GET	28
2.6.5 การเลือกรูปแบบในการที่จะเขียนสคริปต์	29
2.6.6 หลักทั่วไปในการเขียน ซีจีไอ	30
2.6.7 การใช้ ฟอรั่ม เพื่อรับข้อมูลจากผู้ให้บริการ (FORM TAG)	31
2.7 ASP	33
2.7.1 พื้นฐาน	37
2.7.2 Content Rotator	40
2.7.3 Banner Rotator	42
2.7.4 การรับข้อมูลจากผู้ใช้	44
2.7.5 การตรวจสอบประเภทรูปของบราวน์เซอร์	50

2.7.6 ไฟล์ Global.asa	53
2.7.7 การเขียนข้อมูลลงไฟล์ text ด้วยภาษา ASP	56
2.7.8 การใช้ ASP เรียกใช้ฐานข้อมูล	57
2.8 ฐานข้อมูล (Database)	58
2.8.1 ฐานข้อมูลคืออะไรและทำงานอย่างไร	58
2.8.2 การสร้างฐานข้อมูล	61
2.8.3 การแสดงหรือการเปลี่ยนแปลงคุณสมบัติทั่วไปของฐานข้อมูล	62
2.8.4 การเปลี่ยนชื่อ การลบ การคัดลอก หรือการย้ายฐานข้อมูล	62
2.8.5 ความรู้และทฤษฎีพื้นฐาน โมเดลเชิงสัมพันธ์ (Relation Model)	63
บทที่ 3 การสร้าง	71
3.1 ระบบสืบค้นรายชื่อนักศึกษา	71
3.1.1 วัตถุประสงค์ของระบบสืบค้นรายชื่อนักศึกษา	71
3.1.2 ส่วนประกอบของระบบสืบค้นรายชื่อนักศึกษา	71
บทที่ 4 การทดลองและผลการทดลอง	82
4.1 เกี่ยวกับโปรแกรมการสร้างระบบผู้ให้บริการ	82
4.2 การทดลอง	82
บทที่ 5 บทวิจารณ์ และสรุป	83
ผลดีของการที่ศึกษาโครงการนี้	84
ภาคผนวก	85
เอกสารอ้างอิง	86

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันนี้ระบบอินเทอร์เน็ตมีบทบาทอย่างมากในการติดต่อสื่อสารจะเห็นได้ว่าโปรแกรมที่ถูกพัฒนาให้สามารถทำงานได้บนระบบ อินเทอร์เน็ต มีปริมาณเพิ่มขึ้นอย่างรวดเร็ว เนื่องจากประโยชน์ที่ได้รับมีมากมาย หลายด้าน เช่นด้านการติดต่อสื่อสาร ด้านการเผยแพร่ความรู้ หรือเป็นแหล่งรวบรวมความรู้ในสาขาต่าง ๆ

เดิมที่การติดต่อสื่อสารบนระบบอินเทอร์เน็ต เราใช้ตัวอักษรในการสื่อสารซึ่งมีความยุ่งยากโดยเฉพาะผู้ที่ไม่สามารถพิมพ์สัมผัสได้ จึงได้มีการพัฒนาสร้างโปรแกรมประยุกต์ขึ้นมาโดยใช้พื้นฐานการติดต่อสื่อสารบนระบบอินเทอร์เน็ต โดยโปรแกรมประยุกต์นี้มีความสามารถได้ใช้การติดต่อสื่อสารโดยใช้เสียงซึ่งนับเป็นความสะดวกสบายอีกระดับหนึ่งในการสื่อสารและด้วยความสามารถของระบบอินเทอร์เน็ต เราสามารถติดต่อสื่อสารจากที่ใด ๆ ก็ได้ที่เชื่อมต่อเข้ากับระบบเครือข่ายอินเทอร์เน็ต

1.2 วัตถุประสงค์

1. สามารถเรียนรู้การทำงานของ โปรโตคอล TCP/IP และประยุกต์ใช้งานได้
2. สามารถเรียนรู้การใช้งานของ CGI (Common Gateway Interface)
3. สามารถเรียนรู้การใช้งานของ ASP (Active Server Page)
4. สามารถเรียนรู้การใช้งานของ Windows NT Server4.0
5. สามารถเรียนรู้การใช้งานของ IIS (Internet Information Server)
6. สามารถเปรียบเทียบข้อดีข้อเสียของ CGI แต่ละชนิด

1.3 ขอบเขตของภาคการศึกษานี้

1. สามารถติดตั้ง Windows NT Server ให้บริการ WWW Server, FTP Serverเชื่อมต่อกับ โปรโตคอล TCP/IP
2. สามารถเขียนโปรแกรม CGI ติดต่อกับ file นำเสนอบน Web Browser ได้
3. เข้าใจวิธีการใช้ CGI ในการติดต่อกับฐานข้อมูลผ่าน ODBC ได้

1.4 แนวทางในการพัฒนา

ใช้ CGI ในการประยุกต์สร้าง

1. Search โปรแกรมค้นหาข้อมูลในฐานข้อมูล
2. Counter ในการนับจำนวนผู้ที่เข้ามาเยี่ยมชมเว็บไซต์ของเรา
3. Guest book เพื่อรับความคิดเห็นของผู้ที่เยี่ยมชมเว็บไซต์
4. Web board เพื่อประกาศข่าวต่าง ๆ บนเว็บไซต์
5. Application อื่น ๆ



บทที่ 2

ความรู้เบื้องต้น

2.1 ระบบเครือข่าย และการโปรแกรมบนระบบเครือข่าย

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควรในหัวข้อนี้จะอธิบายกับศัพท์พื้นฐาน และหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายดังจะกล่าวต่อไป

2.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์ และอินเทอร์เน็ตเวิร์ก

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End-System) ซึ่งระบบปลายทางเป็นระบบที่เป็นอิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นได้ ตั้งแต่ไมโครคอมพิวเตอร์ (Microcomputer) ไปจนกระทั่งซูเปอร์คอมพิวเตอร์ (Supercomputer) ขนาดใหญ่เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งปันทรัพยากรของระบบเช่น ไฟล์ (File) ข้อมูล, เครื่องพิมพ์ (Printer), โมเด็ม (Modem) ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing database) อินเทอร์เน็ตเวิร์ก หรืออินเทอร์เน็ต (Internet) คือการเชื่อมต่อของระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้น คอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่ายอื่นๆ ได้ เช่น เน็ตเวิร์คโทเคนริงค์ (Tokenring network) เชื่อมกับเน็ตเวิร์คอีเทอร์เน็ต (Ethernet network) โดยมีเกตเวย์ (Gateway) เป็นตัวเชื่อม

2.1.2 รูปแบบของโปรแกรมบนระบบเครือข่าย (Network Programming Models)

จากหัวข้อ 2.1.1 ได้ให้ความหมายของระบบเครือข่าย แสดงถึงวิธีการที่ระบบคอมพิวเตอร์ใดๆ จะทำการเชื่อมต่อกันระบบเครือข่าย แต่ว่าคอมพิวเตอร์ที่เชื่อมต่ออยู่กับระบบเครือข่ายนั้นมีวิธีการอย่างไรในการแลกเปลี่ยน ข้อมูลและแบ่งปันทรัพยากรเหล่านั้นได้ทำให้ต้องมีโปรแกรมประยุกต์ ซึ่งสามารถที่จะจัดการในสิ่งที่กล่าวมาแล้ว อย่างเหมาะสม ซึ่งในหัวข้อนี้จะกล่าวถึงรูปแบบของโปรแกรมบนระบบเครือข่าย 2 รูปแบบคือ ไคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing) และการประมวลผลแบบกระจาย (Distributed Computing)

2.1.2.1 การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing)

การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์นี้การประมวลผลของโปรแกรมประยุกต์จะแบ่งออกเป็น 2 ส่วนคือ ส่วนฟรอนต์เอนด์ (front-end) ที่ทำงานบนไคลเอนต์ ส่วนนี้จะทำหน้าที่แสดงผลที่ได้จากการประมวลผล และรับข้อมูลจากผู้ใช้ อีกส่วนหนึ่งคือแบ็คเอนด์ (back-end) ทำงานบนเซิร์ฟเวอร์มีหน้าที่ในการเก็บรวบรวม และจัดการข้อมูลจากฟรอนต์เอนด์ในรูปแบบการประมวลผลไคลเอนต์ เซิร์ฟเวอร์นี้เครื่องเซิร์ฟเวอร์นี้มักจะเป็นเครื่องที่มีความสามารถสูงกว่าเครื่องไคลเอนต์ โดยปกติเครื่องเซิร์ฟเวอร์มักจะเป็นเครื่องเมนเฟรม หรือมินิคอมพิวเตอร์และเครื่องไคลเอนต์มักจะเป็นเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งการติดต่อกันระหว่างส่วนฟรอนต์เอนด์และแบ็คเอนด์ ทำโดยผ่านระบบเครือข่ายซึ่งในทางปฏิบัติแล้วในส่วนของแบ็คเอนด์ที่อยู่บนเซิร์ฟเวอร์ จะเป็นฝ่ายให้บริการแก่งานของไคลเอนต์หลายงานในเวลาเดียวกัน

2.1.2.2 การประมวลผลแบบกระจาย (Distributed Computing)

การประมวลผลของโปรแกรมประยุกต์มีรูปแบบการประมวลผล 2 แบบคือพรีคอลเล็คชัน (precollection) และการประมวลผลแบบขนาน (parallel processing) ดังนี้

1. พรีคอลเล็คชัน เป็นลักษณะการทำงานที่ข้อมูลที่ต้องการในการจัดเก็บ และส่งต่อไปที่ระบบเครือข่ายอยู่ตลอดเวลาอย่างสม่ำเสมอ การทำงานในลักษณะนี้จะเหมาะสมกับงานบางงาน เช่น ต้องการเก็บสถานะของคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายหนึ่งๆ ทุกเครื่อง

2. การประมวลผลแบบขนาน การประมวลผลลักษณะนี้งานใดๆจะถูกประมวลผลด้วยคอมพิวเตอร์หลายๆเครื่องโดยเครื่องคอมพิวเตอร์เหล่านั้นสามารถจะติดต่อกันโดยระบบเครือข่าย เช่น การทำการพัฒนาโปรแกรมประยุกต์ขนาดใหญ่ โดยทีมพัฒนาที่มีผู้พัฒนาหลายคน สามารถลดเวลาของการแปลและการรวมโมดูล (module) ต่างๆเข้าเป็นโปรแกรมเดียวกันโดยการแบ่งงานการแปลโมดูลเหล่านั้นแก่คอมพิวเตอร์ในระบบเครือข่ายทำการแปลในเวลาเดียวกัน

2.1.3 โมเดล OSI

OSI เป็นคำย่อที่มาจากคำว่า Open System Interconnection โดยที่เป็นมาตรฐานที่ถูกเสนอขึ้นโดย International Standards Organization ซึ่งเป็นองค์กรที่จัดตั้งขึ้นมาเพื่อดูแล และส่งเสริมตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรม แบบระบบเปิด (Open System) เพราะมุ่งที่จะให้ระบบคอมพิวเตอร์ในหลายๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโปรโตคอล (protocol) ในการสื่อสารออกเป็น 7

เลเยอร์ (layer) ซึ่งโปรโตคอลคือชุดของกฎหรือข้อตกลงในการติดต่อ ข้อสังเกตโมเดล OSI เป็นเพียงข้อเสนอแนะมิใช่ข้อกำหนดและควรรู้ว่ายังไม่มีระบบการเชื่อมต่อใดที่สร้างเหมือนกับโมเดล OSI จริงๆ

7.	Application Layer
6.	Presentation Layer
5.	Session Layer
4.	Transport Layer
3.	Network Layer
2.	Datalink Layer
1.	Physical Layer

รูปที่ 2.1 OSI โมเดลทั้ง 7 เลเยอร์(layer)

ในหนึ่งชั้นของเลเยอร์ไม่ได้กำหนดว่าจะต้องมีเพียงหนึ่งโปรโตคอลเท่านั้น ที่อยู่ในระดับเลเยอร์เดียวกัน และในทางตรงข้าม ชุดของโปรโตคอลใดๆ อาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกับเป็นข้อกำหนดของ ระบบเครือข่ายเรียกว่าชุดโปรโตคอล (Protocol Suite) เช่น ชุดโปรโตคอล TCP/IP (Transmission Control/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเป็นเลเยอร์ คือกำหนดการติดต่อระหว่างเลเยอร์ ทำได้โดยไม่ต้องคำนึงถึงการเปลี่ยนในเลเยอร์ใดๆ ที่ติดกัน

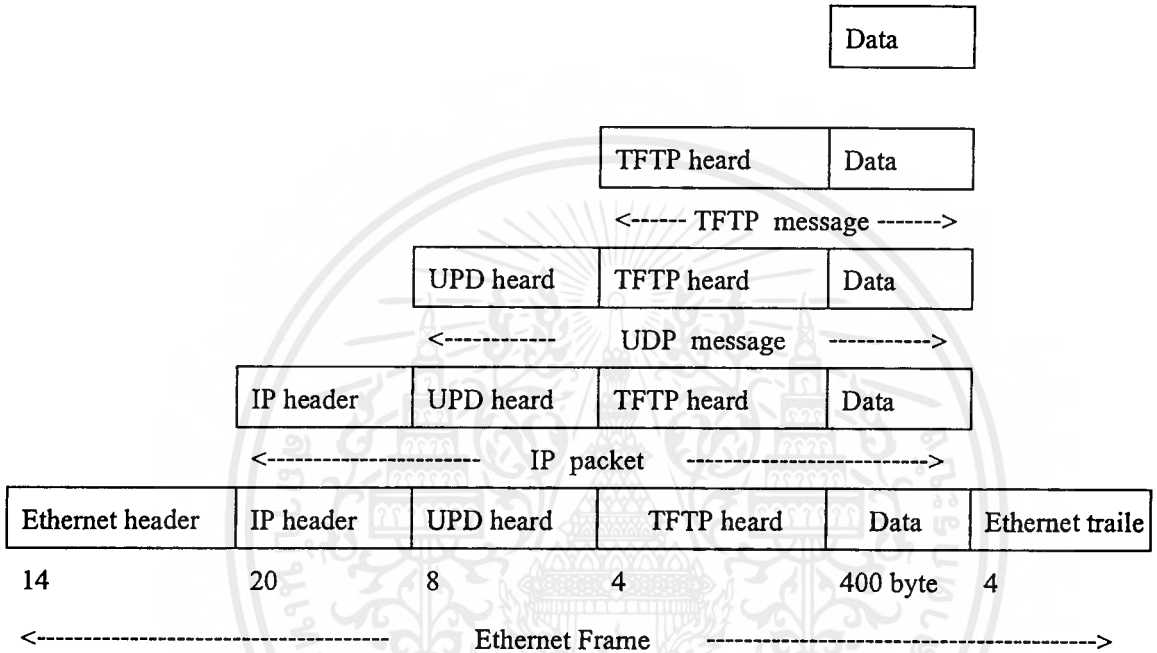
2.1.4 เอ็นแคปซูลชัน (Encapsulation)

พิจารณาโปรแกรมประยุกต์ TFTP (File Transfer Protocol) ซึ่งใช้ในโปรโตคอล UDP (User Datagram Protocol) ระหว่างสองระบบซึ่งเชื่อมต่อด้วยอีเทอร์เน็ต ถ้าโปรแกรมไคลเอนต์ TFTP มีข้อมูล 400 byte ต้องการส่งไปที่โปรแกรมเซิร์ฟเวอร์ โปรแกรมไคลเอนต์ TFTP จะเพิ่มข่าวสารควบคุม 4 byte เป็นส่วนหัวของข้อมูลก่อนที่จะผ่านข้อมูลไปสู่เลเยอร์ UDP การเพิ่มของข่าวสารควบคุมไปที่ข้อมูลเรียกว่า เอ็นแคปซูลชัน ดังแสดงในรูปที่ 2.2

TFTP heard	ata
4 byte	400 byte

รูปที่ 2.2 การเอ็นแคปซูลชัน

เลเยอร์ UDP จะไม่มีการตีความส่วนหัว TFTP 4 byte งานของเลเยอร์ UDP คือส่งข้อมูล 404 byte ไปสู่เลเยอร์ UDP ของโปรแกรมอีกด้านหนึ่งจากนั้นเลเยอร์ UDP จะทำการเพิ่มส่วนหัว 8 byte แล้วส่งข้อมูล 432 byte ไปยังเลเยอร์ค่าต่ำลิ่งที่เลเยอร์นี้จะมีการเพิ่มส่วนหัวอีก 14 byte และส่วนหางอีก 4 byte รูปที่ 2.3



รูปที่ 2.3 ลำดับการเ็นแคปซูลชั้น

2.1.5 ลักษณะของการติดต่อ

แบ่งออกเป็น 2 ชนิด คือ

2.1.5.1 Connection-oriented

คือการติดต่อที่ต้องมีการเชื่อมต่อโปรเซสที่จะทำการติดต่อก่อนที่จะมีการส่งหรือรับข้อมูลซึ่งสามารถใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโปรเซส ถึงแม้ว่าข้อมูลนี้อาจจะผ่าน Packet-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งชั้นการทำงานออกเป็น

1. ชั้นการสร้างการติดต่อ (connection establishment)
2. ชั้นการส่งผ่านข้อมูล (data transfer)
3. ชั้นยกเลิกการติดต่อ (connection termination)

2.1.5.2 Connectionless หรือดาต้าแกรม (Datagram)

คือจะไม่มีชั้นการสร้างการติดต่อและชั้นการยกเลิกการติดต่อแต่จะมีชั้นการส่งผ่านข้อมูลอย่างเดียว โดยข้อมูลเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

2.1.6 แอดเดรส (Address)

การที่ระบบในระบบเครือข่ายสามารถติดต่อกันได้จำเป็นต้องมีแอดเดรสไว้คล้ายกับหมายเลขประจำตัวซึ่งลำดับของแอดเดรสสามารถพิจารณาได้คือ

1. แต่ละเครือข่ายจะต้องมีแอดเดรสสำหรับเครือข่าย
2. คอมพิวเตอร์โฮสต์แต่ละเครื่องในเครือข่ายจะต้องมีแอดเดรส
3. แต่ละโปรเซสในโฮสต์จะต้องมีหมายเลขประจำตัว

โดยทั่วไปแอดเดรสของโฮสต์จะประกอบด้วยหมายเลขเครือข่าย (Network ID) และหมายเลขของโฮสต์ (Host ID) ส่วนแอดเดรสของโปรเซสของผู้ใช้จะอยู่ในรูปจำนวนเต็มซึ่งกำหนดโดยโปรโตคอล เช่น โปรโตคอล TCP/IP จะใช้เลขจำนวนเต็มขนาด 32 บิต ในการกำหนดหมายเลขเครือข่าย และหมายเลขของโฮสต์ และทั้ง TCP และ UDP ใช้เลขจำนวนเต็มขนาด 16 บิต เป็นหมายเลขพอร์ตหรือหมายเลขของโปรเซส ชุดของโปรโตคอลส่วนใหญ่จะมีการกำหนดชุดของแอดเดรสสำหรับการบริการที่เป็นที่รู้จักกันโดยทั่วไป เช่นคอมพิวเตอร์ที่โปรโตคอล TCP/IP ส่วนใหญ่จะมี FTP (File Transfer Protocol) ซึ่งไคลเอนต์สามารถติดต่อได้โดยใช้หมายเลขพอร์ตคือ 21

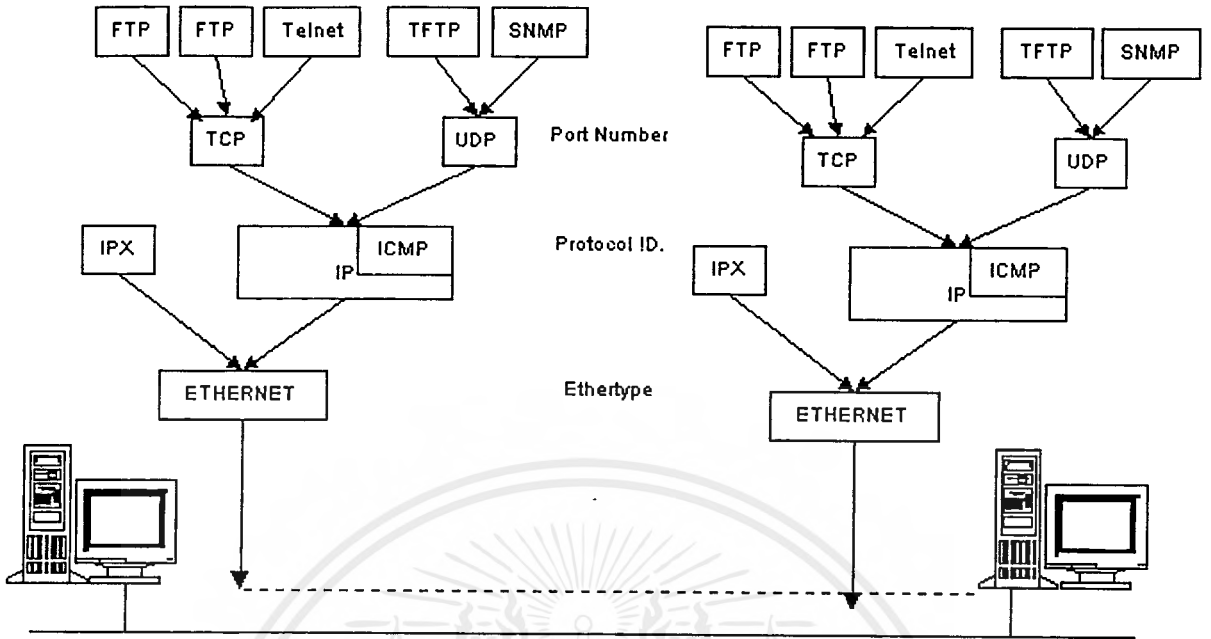
2.2 ความรู้เบื้องต้นเกี่ยวกับโปรโตคอล TCP/IP

2.2.1 มาตรฐานระบบการสื่อสาร

การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ในแต่ละระดับจะติดต่อกันเองในลักษณะที่เรียกว่า Peer-to-peer คือแต่ละ Layer จะมองเฟรมข้อมูลที่มาจาก Layer ข้างบนเป็นข้อมูลและจะนำเอาข้อมูลที่ใช้สำหรับการทำงานใน Layer ของตนเองครอบข้อมูลไว้ที่ส่วนหัว/ส่วนท้าย แล้วส่งต่อไปยัง Layer ข้างล่าง และทำเช่นนี้ไปเรื่อยๆ จนกระทั่งส่งออกไปยังสายการส่ง

OSI	NetWare	UNIX	Apple	LAN Manager
Application	NetWare Core Protocol	Network File System (NFS)	AppleShare	Server Message Block
Presentation			AppleTalk Filing Protocol (AFP)	
Session	Named Pipes NetBIOS	SNMP FTP SMTP Telnet	ASP ADSP ZIP PAP	NetBIOS Named Pipe
Transport	SPX	TCP	ATP NBP AEP RTMP	NetBEUI
Network	IPX	IP	Datagram Delivery Protocol (DDP)	
Data Link	LAN Drivers	LAN Drivers	LAN Drivers	LAN Drivers
	ODI NDIS	Media Access Control	Local- Talk Ether- Talk Token- Talk	NDIS
Physical	Physical	Physical	Physical	Physical

รูปที่ 2.4 แสดงถึงการเปรียบเทียบ โปรโตคอลแบบต่างๆกับมาตรฐานของ ISO



รูปที่ 2.5 แสดงโปรโตคอลย่อยหลายโปรโตคอลซึ่งทำงานในระดับต่างกัน

จากรูปโปรโตคอล TCP/IP ประกอบด้วยโปรโตคอลย่อยหลายโปรโตคอลซึ่งทำงานในระดับต่าง ๆ กัน โดยที่โปรโตคอลย่อยที่ทำงานในระดับ Application Layer ได้แก่ FTP (File Transfer Protocol), TELNET (Terminal Emulator), SNMP (Simple Network Management Protocol) โปรโตคอลเหล่านี้จะเรียกใช้โปรโตคอลย่อยในระดับ Transport ด้านล่างแบบใดแบบหนึ่งระหว่าง TCP (Transmission Control Protocol) หรือ UDP (User Datagram Protocol) และโปรโตคอลย่อยทั้งสองจะเรียกใช้โปรโตคอลระดับ Network ต่อซึ่งได้แก่ IP หรือ ICMP (Internet Communication Message Protocol)

การทำงานในระดับ Transport ที่มีโปรโตคอลย่อย TCP หรือ UDP นั้นโปรแกรมที่อยู่ด้านบนจะเรียกใช้ผ่านช่องทางที่เป็นตัวเลขที่เรียกว่า Port Number โดยหมายเลขนี้เป็นเลขที่มาตรฐานในโปรโตคอลแบบ TCP/IP มาตรฐานของ Port Number ทั้ง TCP และ UDP ได้แสดงไว้ในตารางต่อไปนี้

ตารางที่ 2.1 ตาราง Port Number มาตรฐานในโปรโตคอลแบบ TCP/IP

TCP Prot Number	Application
5	Remote Job Entry
7	Echo
20	FTP Data
21	FTP Control
23	Telnet
25	SMTP
37	Time
53	Domain Name Server
66	Oracle SQL
80	WWW. HTTP
110	Post Office Protocol

UDP Port Number	Application
7	Echo
13	Day Time
37	Time
69	Trival File Tranfer Protocol TFTP
70	Gopher
107	Remote Telnet
161	SNMP (Simple Network Management protocol)

การ Implement โปรโตคอล TCP/IP ของระบบปฏิบัติการต่างๆไม่เหมือนกัน เช่น

1. ในระบบปฏิบัติการยูนิกซ์โปรโตคอล TCP/IP จะเป็นส่วนหนึ่งของระบบปฏิบัติการ
2. ในระบบปฏิบัติการวินโดวส์เอ็นที,VMS,OS/2, Windows จะอยู่ในรูปของ Device Driver

เพื่อให้โปรแกรมเรียกใช้ ซึ่งต้องทำการติดตั้ง Device Driver ก่อน

3. ในระบบปฏิบัติการ MS-DOS มักจะสร้างเป็นโปรแกรมแบบ TSR (Terminate Stay Resident) เมื่อเรียกโปรแกรมแล้วจะฝังตัวอยู่ในหน่วยความจำเพื่อให้โปรแกรมสามารถเรียกใช้งานได้

4. กรณีของระบบปฏิบัติการแบบ Windows โปรโตคอล TCP/IP จะถูกสร้างอยู่ในรูปของ DLL (Dynamic Link Library) เพื่อให้โปรแกรมในวินโดวส์สามารถเรียกใช้งานได้ โดยมีมาตรฐานเรียกว่า WinSock(ย่อมาจาก Window Socket) ซึ่งมีบริษัทต่างๆที่ได้คิดค้นโปรโตคอลแบบ TCP/IP ที่ทำงานในลักษณะนี้หลายๆบริษัท เช่น บริษัทไมโครซอฟต์, บริษัทโนเวล, FTP Software Corp, NetManage Corp รวมทั้งโปรแกรมที่เป็นประเภท Shareware ได้แก่ Trumpet Winsock

2.2.2 เปรียบเทียบระหว่างโปรโตคอล TCP/ IP และ OSI โมเดล

การออกแบบโปรโตคอล TCP/IP นั้นไม่ได้เป็นไปตามรูปแบบของ OSI โมเดล เนื่องจากถูกออกแบบโดยองค์กรขนาดใหญ่ซึ่งใช้เวลานานในการออกแบบตลอดจนการรับรองมาตรฐานต่างกับโปรโตคอล TCP/IP ที่ถูกแบบด้วยความต้องการอันเร่งด่วนทำให้การพัฒนาโปรโตคอล TCP/IP มีเงื่อนไขของในด้านความต้องการที่ต่างจาก OSI โมเดลซึ่งหากเรามองโดยรวมแล้วจะเห็นว่าโปรโตคอล TCP/IP มีการแบ่งเป็นเลเยอร์ที่น้อยกว่า OSI โมเดล คือมี 4 ชั้นเท่านั้น ดังรูปที่ 2.6 โดยแบ่งเป็น

Application Layer
Transport Layer
Internet Layer
Physical Layer

รูปที่ 2.6 เลเยอร์ของโปรโตคอล TCP/IP

1. แอปพลิเคชันเลเยอร์ (Application Layer)

ในเลเยอร์นี้ประกอบโปรแกรมประยุกต์ที่ใช้เครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูล (file-transfer programe) และอาจกล่าวได้ว่าเลเยอร์นี้โปรโตคอล TCP/IP ก็คือ เลเยอร์ในชั้นแอปพลิเคชันเลเยอร์ รวม

กับชั้นพรีเซนเตชันเลเยอร์(Presentation Layer)ใน OSI โมเดลนั่นเองและในเลเยอร์ชั้นนี้ของโปรโตคอล TCP/IP จะกลืนอยู่ในตัวโปรแกรมประยุกต์

2. ทรานสปอร์ตเลเยอร์ (Transport Layer)

ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายถึงจุดปลาย หากเปรียบเทียบกับOSIโมเดล ก็สามารถเทียบได้กับชั้นเซสชันเลเยอร์(Session Layer)ร่วมกับทรานสปอร์ตเลเยอร์นั่นเองโดยโปรโตคอล TCP/IP มีซอกเก็ต (Socket) เป็นจุดปลาย (end-point) ในการสื่อสาร ซึ่งซอกเก็ตนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์และหมายเลขพอร์ต (port) ของเครื่องที่ต้องการส่งข้อมูลไปถึง ในชั้นนี้มีการรับรองการถึงที่หมาย และลำดับของข้อมูลที่ส่งโดยไม่ซ้ำ และความผิดพลาดข้อมูล

3. อินเทอร์เน็ตเลเยอร์ (Internet Layer)

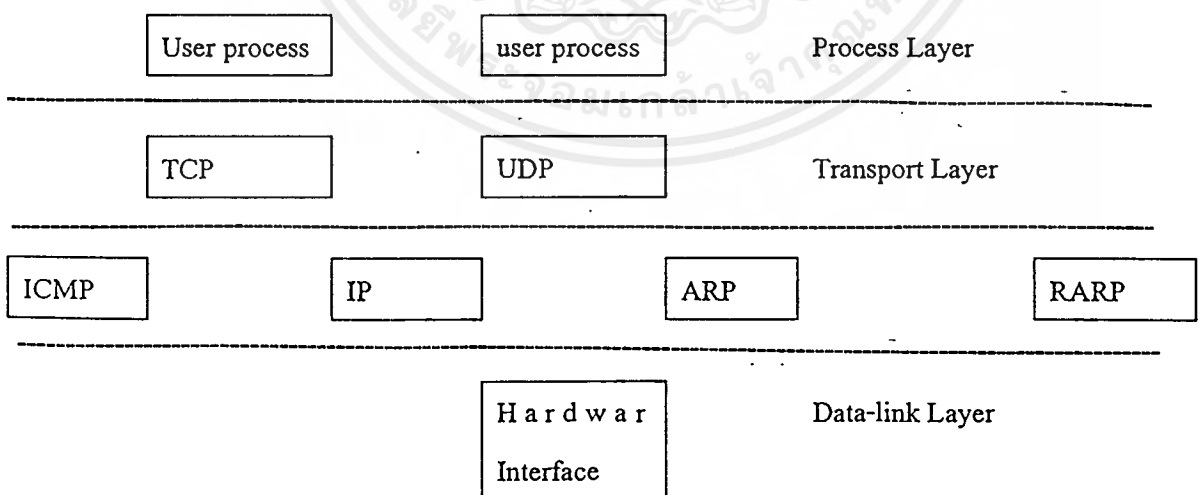
เลเยอร์นี้มีการกำหนดค่าแอดเดรสและทำการหาเส้นทางการส่ง หน้าที่ของเลเยอร์นี้เทียบเท่ากับเน็ตเวิร์กเลเยอร์ (Network Layer) และดาต้าลิงก์เลเยอร์ (Data-link Layer) ของ OSI โมเดล

4. ฟิสิคอลลเยอร์ (Physical Layer)

โปรโตคอล TCP/IP ไม่ได้กำหนดรูปแบบของการเชื่อมต่อในระดับนี้ไว้ใหม่แต่ได้ใช้มาตรฐาน ที่มีอยู่เดิมที่กำหนดไว้ก่อน เช่น RS232, อีเทอร์เน็ต (Ethernet) เป็นต้น

2.2.3 ชุดโปรโตคอล TCP/IP (TCP/IP Protocol Suite)

ชุดโปรโตคอล TCP/IP นอกจากมีโปรโตคอล TCP และ IP แล้วยังมีโปรโตคอลอย่างอื่นอีก ดังภาพที่ 2.7 แสดงความสัมพันธ์ของชุดโปรโตคอลโดยแบ่งตามเลเยอร์



รูปที่ 2.7 ความสัมพันธ์ระหว่างชุดโปรโตคอล

1. โพรโทคอล IP (Internet Protocol)

โพรโทคอล IP เป็นโพรโทคอลแบบคอนเนกชันเลส (Connectionless Protocol) ซึ่งได้กล่าวถึงลักษณะของโพรโทคอล ชนิดนี้ไปแล้วโดยที่โพรโทคอล IP ไม่รับประกันว่าข้อมูลที่ส่งจะไปถึงปลายทางซึ่งแพ็คเกจ (Packet) ของข้อมูลอาจไปถึงในลักษณะที่ผิดพลาด, ช้ากว่า หรือไม่ไปถึงเลย โดยความน่าเชื่อถือของการส่งจะถูกควบคุมในโพรโทคอลในเลเยอร์ต่างๆ ไป การหาเส้นทางของข้อมูลจะทำในระดับของโพรโทคอล IP นี้ โดยพิจารณาแต่ละแพ็คเกจ แยกออกจากกัน และยังมีหน้าที่ในการจัดเรียงข้อมูลใหม่ที่ปลายทาง อีกด้วย

2. โพรโทคอล ARP (Address Resolution Protocol)

โพรโทคอลนี้ทำหน้าที่จับคู่ระหว่างหมายเลข IP แอดเดรสเข้ากับหมายเลขแอดเดรสทางฮาร์ดแวร์ โดยโพรโทคอลนี้ทำการส่งข้อความไปทั่วเครือข่ายท้องถิ่นซึ่งข้อความนี้เป็นลักษณะข้อความที่ตรวจสอบว่ามีคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกับที่ต้องการหาหรือไม่ หากคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกันนั้นได้รับข้อความนี้ก็จะตอบกลับและเป็นที่หน้าสังเกตว่าโพรโทคอลนี้ทำงานได้กับระบบเครือข่ายท้องถิ่นเท่านั้นเพราะว่าโครงสร้างหมายเลขทางฮาร์ดแวร์ของเครื่องคอมพิวเตอร์จะขึ้นอยู่กับชนิดของเครือข่ายด้วย

3. โพรโทคอล ICMP (Internet Control Message Protocol)

เป็นโพรโทคอลที่จัดการเกี่ยวกับข่าวสารความผิดพลาดและการควบคุมเกิดเว็และเรื่องคอมพิวเตอร์ในระบบเครือข่าย

4. โพรโทคอล RARP (Reverse Address Resolution Protocol)

เป็นโพรโทคอลที่ทำหน้าที่จับคู่ระหว่างหมายเลขของฮาร์ดแวร์กับหมายเลข IP แอดเดรส หรือทำงานกลับกันกับโพรโทคอล ARP

5 โพรโทคอล UDP (User Datagram Protocol)

โพรโทคอลนี้เป็นโพรโทคอลที่อยู่ในระดับทรานสปอร์ตเลเยอร์ และมีความสำคัญเพราะว่าเป็นโพรโทคอลที่ผู้พัฒนาโปรแกรมสามารถใช้ได้โดยตรงโพรโทคอลนี้เป็นโพรโทคอลแบบคอนเนกชันเลสมีความน่าเชื่อถือต่ำไม่รับรองว่าข้อมูลที่ส่งไปจะไปถึงปลายทางหรือไม่ และอาจช้าช้อน หรือผิดพลาดได้ แต่ข้อดีของโพรโทคอลนี้ คือ ค่าความสิ้นเปลือง (overhead) ที่ต่ำ

6 โพรโทคอล TCP (Transmission Control Protocol)

โพรโทคอลนี้อยู่ในระดับทรานสปอร์ตเลเยอร์ เหมือนกับโพรโทคอล UDP แต่มีลักษณะที่ตรงข้ามกันคือ เป็นโพรโทคอลแบบคอนเนคชันออเรียนเต็ด โดยจะมีความน่าเชื่อถือในการรับ ส่งข้อมูลและลำดับของข้อมูลจะมีลำดับเหมือนกับต้นทางและเนื้อข้อมูลไม่ผิดพลาดจึงทำให้เกิดความสิ้นเปลืองในการเชื่อมต่อของการส่งข้อมูลมากกว่าโพรโทคอล UDP

2.3 ผู้ให้บริการเว็บ (WEB SERVER)

เวิลด์ไวด์เว็บ มีลักษณะการทำงานแบ่งออกเป็น 2 ส่วน คือส่วนผู้ใช้บริการ และผู้ให้บริการ ในสมัยเริ่มแรกที่มีระบบเครือข่ายการติดต่อ ระหว่างเครื่องที่ต่างลักษณะกัน นั้นทำได้โดยมีกาใช้กติกากในการติดต่อระหว่างเครื่องที่ เรียกว่า โพรโทคอล (Protocol) ซึ่งใช้ในระบบอินเตอร์เน็ตได้แก่ ทีซีพี/ไอพี (Transmission Control Protocol/Internet Protocol, TCP/IP) โดยที่จะใช้ในการส่ง จดหมาย , การโอนย้ายแฟ้มข้อมูล (File Transfer Protocol, FTP) และ โพรโทคอลที่ใช้ในการส่ง แฟ้มข้อมูลและสามารถที่จะดูแฟ้มข้อมูลได้ด้วย คือ เอชทีทีพี (Hypertext Transfer Protocol, HTTP) ก็ใช้กติกากการส่งแบบ ทีซีพี ไอพี นี้เช่นกัน

แนวคิดของเวิลด์ไวด์เว็บ พัฒนาขึ้นที่สำนักวิจัยฟิสิกส์พลังงานชั้นสูง ที่กรุงเจนีวา ประเทศสวิสเซอร์แลนด์ โดย ทิม เบอร์นส์ ลี (Tim Berners Lee) ได้ทำการรวบรวมข้อมูลจากที่ต่างๆ ไว้เป็นหมวดหมู่ และสามารถที่จะค้นหา หรือใช้งานได้สะดวก โดยวิธีที่เรียกว่า ไฮเปอร์เท็กซ์ (Hypertext) และอีกทั้งยังใช้ประโยชน์ในเครือข่ายได้ด้วย วิธีการคือจะจัดเก็บข้อมูลไว้ที่เครื่อง หนึ่งซึ่งทำหน้าที่ ผู้ให้บริการ เรียกว่า เว็บเซิร์ฟเวอร์ (Web Server) ซึ่งข้อมูลจากเครื่องให้บริการแต่ละเครื่องที่เชื่อมโยกันอยู่นี้ เปรียบเสมือนใยแมงมุม (WEB) และเครื่องที่เป็นผู้ใช้บริการ (Client) จะสามารถใช้หรือดูข้อมูลได้โดยอาศัยโปรแกรมที่ทำหน้าที่แสดงผลของเอกสารได้เรียกว่า บราวเซอร์ (Web Browser) ทำให้สามารถที่จะดูเอกสารได้เพียงอย่างเดียว ซึ่งในปัจจุบันได้มีการ พัฒนาให้สามารถที่ดู และโต้ตอบกันได้ผ่านทาง บราวเวอร์ นี้ได้เช่นกัน

สำหรับรายละเอียดของการทำงานและติดตั้ง ผู้ให้บริการเว็บ จะขึ้นอยู่กับเครื่องและโปรแกรมที่ใช้ ซึ่งในส่วนของผู้ให้บริการเว็บนี้ จะมีโปรแกรมที่ทำหน้าที่นี้อยู่หลายแห่ง แต่ละแห่งหรือแต่ละเซิร์ฟเวอร์ จะเป็นโปรแกรมที่มีทั้งข้อดีข้อเสียต่างกันไป แต่โดยรวมทั้งหมดจะติดตั้ง ใช้งานสะดวก การติดตั้ง เว็บเซิร์ฟเวอร์ สามารถทำได้ในหลายๆรูปแบบ ถ้าเป็นในอินเตอร์เน็ต จะเป็นการใช้บนยูนิคซ์ หรือวินโดวส์เอ็นที ก็ได้ สำหรับแบบอินทราเน็ต จะขึ้นกับระบบปฏิบัติการ ที่ใช้มักจะใช้ วินโดวส์เอ็นที หรือ วินโดวส์ 95 ซึ่งนิยมใช้งานกันในปัจจุบัน โดยเฉพาะระบบปฏิบัติการ วินโดวส์ 95 จัดว่าเป็นระบบที่ใช้กันมากในประเทศไทย เนื่องจากได้ พัฒนาให้สามารถที่จะทำงานในลักษณะของเครือข่ายได้ดี สามารถที่จะทำเป็น เครื่องผู้ให้บริการได้ และทำเป็นเครื่องผู้ใช้บริการก็ได้ เช่น

กัน ดังนั้นการทำให้ Windows NT เป็น เซิร์ฟเวอร์ จะสะดวก และใช้งานง่าย แหล่งที่สามารถนำเอาซอฟต์แวร์ของเว็บเซิร์ฟเวอร์ มาใช้งานได้ง่าย ที่

<http://www.microsoft.com/ntserver/tool/web.html>

<http://www.fas.harvard.edu/~glau/httpd> เป็นต้น

วิธีการกำหนดประเภทแฟ้มข้อมูลที่ส่งจากผู้ให้บริการเว็บนั้นในส่วนของผู้ให้บริการเว็บจะกำหนดจากแฟ้มข้อมูลประเภท เอ็มไอเอ็มอี (mime.conf) ซึ่งเป็นการระบุว่าแฟ้มข้อมูลที่ส่งออกเป็นแฟ้มข้อมูลเกี่ยวกับอะไร มีนามสกุลหรือส่วนขยาย (Extension) อะไร

ตารางที่ 2.2 แสดงประเภท แฟ้มข้อมูล

ประเภท	ชนิด
Audio/x-aiff	aif aiff aifc
Audio/x-wav	wav
Image/gif	gif
Text/html	html
Text/plain	txt
Video/mpeg	mpeg mpg mpe
Video/x-sgi-movie	movie

2.3.1 เว็บเซิร์ฟเวอร์ บนระบบยูนิกซ์

เว็บเซิร์ฟเวอร์ ที่ใช้งานสามารถที่นำมาใช้ได้ฟรี และแบบมีระยะกำหนดเวลา (Shareware) ในระบบยูนิกซ์ จะนิยมใช้ของ เอ็นซีเอสเอ (NCSA) ที่เรียกว่า เอชทีทีพีดี (httpd) หมายถึง โปรแกรม ที่ทำหน้าที่เป็น เดย์มอน (Daemon) คือการทำงานหลายๆงานในเวลาเดียวกัน ของระบบยูนิกซ์ จะอาศัยวิธีการจัดโปรเซส หรือโปรแกรมที่ประมวลผล เมื่อเริ่มทำงานครั้งแรก จะทำการโหลด โปรแกรมที่ทำหน้าที่ คอยรับคำสั่งหรือทำหน้าที่จัดการกับโปรเซสที่เกี่ยวข้องเมื่อเรียกใช้งาน ในกรณีที่ไม่มีเรียกใช้งาน โปรแกรมหรือโปรเซส จะหยุดไม่มีการทำงานอะไร เรียกการทำงาน แบบนี้ว่า เดย์มอน จะคล้ายกับการทำงานแบบ เรสซิเด็นท์ (Resident) ของระบบ ดอส จะสังเกตได้จากโปรแกรมที่ทำงานแบบนี้จะมีนามสกุล ที่ลงท้ายด้วยตัว "d" เช่น เอฟทีพีดี (ftpd), เอชทีทีพีดี (httpd), ไอเอ็นอีทีดี (inetd) เป็นต้น ทำหน้าที่ในการ คอยรับการร้องขอ (Request) จากโปรแกรมที่ใช้ดูข้อมูลเอกสาร จาก

นั่นผู้ให้บริการ ก็จะส่งเพิ่มข้อมูลที่ขอไปให้มา โดยจะมีส่วนเพิ่มเติมที่ส่วนหัวของเอกสาร เพื่อใช้บอกประเภทของเพิ่มข้อมูลที่ส่งมานั้น

ในการทำงานของผู้ให้บริการเว็บนี้ต้องมีการกำหนดส่วนที่ใช้ในการให้บริการ โดยจะเก็บเพิ่มนี้ไว้ที่ ชื่อว่า /etc/services โดยบริการหลักจะถูกกำหนดขึ้นในส่วนคั่นๆ และมีการกำหนดใช้งานเป็นหมายเลข พอร์ต (Port) ที่ให้บริการ สำหรับการทำงานของ เว็บ จะใช้ พอร์ต หมายเลข 80 เสมอ ถ้าเป็นการทำงานแบบ พรอกซี่ (Proxy) คือ การทำงานที่เป็นลักษณะ แคช (Cache) ของ เว็บเพจ ซึ่งเมื่อมีการใช้งานมากๆ จะเก็บค่าของ เว็บเพจ นั้นไว้ เพื่อให้สามารถ เรียกดูได้รวดเร็วยิ่งขึ้น จะใช้ พอร์ต หมายเลข 8080 และยังเป็นภาระลดความหนาแน่นของการ คิดต่อได้

การตรวจสอบเมื่อติดตั้ง เว็บเซิร์ฟเวอร์ แล้ว ทดลองการทำงานโดยเขียน โฮมเพจ ที่ต้องการขึ้นมาใช้งาน แล้วใส่ไว้ที่ ไดรกทอรีของเอกสาร ในเว็บเซิร์ฟเวอร์ จากนั้นใช้โปรแกรมที่ใช้ดู เอกสาร (Web Browser) เรียกดูในไดเรกทอรี ที่เก็บเอกสารไว้ ถ้าสามารถดูได้ แสดงว่าโปรแกรม งานปกติ แต่ ถ้ามีข้อผิดพลาดแจ้งให้ทราบว่าจะหาข้อมูลไม่เจอให้ลองตรวจสอบระบบเครือข่ายว่าปกติหรือไม่หรือตรวจสอบว่าป้อนชื่อข้อมูลถูกต้องหรือไม่ควรสังเกตว่านามสกุลของเพิ่มจะเป็น เอชทีเอ็มแอล (html) เสมอ ถ้าเป็นบนระบบวินโดว์ จะเป็นนามสกุล เอชทีเอ็ม(htm)เท่านั้น

2.3.2 เว็บเซิร์ฟเวอร์ บนระบบวินโดว์

สามารถที่จะนำโปรแกรมมาใช้งานโดยใช้ระบบอินเตอร์เน็ตจากการดาวน์โหลด (Download) จากแหล่งที่มีบริการไว้ให้ โดยจะมีทั้งชนิดต้องซื้อ (Commercial) หรือให้ใช้ ฟรี (Freeware) หรือเป็นแบบให้ทดลองใช้ดูก่อน (Shareware) โดยขึ้นกับความต้องการ อีกทั้งต้อง คุรรายละเอียดให้คิดว่า เป็นการใช้งานกับ วินโดว์ 95 หรือ วินโดว์เอ็นที หรือบางโปรแกรมสามารถ ใช้งานได้ทั้ง วินโดว์95 และวินโดว์เอ็นที โดยปกติการติดตั้งของระบบวินโดว์จะ ทำได้ง่าย เนื่องจาก ถ้าเป็นระบบเครือข่าย มักจะมีขนาด จำนวนผู้ใช้งานไม่มาก ต้องมีการติดตั้งระบบ เครือข่ายไว้ก่อนแล้ว อีกทั้งยังต้องติดตั้ง โปโตรคอล ทีซีพี ไอพี ไว้ด้วยแล้ว

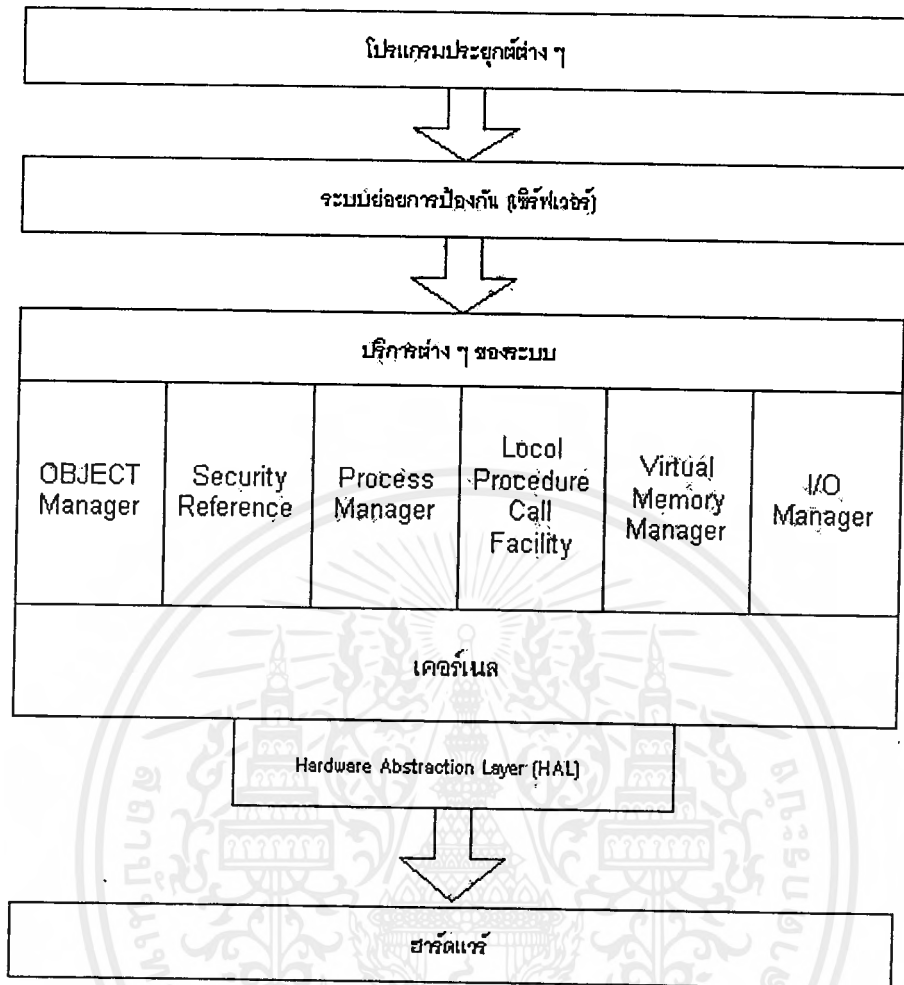
นอกจากนี้การใช้งานเว็บเซิร์ฟเวอร์ในปัจจุบันนิยมใช้งานในระบบ Windows95 และ NT มากขึ้น โดยจะทำการเป็นระบบ อินทราเน็ต กันมาก ดังนั้นถ้าได้ศึกษาการทำงานเกี่ยวกับ วินโดว์ 95 ในระบบ อินเตอร์เน็ต จะทำให้มีความรู้ ที่จะพัฒนาการทำงานเกี่ยวกับ เว็บเพจ หรือ โฮมเพจ ทำได้สะดวก มีเครื่องมือช่วยในการทำงานมากมาย

ดังนั้น การใช้งานและการติดตั้ง เว็บเซิร์ฟเวอร์ ในระบบวินโดว์จะทำได้สะดวกกว่าระบบยูนิกซ์ แต่มีการทำงานและใช้งานมากจะใช้ระบบยูนิกซ์ ในการสร้างระบบ เว็บเซิร์ฟเวอร์ จะต้องการดูแลเกี่ยวกับ การติดตั้งการจัดข้อมูลของเอกสารที่น่าสนใจ การจัดหมวดหมู่ ของเอกสาร และวิธีการที่จะทำให้ เว็บเพจที่สร้างขึ้นมา มีความน่าสนใจ มีผู้ใช้บริการเข้าดู เว็บไซต์ ของเรา

ขาดหน่วยประมวลผลกลางจะอนุญาต ให้มีการทำงานควบคู่กัน ไปทั้งงานเบื้องหน้าและงานเบื้องหลัง โดยไม่ระบุเจาะจง ให้ทำงานใดงานหนึ่งโดยเฉพาะ กล่าวคือโปรแกรมจะจับจองขอทำงานเพียงผู้เดียว ไม่ได้โดยหากมีโปรแกรมอื่นขอจัดจังหวะจะต้องปล่อยให้โปรแกรมนั้นแทรกขึ้นมาทำงานได้ด้วย ความสามารถในการประมวลผลแบบหลายเทรค หมายถึงการกระจายการทำงานหรือการประมวลผลออกเป็นหลาย ๆ เส้นทาง แล้วดำเนินการควบคู่กัน ไปให้เสร็จสิ้นในการบวนการเดียว Windows NT ใช้ระบบไฟล์แบบ NT (NTFS) ซึ่งเป็นระบบไฟล์ที่รองรับการตั้งชื่อไฟล์ได้ยาวถึง 256 อักขระ และมีความสามารถในการติดตามการทำงาน ซึ่งหมายความว่าเมื่อระบบทำงานผิดพลาดกลับลงไปได้ การเชื่อมโยงข้อมูลของ Windows NT สนับสนุนตามข้อกำหนดระดับชั้นการเชื่อมโยงข้อมูลของมาตรฐาน IEEE 802.2 สำหรับระบบเครือข่ายทั้งแบบโทเคนริงและอีเธอร์เน็ต รวมทั้งข้อกำหนดโปรโตคอล SDLC, โปรโตคอล X.25/QLLC และ DFT (distributed functionterminal) นอกเหนือจากนี้ Windows NT รุ่น Advanced Server ยังมีคุณลักษณะเด่นอื่นๆ อีกหลายประการ รวมถึงความสามารถมรการเชื่อมต่อกับระบบปฏิบัติการเครือข่ายอื่นๆ

2.4.2 สถาปัตยกรรมของ Windows NT

ดังแสดงในรูปที่ 2.8 จะเห็นว่า Microsoft ได้ตั้งใจออกแบบ Windows NT ขึ้นมาให้มีลักษณะเป็นโมดูล (modular) และมีความยืดหยุ่นในการโยกย้ายไปมา โดยมีแก่นสำคัญในการรวมระบบย่อยทั้งหลายให้ทำงานร่วมกันได้ที่เคอร์เนล (kernel) ระบบย่อยดังกล่าวสามารถรองรับได้ทั้งโปรแกรมใช้งานของ OS/2 และโปรแกรมที่เป็นไปตาม POSIX และส่วนการจำลองเป็นเครื่อง DOS หรือ VDM (virtual DOS machine) ที่ทำให้สามารถใช้งานโปรแกรมของ DOS และของ Windows แบบ 16 บิตบน Windows NT ได้ เคอร์เนลคือแก่นของ Windows NT ที่คอยรองรับการทำงานพื้นฐานและคอยกำกับความสัมพันธ์ในการทำงานของส่วนประมวลผลทั้งหลาย รวมทั้งการควบคุมการจัดจังหวะในการทำงานและการข้ามความผิดปกติ NT Executive ทำหน้าที่จัดการการเชื่อมต่อระหว่างเคอร์เนลกับระบบย่อยต่าง ๆ ส่วน I/O Manager คอยควบคุมการร้องขอใช้งานทางเข้าและ/หรือออกอย่างเป็นอิสระต่ออุปกรณ์



รูปที่ 2.8 สถาปัตยกรรมของ Windows NT

HAL หรือ Hardware Abstraction Layer ทำหน้าที่แปลงคำสั่งจาก NT Executive ให้อยู่ในรูปแบบที่ Hardware เข้าใจได้ตามแพลตฟอร์มของ Hardware ที่ติดตั้งไว้ ซึ่งผลของการแยกชุดคำสั่งในการควบคุม Hardware ของ Windows NT ออกมาต่างหากอยู่ในรูป HAL นี้เป็นไปตามความต้องการของ Microsoft ที่ตั้งใจออกแบบ Windows NT ขึ้นมาให้เป็นระบบปฏิบัติการเครื่อง่ายที่สามารถโยกย้ายไปใช้กับแพลตฟอร์มต่าง ๆ ได้อย่างหลากหลายผ่านการโปรแกรมด้วยภาษา C

2.4.3 ประสิทธิภาพของ Windows NT

Windows NT เป็นระบบปฏิบัติการเครื่อง่ายแบบ 32 บิตที่สามารถทำงานหลาย ๆ งานควบคู่กันไปได้ และยังสนับสนุนระบบหลายโปรเซสเซอร์ รวมทั้งการประมวลผลแบบสมมาตรที่เป็นการกระจายงาน รวมทั้งการประมวลผลแบบสมมาตรที่เป็นการกระจายงานออกไปยังโปรเซสเซอร์ทั้งหลายให้ช่วยกันทำ การประมวลผลอย่างได้สมดุลซึ่งด้วยคุณลักษณะดังกล่าวนี้ทำให้บริษัททั้งหลายที่ต้องการระบบเครื่อง่าย ที่มีประสิทธิภาพสูงสามารถจัดซื้ออุปกรณ์เซิร์ฟเวอร์ ที่มีไมโครโปรเซสเซอร์

หลาย ๆ ตัวมาใช้งานร่วมกับ Windows NT ได้อย่างมีประสิทธิภาพ ตัวอย่างเช่น เครื่องที่มีไมโครโปรเซสเซอร์ 6 ตัวก็จะมีความเร็วสูงกว่าเครื่องที่มีไมโครโปรเซสเซอร์ตัวเดียวขึ้นไปอีก 6 เท่า เป็นต้น

ภายใน Windows NT มีความสามารถในการทำระบบเครือข่ายแบบเท่าเทียมมาให้ด้วย กล่าวคือผู้ใช้ Windows NT สามารถแบ่งเป็นร่วมใช้ทรัพยากรอื่น ๆ กับผู้ใช้อื่นที่ใช้ Windows NT, Windows, DOS และ OS/2 ได้ด้วยนั่นเอง นอกเหนือจากนี้ใน Windows NT ยังใช้เทคนิคในการจัดการกับหน่วยความจำเป็นแบบแฟลต (flat) แทนที่จะใช้วิธีการแบ่งเพจ (page) อันทำให้การทำงานของ Windows NT มีประสิทธิภาพสูงขึ้น โดยเฉพาะอย่างยิ่งในการใช้งานกับโปรแกรมกราฟิกซึ่งจะทำงานได้เร็วขึ้นมาก

ประสิทธิภาพที่สูงขึ้นของ Windows NT ในความเป็นจริงแล้วเป็นผลพวงมาจากความต้องการทรัพยากรของตัว Windows NT เองที่ระบุเอาไว้ว่าต้องเป็นเครื่องที่ใช้หน่วยประมวลผลกลางเบอร์ 80486DX ของ Intel ขึ้นไป และต้องมีหน่วยความจำแรมอย่างต่ำ 12 เมกะไบต์ และมีพื้นที่ว่างในการติดตั้งบนฮาร์ดดิสก์อีก 100 เมกะไบต์ โดยนอกเหนือจากเครื่องที่ใช้ชิปของ Intel แล้ว Microsoft ยังได้ร่วมมือกับอีกหลาย ๆ บริษัท ซึ่งรวมทั้ง Digital Equipment Corporation และ MIPS Computer Systems เพื่อให้สามารถนำ Windows NT ไปใช้งานบนเครื่องที่มีฮาร์ดแวร์แพลตฟอร์มต่าง ๆ ได้หลายแบบ

2.4.4 การรักษาความปลอดภัยภายใต้ Windows NT

ในการเข้าทำงาน Windows NT นั้น ผู้ใช้จะต้องป้อนรหัสผ่านเพื่อแสดงการขอเข้าใช้ มิฉะนั้น Windows NT จะไม่อนุญาตให้เข้าใช้เซิร์ฟเวอร์ได้ โดยใน Windows NT จะมีโปรแกรม User Manager ที่ใช้ในการจัดการเกี่ยวกับบัญชีผู้ใช้ โดยสามารถแบ่งระดับความสามารถในการเข้าถึงระบบเครือข่ายของผู้ใช้ต่าง ๆ ออกเป็นหลาย ๆ ระดับได้ด้วย นอกเหนือจากนี้ยังสามารถสร้างกลุ่มผู้ใช้และกำหนดสิทธิให้กลุ่มได้กล่าวคือ ผู้ใช้ที่อยู่ในกลุ่มผู้ใดก็จะมีสิทธิในการใช้ระบบเครือข่ายเท่าเทียมกัน

อีกคุณลักษณะเด่นหนึ่งของ Windows NT คือโปรแกรม Event Viewer ซึ่งเป็นโปรแกรมที่ผู้บริหารระบบเครือข่ายใช้ในการตรวจสอบดูสิ่งผิดปกติทั้งหมดที่เกิดขึ้นในระบบเครือข่าย รวมทั้งการฝ่าฝืนต่าง ๆ ซึ่งจะมีข้อมูลแสดงวันที่ เวลา และรูปแบบการฝ่าฝืน รวมทั้งสถานที่ที่ผู้ฝ่าฝืนดำเนินการ และชื่อของผู้ฝ่าฝืน ทุกครั้งที่จะเริ่มใช้งาน Windows NT ผู้ใช้จะต้องป้อนรหัสผ่านก่อนทุกครั้ง ด้วยการกด Ctrl-Alt-Del เหมือนการบูตเครื่องใหม่ เพื่อทำการป้อนชื่อและรหัสผ่านของตน

การรักษาความปลอดภัยภายใน Windows NT เป็นไปตามมาตรฐานการรักษาระดับ C2 ของรัฐบาลสหรัฐอเมริกา นอกเหนือจากนี้ทาง Microsoft ยังได้แจ้งด้วยว่าจะเพิ่มระดับการรักษาความปลอดภัยใน Windows NT ให้สูงกว่านี้ขึ้นไปอีกในอนาคต

2.4.5 รัน Windows NT กับระบบปฏิบัติการเครือข่ายอื่น

Windows NT มีคุณสมบัติของการเชื่อมต่อโปรแกรมใช้งานหรือ API (application programming interface) อันเอื้ออำนวยให้ผู้พัฒนาระบบปฏิบัติการเครือข่ายอื่น ๆ สามารถพัฒนาโปรแกรมไคลเอ็นต์บนผลิตภัณฑ์ของตนติดต่อกับ Windows NT ได้ง่ายขึ้น โดยทาง Novell ได้แจ้งว่าตนจะสามารถปล่อยซอฟต์แวร์ไคลเอ็นต์สำหรับ Windows NT ของตนตามออกมาได้อย่างทันที่ทันใดที่ Windows NT ออกวางจำหน่าย

โปรโตคอลสแต็ค TCP/IP และโปรแกรมใช้งานเป็นไคลเอ็นต์ของ TCP/IP อย่างเช่น Telnet และ FTP มีให้มาด้วยแล้วใน Windows NT นอกจากนี้ยังสนับสนุนการเรอ้า TCP/IP อีกด้วย แต่เนื่องจาก Microsoft มิได้ให้โปรแกรมเซิร์ฟเวอร์สำหรับ Telnet และ FTP มาด้วย ดังนั้นสถานีงานของ UNIX จึงจะยังไม่สามารถเข้ามาใช้ Windows NT โดยตรงได้ในตอนนี้ แต่ทาง Microsoft ได้แถลงว่าจะเพิ่มลงไปด้วยใน Windows NT ยังสนับสนุนโปรโตคอลการจัดการการทำงานของเซิร์ฟเวอร์ Windows NT ด้วยโปรแกรมการจัดการระบบเครือข่าย SNMP ได้

Windows NT สามารถโต้ตอบและแลกเปลี่ยนข้อมูลกับ NetWare Vines และ Sun NFS ได้ ทั้งที่แรกเริ่มที่ออกแบบขึ้นมา นั้น ตั้งใจจะให้รองรับเฉพาะ LAN Manager แต่ก็ปรากฏว่าสามารถรองรับระบบเครือข่ายอื่น ๆ ได้ด้วย ขึ้นอยู่กับว่าระบบเครือข่ายนั้น ๆ รองรับการ์ดเชื่อมต่อระบบเครือข่ายตามข้อกำหนดการเชื่อมต่อส่วนขับเคลื่อนเครือข่าย NDIS (network driver interface specification) ของ Microsoft หรือไม่

การเชื่อมต่อ Windows NT เข้ากับโลกภายนอก Windows NT มีส่วนบริการการทำเซิร์ฟเวอร์การเข้าใช้ในระยะไกลที่เรียกว่า Remote Access Server มาด้วย ซึ่งเซิร์ฟเวอร์ดังกล่าวจะมีความสามารถในการให้ไคลเอ็นต์ระยะไกลที่ใช้อยู่บน DOS, Windows หรือ Windows NT สามารถหมุนโทรศัพท์เข้าติดต่อกับระบบเครือข่าย Windows NT และทำการลงบันทึกเข้า (log in) ใช้งานระบบเครือข่ายได้ประหนึ่งเชื่อมต่อระบบเครือข่ายโดยตรง จะต่างอยู่บ้างก็ตรงความเร็วซึ่งค่อนข้างช้ามาก โดยเซิร์ฟเวอร์ดังกล่าวนี้สามารถรองรับการเชื่อมต่อได้มากถึง 64 การเชื่อมต่อ และมีการรักษาความปลอดภัยตามพื้นฐานการรักษาความปลอดภัยของ Windows NT เอง ซึ่งรวมถึงการใส่รหัสผ่านที่มีการเข้ารหัสข้อมูลตามมาตรฐาน DES (data encryption standard) และความสามารถในการเรียกกลับ (call-back) เพื่อป้องกันบุคคลไม่พึงประสงค์แอบแฝงเข้ามา โดยในการติดต่อสื่อสารระยะไกลนี้ Windows NT รองรับทั้งการใช้โปรโตคอล X.25 และ ISDN (integrated service digital network)

ความสามารถทางอินเทอร์เน็ตของ Windows NT ในปัจจุบันอินเทอร์เน็ตได้เข้ามามีบทบาทกับชีวิตประจำวันเป็นอย่างมาก องค์กรต่าง ๆ จึงจำเป็นต้องเชื่อมต่อเครือข่ายของตนเข้ากับอินเทอร์เน็ต

เน็ต เพื่อให้สามารถแลกเปลี่ยนข้อมูลข่าวสารกับโลกภายนอกได้ เนื่องจากเทคโนโลยีทางด้านอินเทอร์เน็ตก้าวหน้าอย่างรวดเร็ว หลาย ๆ องค์กรในปัจจุบันจึงสร้างระบบ อินทราเน็ต (Intranet) ขึ้นมาในองค์กรเป็นเหมือนกับเครือข่ายอินเทอร์เน็ตแต่จำกัดการใช้งานอยู่เพียงแคภายในองค์กรเท่านั้น

Windows NT 4.0 ได้ถูกออกแบบให้รับงานด้านอินเทอร์เน็ตและอินทราเน็ตหลายประการดังนี้

1. Internet Information Server (IIS)

โปรแกรมนี้สามารถทำให้ Windows NT 4.0 กลายเป็น WWW server, FTP server และ Gopher server เพื่อเผยแพร่ข้อมูลได้ทันที

2. Domain Name Server (DNS)

โปรแกรมนี้จะช่วยควบคุมและแปลง IP address ซึ่งเป็นตัวเลขที่จำได้ยากให้กลายเป็นชื่อที่จำง่ายขึ้น

3. Microsoft Frontpage

Microsoft Frontpage คือ เครื่องมือช่วยสร้างเวปเพจที่มีประสิทธิภาพสูง โดยผู้สร้างเวปเพจไม่จำเป็นต้องมีความรู้ด้าน HTML ก็สามารถสร้างเวปเพจได้อย่างสวยงาม

4. Index server

ช่วยจัดทำเครื่องมือค้นหา (Search Engine) ในเวปซึ่งสามารถค้นหาได้ทั้งเวปเพจและเอกสารที่สร้างโดยโปรแกรมในชุดไมโครซอฟท์ออฟฟิศ นอกจากนี้ที่กล่าวถึงนี้ บริษัทไมโครซอฟท์ยังมีผลิตภัณฑ์ทางด้านอินเทอร์เน็ตอีกมาก ซึ่งคุณสามารถค้นหาข้อมูลเพิ่มเติมหรือ ดาวน์โหลดซอฟต์แวร์ด้านอินเทอร์เน็ตได้ฟรี ที่ <http://www.microsoft.com/msdownload>

2.5 Internet Information Server (IIS)

Internet Information Server หรือที่เรียกย่อ ๆ ว่า IIS เป็นซอฟต์แวร์ที่จะทำให้ Windows NT Server ของคุณกลายเป็นอินเทอร์เน็ตเซิร์ฟเวอร์ ซึ่งคุณสามารถจัดทำ Web site, FTP site หรือ Gopher site ได้โดยง่าย

ซอฟต์แวร์ IIS มีแถมมาในซีดีติดตั้ง Windows NT 4.0 Server คุณสามารถนำไปใช้งานได้โดยไม่ต้องจัดซื้อซอฟต์แวร์อื่น ๆ เพิ่ม

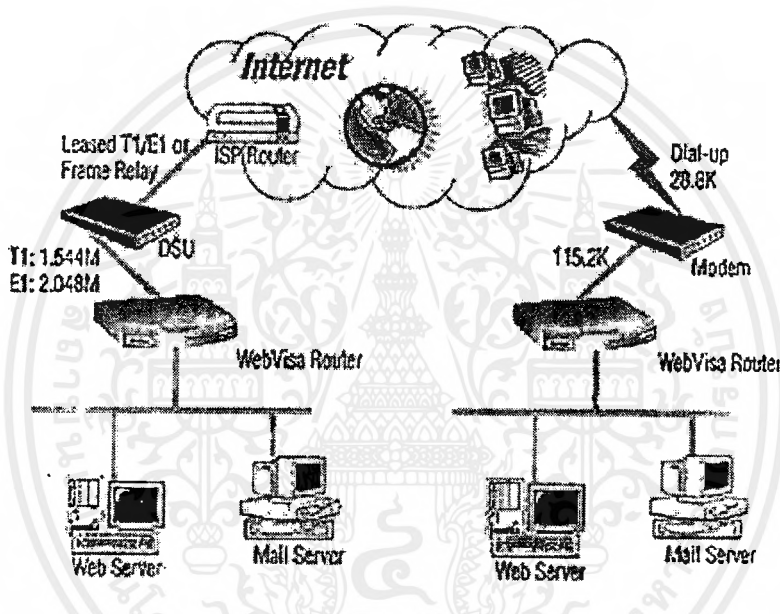
2.5.1 คุณสมบัติของ IIS

1. เป็นเว็บเซิร์ฟเวอร์ (WWW Server) ซึ่งใช้เก็บเวปเพจ ผู้ที่เข้ามาเยี่ยมชมจะสามารถใช้ IP address (เช่น 177.78.21.10) หรือ URL (เช่น WWW.ntbook.com) เพื่อเข้ามาเยี่ยมชมเวปเพจผ่านทางบราวเซอร์ได้โดยผ่านโปรโตคอล HTTP

2. เป็น FTP server ให้บริการไฟล์ช่วยให้ผู้ที่เข้ามาเยี่ยมชมเว็บไซต์สามารถดาวน์โหลดไฟล์ได้โดยตรง โดยผ่านทางโปรโตคอล FTP

3. เป็น Gopher server ให้บริการข้อมูลช่วยให้ผู้ที่เข้าเยี่ยมชมสามารถค้นหาไฟล์ที่ต้องการได้ง่ายขึ้นกว่า การเข้าค้นหาใน FTP โดยค้นหาผ่านเมนูของ Gopher แทน

สร้างอินเทอร์เน็ตไม่จำเป็นจะต้องเชื่อมต่อ Windows NT Server เข้ากับเครือข่ายอินเทอร์เน็ต แต่ถ้าสร้างเว็บไซต์ในอินเทอร์เน็ต ต้องเชื่อมต่อ Windows NT เข้ากับอินเทอร์เน็ตก็ ต้องเชื่อมต่อเข้ากับอินเทอร์เน็ตโดยใช้สายสัญญาณแบบต่าง ๆ เช่น สายโทรศัพท์ และสาย Leased Line



รูปที่ 2.9 แสดงการเชื่อมต่อผ่าน Internet

2.5.2 การรักษาความปลอดภัย

เมื่อนำเครือข่ายของอินเทอร์เน็ต สิ่งที่ต้องคำนึงถึง ก็คือความปลอดภัยเนื่องจาก อาจมีผู้ไม่ปรารถนาดีพยายามจะเจาะระบบผ่านทางอินเทอร์เน็ตซึ่งเปิดกว้างให้คนจากทั่วโลกเข้ามายังเซิร์ฟเวอร์ได้ตลอดเวลา จึงควรคำนึงถึงความปลอดภัยเพื่อป้องกันเหตุการณ์เหล่านั้นไว้ก่อน

การรักษาความปลอดภัยที่สำคัญควรเริ่มต้นที่ชื่อผู้ใช้ของผู้บริหารระบบเอง ควรมึรหัสผ่าน (Password) ที่มีความลับไม่ควรบอกผู้อื่นให้ทราบและคุณเองควรตั้งรหัสผ่านด้วยคำที่ไม่มีมีความหมายเพื่อให้ผู้อื่นเดาได้ยาก และเปลี่ยนรหัสผ่านของ Administrator บ่อย ๆ อย่างน้อยเดือนละครั้ง

ถ้ามีสายโทรศัพท์ซึ่งยอม ให้มีการ Login จากภายนอก ควรเข้ารหัส (Encrypt) การติดต่อผ่านสายโทรศัพท์กับเซิร์ฟเวอร์ทั้งหมด เพราะอาจมีผู้ดักเอาสัญญาณเหล่านั้น ไปใช้ได้

ไม่ควรใช้ Guest หรือ Anonymous เป็นชื่อผู้ใช้ ควรลบออกและใส่ชื่อผู้ใช้ที่มีผู้ใช้งานจริงลงไปแทนรหัสผ่านที่ใช้ต้องเป็นรหัสผ่านที่คาดเดายากและจะต้องกำชับ ให้ผู้ใช้ทุกคนรักษารหัสผ่านของตนเองเป็นความลับหรืออาจเสนอผู้บริหารขององค์กรก็จะเป็นการดีที่สุด

เป็นที่น่าแปลกใจว่าความเล็ดลอดส่วนใหญ่มักเกิดจากการเปิดเซิร์ฟเวอร์ทิ้งไว้โดยไม่มีผู้ดูแลทำให้มีผู้ที่ไม่ประสงค์ดีเข้าสู่ระบบได้อย่างง่ายดายคุณจึงควรใส่รหัสผ่านหน้าจอของ Windows NT Server ไว้ใน Screen Saver และควร Logout ทุกครั้งเมื่อใช้งานเสร็จ

ให้ความรู้และความตระหนักกับผู้ใช้ในองค์กรเกี่ยวกับความปลอดภัย เพราะถึงแม้ว่าวันนี้จะยังไม่มีข้อมูลสำคัญในเครือข่ายแต่การปลูกฝังนิสัยที่ดีให้กับผู้ที่ย่อมจะช่วยให้งานยุ่งยากน้อยลง

2.5.3 ขั้นตอนในการจัดทำเว็บไซต์สำหรับอินเทอร์เน็ตและอินทราเน็ต

ก่อนการจัดทำเว็บไซต์สำหรับอินเทอร์เน็ตในองค์กรควรที่จะปรึกษา ISP (Internet service Provider) เพื่อขอคำแนะนำ เพราะทาง ISP จะเป็นผู้ที่วิเคราะห์ความเหมาะสมให้คุณได้ดีที่สุด

ถ้าต้องการที่ทำเว็บไซต์แต่ไม่ต้องการที่จะติดตั้งเซิร์ฟเวอร์ในองค์กรอาจจะขอเช่าพื้นที่ของ ISP ในเมืองไทยหรืออาจจะขอเช่า ISP ในต่างประเทศก็ได้

ถ้ามีเว็บไซต์อยู่แล้วหรือกำลังจะจัดทำระบบอินทราเน็ต สามารถศึกษาเพื่อที่ทำการติดตั้ง IIS (Internet Information Server) และจัดทำเว็บไซต์และเว็บเพจด้วยตนเอง

หลังจากการติดตั้ง IIS เสร็จเรียบร้อย สามารถจัดทำเว็บเพจได้ทันทีโดยอาจจะเลือกใช้เครื่องมือ Microsoft Frontpage ซึ่งเป็นเครื่องมือที่สามารถให้คุณสร้างเว็บเพจโดยที่ไม่จำเป็นต้องมีความรู้ในภาษา HTML หรือถ้าคุณต้องการให้ User ทั่วไปเป็นผู้ที่สร้างเว็บเพจ คุณอาจจะเสนอให้ User ขอใช้โปรแกรมในชุด Microsoft Office 97 ซึ่งสามารถนำเอกสารที่สร้างใน Word, Excel และ PowerPoint มาแปลงเป็น HTML ได้อย่างง่ายดาย

2.6 ตัวกลางเชื่อมต่อใช้งานโปรแกรมร่วม (Common Gateway Interface :CGI)

เป็นส่วนที่กำหนดรูปแบบการติดต่อระหว่าง ผู้ให้บริการเว็บ (Web Server) กับโปรแกรมภายนอก (Application) โดยรับรายการจากเพิ่มข้อมูล ของ เว็บเพจ (Web Page) โดยที่มี เกตเวย์ (Gateway) ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อ

เกตเวย์ สามารถที่จะออกแบบและใช้งานได้หลายประโยชน์ โดยทั่วไปจะใช้ ป้าย (TAG) <ISINDEX> และ ป้าย <FORM> ใน HTTP เป็นรูปแบบในการทำ คิวรี่(Query) รับ-ส่งข้อมูลจาก ผู้ใช้บริการ กับ ผู้ให้บริการโดยมี ซีจีไอ (CGI) เป็นตัวกลางทำหน้าที่ในการรับ-ส่ง ข้อมูลให้ ซึ่งจะใช้ ซีจีไอ เมื่อ

1. ใช้ในการป้อนข้อมูลที่ ผู้ใช้บริการ คอบคำถามกลับมายัง ผู้ให้บริการ โดยใช้ ฟอรัมของ เซชทีเอ็มแอล (HTML FORM)
2. ใช้เปลี่ยนเอกสารที่มีในระบบที่เปลี่ยนแปลง (Dynamic) ได้ให้เป็น เซชทีเอ็มแอล เพื่อให้ง่ายต่อการค้นหาใน เว็บเบส (Web-Base)
3. ใช้เป็นการตั้งคำถาม ที่ต้องการใช้ฐานข้อมูล และให้ผลเป็นเอกสารของ เซชทีเอ็มแอล

เนื่องจากการบริการพื้นฐานของ เวิลด์ไวด์เว็บมีการเรียกขอรายการ (Request) จากเครื่องที่ให้ บริการ ซึ่งมีการจัดเก็บโดยแยกกันเป็นส่วนๆ ภายใต้ ไคเร็กทอรี (Directory) ใน เครื่องผู้ให้บริการ โดยความสัมพันธ์ระหว่างแฟ้มข้อมูลจะอาศัยตัวเชื่อม(Link) ระหว่างเอกสาร เท่านั้น บางครั้งต้องการ เป็นงานในลักษณะสรุปเรื่องใดเรื่องหนึ่ง ซึ่งจะได้จากระบบฐานข้อมูล หรือ ต้องการใช้ เซชทีเอ็มแอล เป็นส่วนที่ติดต่อกับผู้ใช้โดยตรง โดยการ ใช้ ซีจีไอ นี้ ซึ่งจะรับการทำงาน ของ ผู้ให้บริการ เว็บ การพัฒนางานให้มีความสามารถดีขึ้น ได้, แก้ไขรายการ โดยไม่มีผลกระทบ ต่อฐานข้อมูล, ผู้ใช้ ไม่ต้องเรียนรู้ถึงการใช้งานใหม่กับเครื่องต่างระบบกันได้ การเขียน โปรแกรมใช้งานซีจีไอ (CGI Application) จะใช้ภาษาที่ลักษณะเป็นภาษาสคริปต์ ที่มีโครงสร้าง ไม่ซับซ้อน เช่นในระบบ ยูนิกซ์ (Unix) อาจใช้ภาษา เพิร์ล (Perl) , ทีซีแอลทีเค (Tcl/tk), แพนทอม (Python) และ ยูนิกซ์ สคริปต์เชลล์ (Unix Shell Script) ก็ได้ ในระบบ ดอส (Dos) อาจใช้ภาษา ซี (C) , ซีพลัสพลัส (C++), ปาสคาล (Pascal) , วิซวลเบสิค (Visual Basic) ก็ได้ หรืออาจกล่าวว่าการเขียน โปรแกรมใช้งานซีจีไอ สามารถที่จะเขียนด้วยภาษาใดก็ได้ ที่มี ส่วนติดต่อการทำงานกับ ภายในภายนอกที่เป็นมาตรฐานชนิดเดียวกัน (Standard Input/Output) เป็น การทำงานในการส่ง ค่า ข้อมูลได้ และขึ้นกับ การทำงานของผู้ ให้บริการที่ใช้งานในขณะนั้นๆ และภาษาที่นำมาใช้ ควรเป็น ภาษาที่สามารถใช้ประมวลผลข้อความ และจัดการกับแฟ้มข้อมูลได้สะดวกอีกทั้งมีมาตรฐานความ ปลอดภัยของข้อมูลพอสมควร ในระบบ ที่ทำงานด้วย ผู้ให้บริการเซชทีทีพี (HTTP Server) มี โปรแกรมทำงานที่ใช้ ซีจีไอ ในการทำงานนั้น ด้าน ผู้ใช้บริการ จะต้องมีการขอใช้งาน มายัง ผู้ให้บริการ และ เซชทีทีพี จะได้รับเป็นข้อมูลประกอบ ด้วย

1. ที่อยู่ของเอกสาร (Uniform Resource Identifier, URI)
2. วิธีการรับข้อมูล
3. ข้อมูลที่สำคัญต่างๆ ที่ต้องการ โดยใช้ ซีจีไอ ช่วย

2.6.1 หน้าที่ของ ซีจีไอ (CGI Application)

ตามปกติแล้ว ผู้ให้บริการเว็บ ได้ค่าที่อยู่ของเอกสาร จากการกดเลือกรายการที่มีหรือใส่ที่ อยู่ในรายการ จากนั้น ผู้ให้บริการเว็บ จะทำการติดต่อไปยัง ผู้ให้บริการเซชทีทีพี ตามที่อยู่ ของเอกสาร ที่ระบุไว้ นั้น เมื่อ Server ได้รับการขอเพิ่มข้อมูลเข้ามา ก็จะทำการส่งเพิ่มข้อมูล นั้นกลับไปให้ บาง

ครั้งในการขอเพิ่มข้อมูลอาจจะต้องมีการตรวจสอบผู้ใช้ก่อน โปรแกรมนี้จะใช้ ซีจีไอ ทำหน้าที่นี้ โดยใน ผู้ให้บริการเว็บ จะสามารถดูได้จาก ไคลเรททอรี ที่ชื่อ ซีจีไอ บีไอเอ็น (/cgi-bin/) นอกจากนี้ยังสามารถใช้ ซีจีไอ นี้ในการ นับ (Count) จำนวนผู้ที่เข้ามาดู เว็บเพจ (Web Page) ของ ผู้ให้บริการเว็บ ที่ให้บริการอยู่ ส่วนที่ติดต่อกับผู้ใช้งานเว็บ จะทำหน้าที่เพียงรับรายการ จากผู้ใช้เท่านั้น แต่จะมีการประมวลผลก็ต่อเมื่อ ส่งมาถึงเครื่อง ผู้ให้บริการเว็บ แล้วเท่านั้น ซึ่งการทำงานแบบ ผู้ให้บริการและผู้ให้บริการนี้ส่วนที่ติดต่อกับผู้ใช้คือ โปรแกรมที่ใช้ดูเว็บ (Web Browser) จะทำหน้าที่ รับรายการจากผู้ใช้ แล้วส่งต่อไปให้ ผู้ให้บริการเว็บ เมื่อผู้ให้บริการเว็บได้รับข้อมูลเป็น อินพุท (Input) ในรูป การร้องขอ เอชทีทีพี (HTTP Request) ก็จะไปทำการเรียก โปรแกรม ซีจีไอ โดยจะส่งค่าตัวแปร (Parameter) ผ่านค่อให้กับ ซีจีไอ นี้ ซึ่งจะทำการแปล ความหมาย (parse) ข้อมูลที่รับเข้ามา เช่น ถ้าเป็นการขอ อนุญาตเข้าใช้งานใน ไคลเรททอรี ที่ระบุ ผู้ใช้งานไว้ จะมีการส่งค่าที่เป็น ชื่อผู้ใช้ (User name) และ รหัสผ่าน (Password) ให้ผู้ขอใช้ป้อนกลับ มาแล้วทำการตรวจสอบว่าถูกต้องหรือไม่ ถ้าใช่ ก็จะส่งข้อมูลที่ต้องการให้หรือถ้าเป็นการสอบถาม รายการ ค่าตัวแปร ก็จะเป็น ตัวบอกของรายการที่สอบถาม ซีจีไอก็จะนำค่านี้มาค้นหารายการในฐาน ข้อมูล จากนั้นฐานข้อมูลจะส่งข้อมูลที่ต้องการกลับมายัง ซีจีไอ โปรแกรม ซีจีไอ จะทำการสร้าง เพิ่มข้อมูลในแบบ เอชทีเอ็มแอล ใต้ผลลัพธ์ที่ได้จากการสอบถามจากฐานข้อมูลไว้ในเพิ่มข้อมูลนี้ แล้วส่งกลับมาให้ ผู้ให้บริการเว็บ เพื่อให้ ผู้ให้บริการ ส่งกลับมายัง โปรแกรมที่ใช้ดูเอกสารให้ผู้ใช้ ทราบต่อไป แต่รายการที่ส่งกลับมามีการระบุชนิดของเพิ่มข้อมูล นั้นไว้ด้วย โดย ผู้ให้บริการ จะใส่ ค่าตัวแปรที่ใช้งานหลัก (Environment Variable) ชนิด การบอกการใช้เอกสาร (Content_type) เพื่อบอกชนิดของเพิ่มข้อมูล ไว้ที่ส่วนหน้าสุดของเพิ่มข้อมูล (Header Field) ด้วย

2.6.2 ตัวแปรรอบด้าน (Environment Variable)

คือ สิ่งที่เราสนใจ (Entity) ที่อยู่ในส่วนใช้งานกับ สิ่งรอบด้าน ของ เครื่องคอมพิวเตอร์ ของ ผู้ใช้งานที่ใช้งาน ซึ่ง ซีจีไอ เป็นตัวผ่านข้อมูลของ เอชทีทีพี ไปยัง เครื่องที่ให้บริการ โดยมีโปรแกรมการทำงาน ของ ซีจีไอ ทำหน้าที่นี้โดยที่ค่าตัวแปรจะถูกค้นหาได้ทั้งจากเครื่องที่ให้บริการ และ โปรแกรมการทำงาน ของ ซีจีไอ เอง

ผู้ให้บริการเว็บ ใช้ ค่าตัวแปรที่ทำงานทีละบรรทัด (Command Line Argument) ที่เรียกว่า ค่าตัวแปรรอบด้าน เป็นตัวผ่านข้อมูลที่ต้องการจากผู้ให้บริการ โดยมีการ กำหนด หรือควบคุม ค่าตัวแปรของโปรแกรม หรือที่จุดเริ่มต้นของโปรแกรมได้ ซึ่งวิธีการนี้ทำให้ ผู้ให้บริการ ผ่าน การรับข้อมูล ที่เป็นการต่อเนื่องสั้นๆ ไปยัง โปรแกรมการทำงาน ของ ซีจีไอ ที่ เกี่ยวข้องกัน ในทาง เดียวกัน ตัวแปรเหล่านี้จะถูกจัด หรือกำหนดให้เป็นค่าใดๆ เมื่อ เครื่องที่ให้บริการเริ่มทำงาน กับ โปรแกรมการทำงาน ของ ซีจีไอ โดยที่ ค่าตัวแปรรอบด้าน

2.6.3 การทำงานของซีจีไอ กับการรับและแสดงผล (Input/Output CGI)

คำคำ อินพุท เมื่อเครื่องที่ให้บริการ ได้รับข้อมูลจากเครื่องที่ใช้บริการ ในการทำงาน เครื่องที่ให้บริการ จะส่งจำนวน ไบท์(Byte) ของข้อมูลไปยัง โปรแกรมที่ทำงานด้วย ซีจีไอ อย่างน้อยเท่ากับจำนวนใน คอนเทนเลนจ์ (Content_length) โดยใช้ มาตรฐานการรับข้อมูล (Standard Input, STDIN) เป็นตัวผ่านข้อมูลให้กับ ซีจีไอ ซึ่งก็จะอ่านค่าที่เท่ากับจำนวนใน คอนเทนเลนจ์ เช่นกัน

จากนั้นเครื่องที่ให้บริการจะนำค่าข้อมูลที่กำหนดโดย คอนเทนไทป์ ให้ผ่านไปยัง โปรแกรมที่ทำงานด้วย ซีจีไอ เมื่อข้อมูลถูกต้อง (CGI ยอมรับทำงาน) จะใช้ข้อมูลนี้ในการแปล ความหมายใน ส่วนท้ายของข้อมูลโดยไม่มี การส่ง ค่าที่บอกการจบเพิ่ม (End Of File, EOF) ไปให้ ด้วย เพราะผู้ให้บริการ และ ซีจีไอ จะรู้ว่าหมดข้อมูล จากค่าใน คอนเทนเลนจ์ แล้ว ตัวอย่างเช่นถ้าเราใช้ ฟอร์ม (FORM) ของ เอชทีเอ็มแอล ในการทำ คิวรี ด้วยวิธีการแบบ พีไอเอสที (POST) โดยให้มีการเข้ารหัส ทุกๆ 15 ไบท์ จะต้องทำการกำหนดค่าใน คอนเทนเลนจ์ เท่ากับ 15 และถ้าให้ มี คอนเทนไทป์ เท่ากับ application/x-www-form-urlencoded โดยที่ ไบท์แรกของ สคริปต์ มาตรฐานการรับข้อมูล (STDIN Script) จะเป็นตัวบอกค่าเริ่มต้น จนถึงอักขระก่อนตัวสุดท้าย คือตัวที่ 14 จะเป็นการบอกถึงการสิ้นสุดข้อมูลที่ส่งมาของข้อความนี้เช่นถ้าส่งค่า one=bob&two=cat

โปรแกรมจะรับค่าเริ่มต้นไปเรื่อยๆ จนถึงตัวอักษร “a” โปรแกรมจะรู้ว่าสิ้นสุดข้อความแล้ว เพราะเป็นตัวอักษรที่ 14 (เริ่มจาก “o”) และตัวต่อไป ต้องเป็นตัวสุดท้ายแน่นอน

คำคำ เอาท์พุท เมื่อ โปรแกรมที่ทำงานด้วย ซีจีไอ ส่งข้อมูลกลับมาให้เครื่องที่ให้บริการ จะส่ง เป็น มาตรฐานการแสดงผล (Standard Output, STDOUT) และที่ เครื่องที่ให้บริการ จะสร้าง เป็น เอกสาร เอชทีเอ็มแอล จากโปรแกรมที่ทำงานของ ซีจีไอ นั้น หรืออาจเป็นข้อมูลที่ เรียกย้อนคืน คำสั่งของเครื่องที่ให้บริการก็ได้

เอาท์พุท (Output) จะสามารถรับค่ากลับมาได้ 2 แบบ คือ

1. ไม่ทำการแยก หัวของเอาพุท (Header Output) โดยการใช้ <FORM> ซึ่ง โปรแกรม ที่ทำงานด้วย จะทำ การส่งข้อมูลกลับไปอย่างสมบูรณ์ให้กับ เอชทีทีพี
2. ทำการแยก หัวของเอาพุท ออกจากกัน ซึ่ง เครื่องที่ให้บริการ จะทำการใช้ <FORM> นี้ ซึ่งข้อมูลที่ส่งกลับ โดย ซีจีไอ จะถูกแยกเป็น 2 ส่วน คือ
 - ส่วนหัว (Header)
 - ส่วน ตัวข้อมูล (Body)

โดยใช้การเว้นบรรทัด เป็นตัวแยก และที่หัวจะถูกทำการแปลความหมายโดยเครื่อง ที่ให้บริการ ส่วนตัวข้อมูลจะเป็น การเลือกได้ว่า ที่ใช้บอก หรือกำหนด เอ็มไอเอ็มอี ของ คอนเทนไทฟ์ ที่ ส่วนหัว ด้วย

2.6.4 การส่งข้อมูลด้วยวิธีการ POST และ GET

การที่ผู้ใช้บริการ ได้ส่งข้อมูลมายัง เครื่องที่ให้บริการ โดยใช้ โปรแกรมที่ใช้ข้อมูลจะต้องมีการกำหนดวิธีการส่งข้อมูลมาด้วย เพื่อให้ เครื่องที่ให้บริการ ทราบและดำเนินการต่อไป อย่างถูกต้องโดยมีวิธีการอยู่ 2 แบบ คือ

2.6.4.1 วิธีการแบบโพสท์ (POST Method) จะเป็นการบอกเครื่องที่ให้บริการว่าจะข้อมูลมาเป็นส่วนๆ ซึ่งใน คิวรีสตริง จะใช้เครื่องหมาย "&" คั่นระหว่างตัวแปร แต่ใช้เครื่องหมาย "+" แทนการเว้นวรรค (Space) เช่น

```
QUERY_STRING = "star=Eastwood&cert=15&movie=Pale+Rider"
```

เมื่อทำการแยกเป็นคำจะได้ว่า star=Eastwood และ cert=15 และ movie=Pale Rider โดยที่เครื่องหมายแต่ ละตัวจะถูกแปลงให้อยู่ในรูปของเลขฐาน 16 และถูกนำมาแยกตัวแปรเก็บในอาร์เรย์ โดย ซีจีไอสคริปต์ อีกครั้งหนึ่ง จะเป็นวิธีการที่ได้รับความนิยม เนื่องจากข้อมูลที่ส่งจะมี เป็นส่วนๆ ทำให้สะดวกกว่า การใช้วิธีการอื่น

2.6.4.2 วิธีการแบบ เกต (GET Method) เป็นการรวบรวมข้อมูลที่ตำแหน่งที่ใช้เก็บเอกสารแล้วเครื่องที่ให้บริการจะส่งให้ โปรแกรมที่ทำงานด้วยซีจีไอในรูปของตัวแปรคิวรีสตริง โดยมีเครื่องหมายคำถาม "?" คั่นไว้ให้รู้ ว่าหลังเครื่องหมาย ? นี้เป็นคิวรีทั้งหมด โดยที่ ผู้ให้บริการ จะเรียกข้อมูลที่อ้างอิงจากตำแหน่งที่ใช้เก็บเอกสาร ที่ใช้วิธี เกต จาก คิวรีแบบ <ISINDEX> ซึ่งเป็นการเรียกค้นหา ตำแหน่งของฐานข้อมูล จาก การบอกด้วย<ISINDEX> เพื่อใช้อ้างอิงตำแหน่งที่ใช้ เก็บเอกสาร นั้น ส่งค่าไปให้โปรแกรมที่ทำงานด้วย ซีจีไอ ทำการค้นหาข้อมูลที่ต้องการ เช่น

```
<HEAD>
```

```
<ISINDEX href="http://www.hal.com/hal-bin/query">
```

```
</HEAD>
```

จากตัวอย่าง ซีจีไอ จะรู้ว่าตำแหน่งที่ใช้เก็บเอกสาร ที่กำหนดโดย <ISINDEX> โดยมีส่วนที่เป็น เอชอาร์อีเอฟ (HREF) เป็นตัวบอกที่ที่จะติดต่อไป ส่วน

```
http://www.hal.com/hal-bin/query?blues+brothers
```

ซีจีไอ จะได้รับค่าเป็น blues กับ brothers โดยที่ เครื่องหมาย “+” จะถูกแปลงเป็นเลขฐาน 16 = %2B ส่วนช่องว่างข้างหลัง จะถูกแปลงเป็นเลขฐาน 16 = %20 และที่โปรแกรมที่ใช้ดูเอกสาร จะรับค่าต่างๆเหล่านี้ และทำการแยกเป็นคำโดยดูจากเครื่องหมายดังกล่าว ดังตัวอย่าง

<http://www.hal.com/hal-bin/query?blues%2Bbrothers%20>

โดยปกติถ้ามีการใช้เครื่องหมาย ? หลังตำแหน่งที่ใช้เก็บเอกสารใน เอชทีทีพี จะกำหนดค่าวิธีการเป็น เกต โดยอัตโนมัติ และค่าใน คิวรีสตริง จะถูกกำหนดให้เป็นค่าตามหลังเครื่องหมาย ? เสมอ และมีเครื่องหมาย + เป็นตัวบอกการแยกช่องว่างไว้เป็นคำๆ เช่น

 และ

จะได้ผลลัพธ์ที่ คิวรีสตริง เป็น <QUERY_STRING = One+Two+Three+Four> เช่นกันทั้งสองกรณีข้างบน

จากนั้น จะใช้ภาษาสคริปต์ หรือภาษาที่มี มาตรฐานการรับและแสดงผล เขียนเป็น โปรแกรมที่ทำงานด้วย ซีจีไอ เพื่อนำค่าที่ได้รับจากโปรแกรมที่ใช้ดู เอกสาร มาทำการรวม รวมเป็นความหมายต่อไป

2.6.5 การเลือกรูปแบบในการที่จะเขียนสคริปต์

การเขียนสคริปต์ ขึ้นมาด้วยภาษาใดนั้น จำเป็นที่จะต้องศึกษาถึง ระบบปฏิบัติการ ที่ใช้ งานอยู่ โดยที่โปรแกรมที่ทำงานด้วย ซีจีไอ มีงานที่ทำงานได้ง่ายในเครื่องที่มีระบบปฏิบัติการหนึ่ง แต่จะทำได้ยากกับเครื่องที่ระบบปฏิบัติ การอีกอย่างหนึ่ง ดังนั้นการเขียน การทำงานของซีจีไอ ควรที่จะเลือกรูปแบบที่เหมาะสม ให้เข้ากับชนิดของงานที่เราจะทำ ดังต่อไปนี้

2.6.5.1 งานเกี่ยวกับข้อความและการค้นหา (TEXT And SEARCHING)

ควรเขียนโปรแกรมสคริปต์ บนยูนิกซ์ เนื่องจากยูนิกซ์ เป็นระบบที่มีคำสั่ง ในภาษาชั้นสูง ที่จะทำงานเกี่ยวกับการค้นหา การเรียงข้อมูล และอื่นๆที่เกี่ยวข้องกับข้อมูลที่เป็น ตัวอักษร ทำให้เกิดความสะดวกในการเขียนเป็นสคริปต์ขึ้นมา ส่วนในระบบ ดอส ก็สามารถทำได้เช่นกัน แต่มีข้อจำกัดอยู่มาก ส่วน ระบบวินโดวส์ จะต้องใช้ ดอส เป็นตัวช่วย (BATCH) ไว้ เพราะไม่สามารถทำงานในแบบ แปลการทำงานเป็นคำสั่ง (Command Line) ได้ ทำให้ต้องเสียเวลาในการทำงานที่ ดอส ก่อน จึงไม่เหมาะที่จะใช้ทำงานประเภทนี้

2.6.5.2 งานที่ไม่เกี่ยวกับข้อความ (NON TEXT DATA)

ได้แก่งานข้อมูลที่ได้จากการใช้โปรแกรมประมวลผลคำ (Word Processing) เช่น ตารางต่างๆ หรือเอกสารการเปรียบเทียบ ควรเขียนโปรแกรมสคริปต์ ด้วย วินโดวส์ เนื่องจากระบบ

วินโดว์สามารถที่จะแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ด้วยกันได้ดีทำให้ง่ายและสะดวกกว่า คอส และ ยูนิกซ์

2.6.5.3 งานที่เกี่ยวกับการประมวลผลฐานข้อมูล (DATABASE ACCESS)

จะขึ้นกับชนิดของโปรแกรมที่ใช้เป็นฐานข้อมูลนั้นๆ ว่าทำงานอยู่ภายใต้รูปแบบ การทำงาน เช่นไร ถ้าใช้บนยูนิกซ์ เช่น โปรแกรม ออราเคิล (Oracle) หรือ ซายส์เบส (Sybase) ก็จะต้องเขียนโปรแกรม ซีจีไอ บนยูนิกซ์ ถ้าใช้เป็น ระบบวินโดว์ เช่น ใช้ไมโครซอฟท์แอคเซส (Microsoft Access) ก็เขียนโปรแกรมการทำงานของซีจีไอ ภายใต้ระบบวินโดว์

2.6.6 หลักทั่วไปในการเขียน ซีจีไอ

การทำงานของซีจีไอคือการรับข้อมูลจากผู้ใช้บริการส่งมาให้เครื่องที่ให้บริการจากนั้น โปรแกรมการทำงานของซีจีไอที่อยู่ในเครื่องที่ให้บริการจะทำงาน โดยติดต่อกับแหล่งข้อมูล ที่ผู้ใช้บริการต้องการ เมื่อได้เป็นผลลัพธ์ หรือข้อมูลที่ต้องการ ก็จะส่งกลับมาให้เครื่องที่ให้บริการ ทำการแสดงผล ให้ผู้ใช้บริการ ทราบได้ ซึ่งมีหลักในการเขียนซีจีไอ เป็นสคริปต์ ดังนี้

2.6.6.1 การรับข้อมูล (Data Receiving)

ทำได้หลายแบบขึ้นกับ รูปแบบที่ต้องการใช้ และวิธีการที่ผู้ใช้บริการเว็บ สามารถที่จะยอมรับให้ทำงานได้ ดังนี้

ตารางที่ 3 ตารางเปรียบเทียบประเภท ซีจีไอ กับการใช้งานต่างๆ

CGI	GET	POST
UNIX	Environment Variable	STDIN
DOS	Environment Variable	Content File
WINDOWS	CGI Data File	Content File

จะเห็นว่า วินโดว์ มีการใช้เพิ่มข้อมูลใหม่ทั้งแบบ เกต และ โปสท์ คเพื่อส่วนผ่านข้อมูล และ ตัวแปร มายัง โปรแกรมของ ซีจีไอ เพราะวินโดว์ไม่สามารถใช้ตัวแปรแวดล้อม แบบยูนิกซ์ได้

2.6.6.2 การแยกข้อมูล (Data Parsing)

เมื่อผู้ใช้บริการป้อนข้อมูลมายังเครื่องที่ให้บริการเครื่องที่ให้บริการก็จะทำการ แยกข้อความของผู้ใช้บริการออกเป็นส่วนๆเพื่อใช้ในการทำงานของโปรแกรมซีจีไอต่อไปในส่วนของช่องว่างจะเปลี่ยนให้เป็นเครื่องหมาย "+" และเปลี่ยนเครื่องหมายคณิตศาสตร์ รวมทั้ง อักขระพิเศษ ให้เป็นเครื่องหมายตามรหัสแอสกี เมื่อโปรแกรมของซีจีไอ ทำงานก็จะ แปลงค่ากลับ ให้เป็นข้อมูลเดิม เพื่อทำงานต่อไป

2.6.6.3 ส่งผลลัพธ์กลับไปยังผู้ใช้บริการ (Result Returning)

โปรแกรมของซีจีไอ สามารถที่จะส่งข้อมูลกลับ มาให้เครื่องที่ให้บริการ ได้ 3 แบบ

1. ส่งกลับในรูปเอกสารเอชทีเอ็มแอล (HTML Document return) โปรแกรมซีจีไอ จะต้องสร้างเอกสารเอชทีเอ็มแอลขึ้นมาแล้วส่งกลับไปซึ่งจะต้องมีส่วนหัวของ เอกสารเพื่อกำหนดการทำงานไว้ด้วย เช่น

Content_Type : Text/html

<TITLE> HEADER </TITLE>

<H1> RETURNING </H1>

This is Feedback Result. It returning by HTML

2. ส่งข้อมูลกลับในลักษณะอื่นๆ อาจเป็นในรูปของ เสียง, ภาพ, โปรแกรมต่างๆ โดยจะ ต้องแจ้งไว้ในส่วนของหัวเอกสารที่ส่ง ไปให้ผู้ใช้บริการรู้ถึง ชนิดของข้อมูลที่ส่งมาด้วย

Content_Type : Image/gif

Content_Type : Sound/MIDI

3. ส่งกลับในลักษณะของตำแหน่งที่เก็บเอกสาร โดยจะบอกเป็นตำแหน่งที่เก็บเอกสาร ให้ผู้ที่ ใช้งานทราบ เพื่อที่จะได้ ใช้โปรแกรมในการดูเอกสาร ดูข้อมูลที่ตองการนั้น เช่น

Location : protocol://hostname/path_info/

2.6.7 การใช้ FORM เพื่อรับข้อมูลจากผู้ใช้บริการ (FORM TAG)

วิธีการที่จะรับข้อมูลจากผู้ใช้บริการที่ง่ายที่สุด คือการใช้รูปแบบของ แบบฟอร์ม ในการให้ผู้ใช้งานกรอก หรือเติมข้อความที่ต้องการ โดยอาจจะมี ข้อความ, ปุ่ม, เมนู, และใช้การ เลือกข้อความแบบ เชคบอกร์ (Checkbox) เพื่อรับข้อมูลแล้วส่งไปให้ เครื่องที่ให้บริการ ทำงานส่ง ต่อไปให้โปรแกรมที่ซีจีไอ ทำงานที่ต้องการต่อไป

ส่วนประกอบของฟอร์ม มี 3 ส่วนคือ

- ส่วนเริ่มต้น (Header)
- ส่วนแสดงชื่อ และตำแหน่งที่ใช้งาน (Names Input Fields)
- ส่วนที่เป็นการทำงานในการเลือกข้อมูล

1. ส่วนเริ่มต้น

- FORM เป็นตัวบอกเริ่มต้นว่าในเอกสารนั้นจะมีการใช้ เป็นข้อมูลที่ส่งกลับ ไปให้เครื่องที่ให้บริการต่อไป ต้องมีส่วนปิดท้ายบอกการหยุดรับข้อมูลนี้ด้วย </FORM>

2. ส่วนแสดงชื่อ และตำแหน่งที่ใช้งาน

- ACTION เป็นตัวบอกตำแหน่งที่จะมีการทำงานด้วย ซีจีไอ ในการที่นำข้อมูลนี้ไปปฏิบัติ ต้องเขียนร่วมในส่วน FORM เช่น
<FORM ACTION="ตำแหน่งที่เก็บ โปรแกรม ซีจีไอ">
- METHOD เป็นวิธีการที่ใช้ในการติดต่อส่งข้อมูลกัน ระหว่างเครื่องที่ให้บริการ กับ โปรแกรมซีจีไอ ควรเลือกวิธีการนี้ให้ถูกต้องกับเครื่องที่ให้บริการ โดยทั่วไปจะมี 2 แบบ คือ POST , GET
 - POST เป็นการส่งข้อมูลจากแหล่งข้อมูลมายังโปรแกรมที่ใช้ข้อมูล
 - GET เป็นการรับข้อมูลจาก ผู้ใช้บริการ มายังให้บริการ และโปรแกรม ซีจีไอ

3 ส่วนที่เป็นการทำงานในการเลือกข้อมูล

- INPUT เป็นการสั่งให้มีการรับข้อมูลจากผู้ใช้งาน โดยมีรูปแบบย่อยๆ ให้เลือกใช้งาน เพื่อให้น่าสนใจ และสะดวกในการป้อนข้อมูล
- TEXT จะเป็นการกำหนดให้ข้อมูลที่ป้อนเป็นข้อความขนาดสั้นๆ และมีการกำหนดชื่อให้กับข้อความนั้นด้วย
<INPUT TYPE="TEXT" NAME="Name">
ยังสามารถที่จะกำหนดขนาดความกว้างของช่องที่ป้อนข้อมูลได้ด้วยคำสั่ง SIZE เช่น
<INPUT TYPE="TEXT" SIZE= 25 NAME="Name">
- CHECKBOX จะเป็นการเลือกหัวข้อที่กำหนดไว้ให้ โดยผู้ให้บริการสามารถที่จะเลือกโดย กำหนดค่าในช่องนั้นๆ ได้ทำเป็นรูปสี่เหลี่ยม สามารถที่จะเลือกแล้วเกิดเป็นเครื่องหมาย (Mark) แจ้งให้ทราบ
- RADIO จะมีการทำงาน และ ใช้งานที่เหมือนกับ Checkbox แต่รูปร่างจะเป็นวงกลม เมื่อเลือกหัวข้อแล้ว แสดงเป็นจุดคากลางวงไว้ให้ทราบ
- SUBMIT เป็นการสั่งให้ทำงาน โดยจะส่งข้อมูลที่ป้อนไว้แล้วไปยังเครื่องที่ให้บริการ แล้วให้โปรแกรมซีจีไอ ใช้งานต่อไป

จะเกิดเป็นกล่องสี่เหลี่ยม ให้เลือกและมีข้อความตามที่กำหนดได้ด้วย

- RESET เป็นการยกเลิกการทำงาน หรือยกเลิกข้อมูลที่ป้อนไว้ ก่อนหน้านี้ ทั้งหมด โดยจะกลับไปเริ่มที่จุดเริ่มต้นใหม่ ข้อมูลจะยังไม่ถูกส่งไป จะเกิดเป็นกล่องสี่เหลี่ยม ให้เลือก และมีข้อความตามที่กำหนดได้ด้วย เป็นการ ทำงานที่ตรงข้ามกับ SUBMIT
- SELECT เป็นการกำหนดให้ผู้ใช้งาน เลือกข้อมูลในย่านของหัวข้อที่กำหนด เป็นการเฉพาะ มีการใช้งานร่วมกับคำสั่ง OPTION เพื่อใช้เป็นตัวกำหนดหัวข้อให้เลือก เช่น

```
<SELECT NAME="Do">
<OPTION> YES
<OPTION> NO
<OPTION> MAYBE
</SELECT>
```

- TEXTAREA เป็นการกำหนดพื้นที่เพื่อใช้ผู้ใช้งานสามารถที่ป้อนข้อมูล ขนาดมากๆ ได้ เพื่อใช้ในการแสดงความคิดเห็น หรือข้อเสนอแนะ สามารถที่จะกำหนดขนาดที่ให้อป้อนข้อมูลได้
- ```
<TEXTAREA NAME="Comment" COLS=40 ROWS=10>
```

## 2.7 ASP

อะไรคือ ASP

ASP ย่อมาจาก Active Server Pages อ่านออกเสียงว่า "เอ เอส พี" เจ้าของหรือผู้คิดค้นคือ บริษัท Microsoft (<http://www.microsoft.com>) ASP เป็นโปรแกรมคอมพิวเตอร์ชนิดหนึ่งที่เจ้าของได้ ให้นิยามไว้คือ "Server side scripting" ซึ่งหมายถึงภาษาการโปรแกรมที่ทำงานในฝั่งของเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น web server ที่ให้บริการเอกสารหรือสื่อต่างๆ ในอินเทอร์เน็ตหรือ อินทราเน็ต

ASP มีประโยชน์อย่างไรจำเป็นต้องใช้ ASP และใครไม่จำเป็นต้องใช้

ASP ใช้สำหรับสร้างงาน(application)ขั้นสูง ในอินเทอร์เน็ต-อินทราเน็ต เสริมการทำงานที่ไฟล์ html ธรรมดาทำไม่ได้ หรือต้องการให้งานต่างๆ เป็นไปอย่างอัตโนมัติโดยผู้ใช้ไม่ต้องทำการ

ปรับปรุงข้อมูลเอง เหมาะสำหรับผู้ใช้อีเมลหรือผู้ที่สนใจอย่างจริงจัง หรือทำเป็นอาชีพสำหรับผู้  
ใช้ที่ไม่ต้องการเว็บไซต์ที่มีลักษณะดังกล่าวข้างต้น หรือมีข้อมูลน้อยเพียงไม่กี่หน้างานจะปรับปรุง  
ข้อมูลสักครั้ง ก็ไม่จำเป็นต้องใช้ก็ได้ สำหรับงานที่ต้องการให้เป็นอัตโนมัติเช่น Guest Book,  
Counter,สถิติ, ห้องสนทนา ก็ สามารถสมัครใช้บริการหรือใช้ cgi สำเร็จรูปได้ซึ่งมีหลายเว็บไซต์ให้  
บริการโดยไม่คิดค่าใช้จ่าย

### ค่าใช้จ่ายในการใช้ ASP

ในการจัดตั้งเว็บไซต์ที่ใช้ ASP ด้วยจะมีค่าใช้จ่ายเพิ่มขึ้นมาอีกเล็กน้อยเท่านั้น สำหรับนัก  
พัฒนาหรือผู้ต้องการสร้างงานจาก ASP ก็ไม่ต้องลงทุนอะไรเพราะ โปรแกรมต่างๆเป็น โปรแกรมที่ใช้  
ได้โดยไม่เสียค่าใช้จ่าย

### ความต้องการพื้นฐานในการใช้ ASP ในส่วนของนักพัฒนาโปรแกรม/เว็บไซต์

- เครื่องคอมพิวเตอร์ PC ความเร็ว CPU 486 ขึ้นไป RAM 16 MB ขึ้นไป ถ้ายิ่งสูงจะยิ่งดีเนะ  
นำที่ Pentium 150 MHz ขึ้นไป RAM 32 MB

- ระบบปฏิบัติการ(OS) Windows 95/98 หรือ Windows NT 4.0 (server/work-station)

- สำหรับ Windows 95/98 ต้องมี

1. บราวเซอร์ IE 3-5

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

3. โปรแกรม ASP (จะมี ASP รวมอยู่ใน PWS เวอร์ชัน 4 ขึ้นไป) แต่ถ้าลง โปรแกรม  
webserver อื่นๆหรือ PWSต่ำกว่าเวอร์ชัน 4 จะต้องลง ASP ด้วย หาได้จากชุด MS  
Visual Studioสำหรับ Windows NT 4.0 ต้องมี IIS 3.0(Internet Information Server)

ขึ้นไป และ ASP พร้อมอยู่แล้วหรือถ้ายังไม่ให้ลงโปรแกรมเพิ่มเติม!! โปรแกรมที่จำ  
เป็นทั้งหมดสามารถ download ได้จากเว็บไซต์ของ Microsoft  
ที่<http://www.microsoft.com/msdownload> !! ข้อควรจำ ต้องใช้โปรแกรม ASP ให้  
เข้ากับภาษาของ OS ด้วยเช่น ถ้ามี Windows 95 ภาษาไทยท่านก็ต้องใช้ ASP ที่  
สนับสนุนภาษาไทยเป็นต้น

- ความรู้ในการใช้โปรแกรม Windows95/98+PWS หรือ NT+IIS

- ความรู้ในการใช้ HTML ในระดับ Source Code รวมถึงการใช้โปรแกรม Text Editor ใดๆ  
เช่นNotepad เป็นต้นสำหรับผู้ที่ใช้โปรแกรมสำเร็จรูปในการสร้างไฟล์ html ควรทราบถึงวิธีการ  
แทรก script ใน html source code ด้วย

- ความรู้ในการใช้ภาษา Java script, VB script, Perl script อย่งใดอย่างหนึ่ง
- ความรู้ในการใช้เครือข่ายอินเทอร์เน็ต โดยจะต้องใช้โปรแกรม FTP เป็นในระดับหนึ่ง (upload/download/create/editdirectory)
- ถ้าต้องการใช้ฐานข้อมูลในอินเทอร์เน็ตด้วยก็ควรทราบถึงหลักการฐานข้อมูล และการใช้โปรแกรมฐานข้อมูลเช่น MS Access, FoxPro, Dbase หรือ Flat file database ตลอดจน ภาษา SQL และ ODBC

### เว็บไซต์ที่ให้บริการข้อมูลเรื่อง ASP เป็นภาษาอังกฤษ

- <http://www.microsoft.com> (แนะนำ)
- <http://www.activeserverpages.com> (แนะนำ)
- <http://www.aspalliance.com>
- <http://www.asphole.com>
- <http://www.15seconds.com>

### หลักการทํางานของ ASP

เริ่มจากผู้ใ้ ASP สร้างไฟล์ที่มีนามสกุลเป็น .asp ขึ้นมา จากนั้นนำไฟล์นั้นไปไว้ในเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น web server ที่ติดตั้งโปรแกรม ASP ไว้ และเชื่อมต่ออยู่กับเครือข่ายอินเทอร์เน็ต จากนั้นเมื่อมีผู้ใ้รายใดเรียกใช้ไฟล์นั้นผ่านโปรแกรมบราวเซอร์(IE, Netscape ..) โปรแกรม ASP ใน web server จะเรียกไฟล์นั้นขึ้นมาอ่านแล้วทำตามคำสั่งต่างๆที่ผู้สร้างไฟล์นั้นได้กำหนดขึ้น จึงส่งผลที่ได้กลับไปให้ผู้เรียกใช้โดยแสดงผลที่โปรแกรมบราวเซอร์ของผู้เรียก ซึ่งขั้นตอนข้างต้นเป็นหลักการทํางานโดยทั่วไปของ ASP (ตัวอย่างประกอบ) !! สำหรับผู้ที่เคยใช้ html มาแล้วสามารถอธิบายให้ง่ายขึ้นได้คือ ไฟล์ .asp ก็คือไฟล์ .htm หรือ .html ที่จับมาเปลี่ยนนามสกุลไฟล์เสียใหม่ให้เป็น .asp นั่นเอง ซึ่งการที่เปลี่ยนไปแค่นามสกุลก็ถือว่าเป็น ASP แล้ว แต่คงไม่ง่ายๆแค่นี้เนื่องจากผู้ใ้สามารถแทรกภาษาการโปรแกรมไว้ในไฟล์นี้ได้ เหมือนๆกับการแทรก Java Script หรือ VB Script กำหนดให้เกิดงานที่ต้องการได้ตามขอบเขตที่ ASP สามารถทำได้จะแตกต่างกันก็คือการแทรก Script ในไฟล์ .htm/.html Script จะทำงานที่ฝั่งของผู้เรียกใช้ไฟล์ แต่การแทรกในไฟล์ .asp สามารถกำหนดให้ทำงานฝั่ง server หรือฝั่งของผู้เรียกใช้(client)ก็ได้!! หรือสำหรับผู้ใ้คุ้นเคยกับการใช้ perl cgi ASP แต่ละไฟล์ก็เปรียบคล้ายๆกับไฟล์ .cgi.pl แค่นามสกุลต่างกันและมีตัวแปลภาษาต่างกัน กล่าวคือ .cgi/.pl จะใช้ perl ในการแปล แต่ .asp จะใช้ ASP เป็นตัวแปล script ในไฟล์ .asp และสิ่งที่เป็นความจริงส่วนใหญ่ก็คือ การสร้าง ASP จะง่ายกว่าการสร้าง cgi ทั่วไปอย่างมาก

### ตกลงกันก่อนเรียน

- ก่อนเรียนคุณต้อง ทำ/มี หรือรู้สิ่งต่อไปนี้เป็นอย่างดีแล้ว
- ลงโปรแกรมครบถ้วน ในส่วนของ ASP คุณได้ลงส่วนข้อมูล+บทเรียนที่เรียกว่า Roadmap ไปแล้ว (ถ้าลง ASP เต็มๆจะมีตัวนี้อยู่)

- HTML สามารถแก้ไข เพิ่มเติมได้ในระดับ source code ใช้ form / frame / table เป็น
- MS Access + ODBC และการ create DSN (ศึกษาได้จาก Roadmap ถ้าจะไม่ใช่ database ก็ข้ามข้อนี้ไปได้)

- พื้นฐานการใช้ windows เช่น การสร้าง เปลี่ยนชื่อ ลบ ทำสำเนา ไฟล์/folder, การ save ไฟล์ เปิดไฟล์, การ copy/cut & paste เป็นต้น

- ใช้บราวเซอร์เป็นอย่างน้อย 2 ชนิดคือ ie และ netscape
- ใช้ IIS หรือ PWS เป็นแล้ว ถ้ายังศึกษาได้จาก Help ของโปรแกรม
- VB script ไม่จำเป็นต้องเก่ง สามารถศึกษาได้ในบทเรียน
- การศึกษาที่นี้ที่เดียวทำให้คุณทราบพื้นฐาน โดยทั่วไปเท่านั้น เพียงพอที่คุณจะนำความรู้ไปใช้สร้างงานธรรมดาๆในลักษณะเว็บไซต์ไทยที่อบได้ แต่คุณก็ควรศึกษาเพิ่มเติมจากแหล่งอื่นๆ

### ประกอบ

- ในบทเรียนที่จะได้นำเสนอเน้นการใช้งานจริง(In action) ไม่เน้นรายละเอียดในทฤษฎี
- ทุกๆเอกสารในบทเรียนนี้จัดทำโดย webmaster@thaitop.com และสงวนลิขสิทธิ์ตามกฎหมาย

### กำหนดคุณสมบัติของ Folder

ให้คุณสร้าง Folder(Directory) ที่มีโครงสร้างดังต่อไปนี้ไว้ในเครื่องของคุณ การสร้างอาจจะใช้ windows explorer หรือ DOS ก็ได้ เหตุผลที่ต้องสร้างลักษณะนี้เพื่อให้เป็นมาตรฐาน และความเข้าใจเดียวกัน

c:/ (หรือ e:/)

|-http

|- myserver

|- asp

|- cgi-bin

|- database

ในส่วนของ folder myserver เปลี่ยนเป็นชื่ออื่นได้เช่น ถ้าต้องการเว็บไซต์ชื่อ happy.com ก็ให้แทน myserver ด้วย happy เป็นต้น เมื่อสร้างเสร็จแล้วให้เปิด IIS(โดยใช้บราวเซอร์หรือเปิด service) หรือ เปิด PWS ด้วยบราวเซอร์จากนั้นให้กำหนด Default Document เป็น default.htm แล้วจึงกำหนดคุณสมบัติของ Folder ต่างๆที่ได้สร้างขึ้นดังนี้

- http กำหนดให้เป็น Home directory / Access = Read
- asp กำหนด Virtual Directory ให้เป็น /asp และ Access = Execute
- cgi-bin กำหนด Virtual Directory ให้เป็น /cgi-bin และ Access = Execute

สำหรับ Folder database ไม่ต้องทำอะไร การใช้งาน folder ต่างๆเป็นดังนี้คือ

- http เป็น root ของ server เก็บไฟล์ default.htm ซึ่งเป็นหน้าแรกของเว็บไซต์ ตลอดจนเก็บไฟล์อื่นๆที่ยอมให้ผู้ใช้ download ได้ เช่น .txt .gif .jpg .zip เป็นต้น

- asp เก็บไฟล์ และ run โปรแกรมที่เป็น .asp โดยไฟล์ที่เก็บในที่นี้ผู้ใช้โดยทั่วไปไม่สามารถ download ไฟล์ใดๆในที่นี้ได้ เพราะได้ตั้งคุณสมบัติไว้เป็น Excute จะทำให้ source code หรือ ไฟล์อื่นๆไม่ถูกนำไปเผยแพร่ และถ้าท่าน สร้าง Folder ใหม่ภายใต้ Folder นี้จะมีคุณสมบัติเดียวกันและสามารถ run .asp ได้เหมือนกัน

- cgi-bin มีคุณสมบัติเหมือน Folder asp ทุกประการ สำหรับเก็บโปรแกรม cgi อื่นๆได้เช่น counter เป็นต้น

- database สำหรับใช้เก็บไฟล์ฐานข้อมูลในอนาคต

### 2.7.1 พื้นฐาน

สร้างไฟล์ asp ไฟล์แรก และทดลองใช้

ขั้นตอนที่ 1 เปิดโปรแกรม notepad หรือ text editor อื่นๆขึ้นมาแล้วให้ copy code ต่อไปนี้ นำไป paste จากนั้นให้ save ในชื่อ hello.asp ไว้ใน folder asp ที่เคยสร้างไว้ [ /asp/hello.asp ]

```
<HTML>
```

```
<BODY>
```

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

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

```
Hello World!

```

```
<% Next %>
```

```
 ตัวอย่างนี้ นำมาจากไฟล์ตัวอย่างของ <a
```

```
href="http://www.microsoft.com">Microsoft
```

```
</BODY>
```

```
</HTML>
```

## ขั้นตอนที่ 2 เปิดโปรแกรมบราวเซอร์และพิมพ์เรียกที่อยู่ต่อไปนี้

`http://127.0.0.1/asp/hello.asp`

สรุป ผลที่ได้จะเห็นว่า มีคำว่า Hello World! อยู่ 5 ตัวและมีขนาดจากเล็กสุดไปใหญ่สุดซึ่งถ้าใช้ html ธรรมดาให้เกิดผลแบบนี้จะต้องเขียน code กำหนดขนาด font ถึง 5 ชุด แต่ถ้าใช้ asp code ก็จะใช้เพียงไม่กี่บรรทัดเท่านั้น จากนั้นให้คุณลอง view source code ของผลที่ได้ดู สิ่งที่คุณจะพบคือ html code ธรรมดา ไม่มี vb script อยู่ในนั้น ซึ่งแสดงให้เห็นว่าไฟล์ hello.asp ได้ถูกอ่านแล้วประมวลผลออกมาตามรูปแบบที่กำหนดและไม่ว่าจะเรียกใช้กี่ครั้งผลก็จะเหมือนเดิม ตัวไฟล์ hello.asp จะไม่เปลี่ยนไป และผู้เรียกโดยบราวเซอร์ก็ไม่สามารถดู code ตัวจริงในไฟล์ได้

การแทรก vb script ในไฟล์ตัวบอกรายละเอียดของ script ที่โปรแกรม ASP จะต้องแปลและส่งผลออกมาที่ browser จะเปิดด้วย <% และปิดด้วย %> สิ่งที่อยู่ระหว่างเครื่องหมาย 2 อันนี้จะต้องเป็น script ถ้านำ html มาใส่จะ error เมื่อเรียกใช้การ ใช้มีดังตัวอย่าง คือเปิดหน้าคำว่า for ด้วย <% และเริ่มแทรก script คือ for i = 3 to 7 จากนั้นจึงปิดหลังตัว i ด้วย > ซึ่งในขณะนี้หากในส่วนของ script แล้วจากนั้นก็จะเป็น html ธรรมดาไปจนถึงคำว่า size= ก็จะเปิด script อีก แล้วนำตัวแปร i มาใส่สรุปได้ว่าในส่วนของ script จะต้องเปิดด้วย <% และปิดด้วย %> การแทรก script นี้จะแทรกในส่วนของไฟล์ asp ก็ได้ (ดูได้จากตัวอย่างของการแทรกวันที่ในบทเรียนนี้) แต่ส่วนที่แสดงผลออกมาให้ผู้เรียกใช้ไฟล์ผ่านบราวเซอร์คือส่วนที่อยู่ระหว่าง <HTML> จนถึง </HTML>

```
[/asp/hello2.asp]
```

```
<HTML>
```

```
<BODY>
```

```
<%
```

กำหนดค่าให้ i มีค่าตั้งแต่ 3-7 และนำค่านี้ไปใส่แทน i ในส่วนของ size= .... 'วนไปเรื่อยๆตั้งแต่ 3 4... ไปจนถึง 7 จากนั้นจึงเปลี่ยน คำสั่งไปที่ next แล้วจึงปิดไฟล์ REM ถ้ามีการแทรก คำสั่งต่อจากบรรทัดนี้ คำสั่งต่างๆจะถูกแปลให้ทำงานได้

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

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

```
Hello World!

```

```
<% Next %><!-- แต่ถ้านำคำสั่งมาแทรกข้างล่างบรรทัดนี้หรือแทนที่คำสั่งไม่สามารถทำงานได้เพราะอยู่นอกเหนือตัวบอกรายละเอียดของ script ที่โปรแกรม ASP จะต้องแปล -->
```

```
</BODY>
```

```
</HTML>
```

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

4. การแทรกวันที่และไฟล์ ในไฟล์ .asp

[ /asp/hello3.asp ]

<% กำหนดตัวแปรคำว่า showdate ให้เป็นวันที่ธรรมดา 'ซึ่งคำว่า showdate/shownow สามารถเปลี่ยนเป็นคำอื่น ๆ ที่ไม่ขัดต่อการตั้งตัวแปรได้ เช่นคำว่า apple, i\_want\_to\_see\_date เป็นต้น showdate = date 'กำหนดตัวแปรคำว่า shownow ให้เป็นวันที่แบบยาว

shownow = now%>

<HTML>

<BODY>

<% For i = 3 To 7 %>

<FONT SIZE=<% = i %>>

Hello World!</FONT><BR>

<% Next showdate2 = date shownow2 = now %>

<HR>นำวันที่จากตัวแปรที่กำหนดไว้ข้างบน tag HTML มาแสดง <BR>showdate =

<%=showdate%> และ shownow =<%=shownow%><HR>

นำวันที่จากตัวแปรที่กำหนดไว้ข้างล่าง script คำว่า next มาแสดง

<BR>showdate2 = <%=showdate2%> และ shownow2 =<%=shownow2%><HR>

และต่อไปนี้เป็น การแทรกวันที่โดยตรง โดยไม่ต้องตั้งตัวแปรก่อน

<BR><%=date%> หรือ <%=now%>

</BODY>

</HTML>

การแทรกไฟล์ ในไฟล์ .asp มีประโยชน์คือลดการซ้ำซ้อนในการที่จะต้องใส่ข้อความที่เหมือนกันหรือใส่ตัวแปร / รูปภาพในหลายๆไฟล์ ซึ่งจะช่วยให้สะดวกในการแก้ไขในภายหลังไม่ต้องเสียเวลานำไฟล์ทั้งหมดมาแก้เพียงแก้ที่ไฟล์ที่แทรกไฟล์เดียวกันก็สามารถปรากฏผลความเปลี่ยนแปลงได้พร้อมกันทั้งหมดให้เห็นภาพง่ายคือข้อความที่เป็น footnote ของเว็บเพจ เป็นต้น

การทดลองทำการแทรกไฟล์

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

## ขั้นตอนที่ 2 save ไฟล์ต่อไปนี้ และเรียกเพื่อดูผลผ่านบราวเซอร์

```
[/asp/hello4.asp]
<HTML>
<BODY>
<% For i = 3 To 7 %>
<FONT SIZE=<% = i %>>
Hello World!

<% Next %>
 ตัวอย่างนี้ นำมาจาก ไฟล์ตัวอย่างของ Microsoft
<hr>
ต่อไปนี้นำมาจาก ไฟล์ insertit.inc

<!--#include virtual="/asp/insertit.inc"-->
</BODY>
</HTML>
```

เพิ่มเติมในไฟล์ .inc คุณสามารถใส่ html tag(ตาราง, frame, ตัวอักษร, รูปและอื่นๆ) หรือ script ได้ และไม่จำเป็นต้องใช้ชื่อ insertit.inc การใช้งานที่แตกต่างหรือเทคนิคที่สูงขึ้นสามารถอ่านได้จาก

Road map

### 2.7.2 Content Rotator

เมื่อคุณเรียนบทเรียนนี้จบแล้วคุณสามารถสร้างหน้าแสดงข้อความสั้นๆเช่น Tip, สุภาษิต- คำคม หรือข้อความอื่นๆที่คุณต้องการให้ผู้เข้าเว็บไซต์อ่าน ให้แสดงหมุนเวียนกันไปครั้งละ 1 ข้อความได้

#### หลักการทำงาน

เมื่อผู้เรียกใช้ไฟล์ผ่านบราวเซอร์ โดยไฟล์นั้นได้มี script ของ Content Rotator ติดอยู่จะทำให้ ASP ไปอ่านรายการข้อความต่างๆที่เราได้กำหนดไว้ก่อนแล้ว สุ่มนำขึ้นมาแสดงที่บราวเซอร์ของผู้เรียกใช้ไฟล์นั้นได้อ่านครั้งละ 1 ข้อความ

#### ส่วนประกอบและขั้นตอน

1. ไฟล์เก็บรายการข้อความให้คุณ copy ข้อความต่อไปนี้แล้ว นำไป paste ลง notepad จาก นั้น save ไฟล์ไว้ใน folderasp ตั้งชื่อว่า tips.txt

No man is born wise or learned - ไม่มีผู้ใดฉลาดหรือรอบรู้มาแต่กำเนิด  
 Respect a man, he will do the same - จงเคารพผู้อื่นแล้วเขาจะเคารพท่านเช่นกัน  
 Quarreling is the weapon of the weak - การทะเลาะวิวาทเป็นอาวุธของคนอ่อนแอ  
 Men are blind in their own cause - คนเรามักจะตาบอดในเรื่องของตนเอง  
 Let us not be weary in well-doing - จงอย่าท้อถอยในการทำความดี  
 An honest man's word is his bond - คำพูดของคนซื่อสัตย์คือคำมั่นสัญญา  
 He that once deceives, is ever suspected - ผู้ที่เคยโกหกแล้วย่อมเป็นที่สงสัยเสมอ  
 The insolent have no friends - คนอวดดีไม่มีเพื่อน

2. ไฟล์แสดงข้อความ ให้คุณ copy code ต่อไปนี้แล้ว นำไป paste ลง notepad จากนั้น save ไฟล์ไว้ใน folder.asp ตั้งชื่อว่า showme.asp

```
<html>
<body>
<!-- เริ่มแทรก script Content Rotator-->
<%
Randomize
TipNumber = Int(Rnd*10)
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
Set Instream = FileObject.OpenTextFile (Server.MapPath ("/asp") & "\tips.txt", 1,
FALSE,FALSE)
For i = 1 to TipNumber -1
InStream.SkipLine()
Next
TipOfTheDay = Instream.ReadLine
'การดัดแปลงแก้ไข script นี้ ให้ระวังส่วนของ /asp(ชื่อ Folder ที่เก็บไฟล์ tips.txt) และ
\tips.txt%>
<!-- เริ่มส่วนแสดงผล -->
<center>
<h3>Today's Word by Content Rotator</h3>

<center>
<table border="1" CELLPADDING="1" CELLSPACING="0">
```

```

<tr><td VALIGN="middle">
<%= TipOfTheDay %>
</td></tr>
</table>
</body>
</html>

```

3. การเรียกไฟล์ swme.asp ในบราวเซอร์โดยพิมพ์ <http://127.0.0.1/asp/showme.asp> และ reload หรือ Refresh ไปเรื่อยๆ ดูผลที่ได้

### 2.7.3 Banner Rotator

หลังจากได้เรียนเรื่อง Content Rotator ซึ่งเป็นการแสดงข้อความไปแล้วในบทเรียนนี้เรามาเปลี่ยนเป็นแสดงรูปภาพบ้าง

#### หลักการทำงาน

เมื่อผู้เรียกใช้ไฟล์ผ่านบราวเซอร์ โดยไฟล์นั้น ได้มี script ของ Banner Rotator ติดอยู่จะทำให้ ASP ไปอ่านรายการรูปภาพและลิงค์ต่างๆที่เราได้กำหนดไว้ก่อนแล้ว สุ่มนำขึ้นมาแสดงตามความถี่ เป็นเปอร์เซ็นต์ ที่บราวเซอร์ของผู้เรียกใช้ไฟล์นั้น ได้ชมครั้งละ 1 รูปภาพ ซึ่งรูปภาพนี้สามารถกำหนดให้ลิงค์ไปที่เว็บไซต์ที่ต้องการได้ เพราะฉะนั้นเมื่อผู้เรียกกรายใดคลิกที่รูปภาพจะนำเขาไปยังอีกเว็บไซต์หรือไฟล์ที่กำหนดไว้ได้ ประโยชน์หลักๆของ Banner rotator ใช้สำหรับงานการโฆษณาเป็นส่วนใหญ่โดยจะนำ banner ต่างๆมาแสดงวนไปเรื่อยๆ

#### ประกอบและขั้นตอนส่วน

##### 1. สร้างไฟล์เก็บรายการรูปภาพและลิงค์ต่างๆ

ให้คุณ copy ข้อความต่อไปนี้แล้ว นำไป paste ลง notepad จากนั้น save ไฟล์ไว้ใน folder asp ตั้งชื่อว่า adlist.txt (width 410 และ height 60 คือความยาว-กว้างของรูปภาพ ส่วน border คือ ต้องการให้มีขอบสีลิงค์หรือไม่ ถ้าต้องการให้ใส่ 1 แทน 0 จะเห็นว่า มีตัวเลข 30 30 40 อยู่ได้สุดของรูปแต่ละรูปซึ่งก็คือเปอร์เซ็นต์กำหนดให้นำภาพขึ้นมาแสดง ถ้าค่ามากความถี่การแสดงก็จะมาก)

```
redirect /asp/gotoad.asp
```

```
width 410
```

```
height 60
```

```
border 0
```

\*

/hello.gif

http://127.0.0.1/asp/hello.asp

Click here to go to Hello World – คำอธิบายใช้ภาษาไทยก็ได้ 30

/content.gif

http://127.0.0.1/asp/showme.asp

Click here to go to Content Rotator 30

/home.gif

http://127.0.0.1/

Click here to go to HOME 40

2. ไฟล์ที่ทำหน้าที่นำผู้คลิกรูปภาพหรือแบนเนอร์ไปยังลิงค์ที่กำหนด

ให้คุณ copy code ต่อไปนี้แล้ว นำไป paste ลง notepad จากนั้น save ไฟล์ไว้ใน folder asp ตั้งชื่อว่า gotoad.asp

```
<% response.redirect(request.QueryString("url")) %>
```

3. สร้างไฟล์แสดงรูปภาพและลิงค์

ให้คุณ copy code ต่อไปนี้แล้ว นำไป paste ลง notepad จากนั้น save ไฟล์ไว้ใน folder asp ตั้งชื่อว่า showbanner.asp

```
<html>
```

```
<body>
```

```
<!-- เริ่มแทรกรูป Banner Content Rotator-->
```

```
<center>
```

```
<% Set Ad = Server.CreateObject("MSWC.Adrotator") %><%= Ad.GetAdvertisement
("/asp/adlist.txt") %>
```

```
</center>
```

```
</body>
```

```
</html>
```

4. สร้างไฟล์รูปภาพ

ให้ท่านสร้างไฟล์ .gif 3 ไฟล์ โดยมีขนาด 410x60 pixel ตั้งชื่อไฟล์ดังนี้คือ hello.gif, content.gif, home.gif และเก็บไฟล์เหล่านี้ไว้ใน root ของ server หรือถ้าท่านต้องการง่ายๆ ให้ copy ไฟล์ .gif ใดๆที่ท่านมีอยู่ให้เป็น 3 ไฟล์แล้วเปลี่ยนชื่อ จะทำให้ได้ไฟล์สำหรับทดลองเร็วขึ้น

## 5. การเรียกใช้งาน

เรียกไฟล์ showbanner.asp ในบราวเซอร์โดยพิมพ์ <http://127.0.0.1/asp/showbanner.asp> และ reload หรือ Refresh ไปเรื่อยๆ ผลที่ได้ จะลองคลิกดูก็ได้ครับ

### 2.7.4 การรับข้อมูลจากผู้ใช้

หลังจากได้เล่นๆกันมาตั้ง 2 บทแล้ว ในบทเรียนนี้เริ่มเอาจริงแล้วนะครับผู้ที่เป็นผู้ใช้ ASP มีวิธีหลักๆในการรับข้อมูลจากผู้ใช้อยู่ 4 วิธีหลักๆดังนี้ครับ

1. HTML Form
2. Query string
3. Cookies
4. Session

ซึ่งความจริงแล้วมีมากกว่า 4 วิธี แต่ที่นี่จะสอนเพียง 4 วิธีหลักๆก่อนขอตกลงใหม่ [ /query.htm ] ต่อไปนี่ถ้าพบสัญลักษณ์คล้ายๆแบบนี้จะหมายถึง ให้ copy code ที่อยู่ใต้เครื่องหมายนี้ แล้ว paste ลงใน notepad จากนั้น save ไว้ใน root ของ server โดยตั้งชื่อเป็น query.htm หรือถ้าเป็น [ /asp/query.asp ] ก็ให้ save ไว้ใน Folder asp โดยตั้งชื่อเป็น query.asp เป็นต้น

1. HTML Form ในด้านของ Form รับข้อมูลจากผู้ใช้ ก็ทำแบบธรรมดาไม่มีอะไรพิเศษเพิ่มเติมเข้ามาต่อไปนี้เป็นตัวอย่าง Form ที่จะส่งข้อมูลที่มีผู้ใช้กรอกไปให้กับไฟล์ getform.asp เพื่อนำไปแสดงผลหรือไปใช้ต่อไป ในตัวอย่างนี้เราจะให้คุณทำแบบแยกส่วนก่อน คือส่วนแรกเป็นแบบฟอร์ม ส่วนที่สองเป็นไฟล์รับข้อมูลที่ชื่อ getform.asp ต่อไปเมื่อคุณชำนาญแล้วจะสามารถทำไฟล์เดียวให้มีสองส่วนนี้อยู่ด้วยกันได้

```
[/input.htm]
```

```
<HTML>
```

```
<HEAD><TITLE>รับข้อมูล</TITLE></HEAD>
```

```
<BODY>
```

```
<H2>ตัวอย่าง Form รับข้อมูล</H2>
```

```
<P>
```

กรุณากรอกข้อมูลในแบบฟอร์มต่อไปนี้:

```
<FORM METHOD="POST" ACTION="/asp/getform.asp">
```

<P>

First Name: <INPUT NAME="fname" SIZE="48">

<P>Last Name: <INPUT NAME="lname" SIZE="48">

<P>

Title: <INPUT NAME="title" TYPE=RADIO VALUE="mr">Mr.

<INPUT NAME="title" TYPE=RADIO VALUE="ms">Ms.

<P><INPUT TYPE=SUBMIT><INPUT TYPE=RESET>

</FORM>

</BODY>

</HTML>

ต่อไปนี้เป็นไฟล์ getform.asp ซึ่งจะทำหน้าที่รับข้อมูลจาก form ในไฟล์ input.htm  
แล้วนำมาแสดงผลในรูปแบบต่างๆ

[ /asp/getform.asp ]

<%

รับข้อมูลจากแบบฟอร์มแล้วนำมาสร้างเป็นตัวแปรเก็บไว้

FirstN = request.form("fname")

LastN = request.form("lname")

UTitle = request.form("title")

%>

<HTML>

<BODY>

<H3>ตัวอย่างการรับข้อมูลกลับมาใส่แบบฟอร์ม</H3>

<FORM METHOD="POST" ACTION="/asp/getform.asp">

First Name: <INPUT NAME="fname" value="<%=request.form("fname")%>"  
SIZE="48">

<P>Last Name: <INPUT NAME="lname" value="<%=request.form("lname")%>"  
SIZE="48">

<P>

<% if request.form("title") = "mr" then %>

Title: <INPUT NAME="title" TYPE=RADIO CHECKED VALUE="mr">Mr.

```

<INPUT NAME="title" TYPE=RADIO VALUE="ms">Ms.
<% else %>
Title: <INPUT NAME="title" TYPE=RADIO VALUE="mr">Mr.
<INPUT NAME="title" TYPE=RADIO CHECKED VALUE="ms">Ms.
<% end if %>
<P><INPUT TYPE=SUBMIT><INPUT TYPE=RESET>
</FORM>
<hr>
<H3>ตัวอย่างการรับข้อมูลแล้วแสดงผลในทันที</H3>
First Name = <%=request.form("fname")%>

Last Name = <%=request.form("lname")%>

Title = <%=request.form("title")%>
<hr>
<H3>ตัวอย่างการรับข้อมูลจากตัวแปรที่ได้ตั้งไว้ในส่วนบนของไฟล์</H3>
First Name = <%=FirstN%>

Last Name = <%=LastN%>

Title = <%=UTitle%>
</BODY>
</HTML>

```

การใช้งานให้คุณเรียกไฟล์ input.htm ก่อน จากนั้นกรอกข้อมูลและ submit

## 2. Query string

```

[/query.htm]
<HTML>
<BODY>

```

ตัวอย่างการใช้ query string<br>

```

 click
ถึงคี่เพื่อดูผล

```

```

</BODY>

```

```

</HTML>

```

```

[/asp/getquery.asp]

```

```
<%
```

```
'รับข้อมูลจากแบบ Query String แล้วนำมาสร้างเป็นตัวแปรเก็บไว้
```

```
FirstN = request.querystring("firstname")
```

```
LastN = request.querystring("lastname")
```

```
UGender = request.querystring("gender")
```

```
%>
```

```
<HTML>
```

```
<BODY>
```

ตัวอย่างการใช้ query string<br>

```
<a
```

```
href="/asp/getquery.asp?firstname=<%=request.querystring("firstname")%>&lastname=
<%=request.querystring("lastname")%>&gender=<%=request.querystring("gender")%>">
```

```
click ที่นี่เพื่อดูผล
```

```
<H3>ตัวอย่างการรับข้อมูลแล้วแสดงผลในทันที</H3>
```

```
First Name = <%=request.querystring("firstname")%>

```

```
Last Name = <%=request.querystring("lastname")%>

```

```
Gender = <%=request.querystring("gender")%>
```

```
<hr>
```

```
<H3>ตัวอย่างการรับข้อมูลจากตัวแปรที่ได้ตั้งไว้ในส่วนบนของไฟล์</H3>
```

```
First Name = <%=FirstN%>

```

```
Last Name = <%=LastN%>

```

```
Gender = <%=UGender%>
```

```
</BODY>
```

```
</HTML>
```

### 3. Cookies

ก่อนการใช้ cookies เราก็มาสร้างไฟล์ที่ทำหน้าที่สร้าง Cookies กันก่อนดังนี้ครับ

```
[/asp/makecookies.asp]
```

```
<%
```

```
Response.Cookies("mycookies") = "valueisThaiTOP"
```

```
Response.Cookies("mycookies").Expires = Date+365
```

' ชื่อของ cookies จะตั้งเป็นอะไรก็ได้ที่ไม่ขัดกฎของ VB และค่าของ cookies จะรับมาจากที่อื่นก็ได้เช่น

```
' Response.Cookies ("mycookies") = request.form("somenam")
```

```
' หรือจะแทนด้วยตัวแปรก็ได้เช่น Response.Cookies ("mycookies") = date
```

```
%>
```

```
<html>
```

```
<body>
```

```
สร้าง cookies แล้วครับ
```

```
</body>
```

```
</html>
```

ไฟล์ด้านนี้จะทำหน้าที่สร้าง cookies ที่ชื่อ mycookies โดยมีค่าเป็น valueisThaiTOP ซึ่ง cookies นี้จะอยู่ในเครื่องของผู้เรียกไฟล์นี้เป็นเวลา 1 ปี หมายความว่า ถ้าเรียก cookies ที่ชื่อ mycookies ในภายหลังก็จะได้ค่าเป็น valueisThaiTOP สามารถเรียกใช้ได้เป็นเวลา 1 ปีก่อนที่ cookies จะหมดอายุ แล้วถ้าต้องการรับค่า cookies จะต้องทำแบบนี้ครับ

```
[/asp/showcookies.asp]
```

```
<%
```

```
showcookies = request.Cookies("mycookies")
```

```
%>
```

```
<html>
```

```
<body>
```

ค่าของ cookies ที่ตั้งไว้คือ <%=showcookies%> หรือจะเรียกโดยไม่ผ่านตัวแปรก็ได้ดังนี้  
ครับ <br>

```
ค่าของ cookies ที่ตั้งไว้คือ <%=request.Cookies("mycookies")%>
```

```
</body>
```

```
</html>
```

เมื่อสร้างได้ก็ลบได้โดยทำดังนี้

```
[/asp/clearcookies.asp]
```

```
<%
```

```
Response.Cookies("mycookies").Expires = now
```

```
%>
```

```
<html>
```

```
<body>
```

```
ลบ cookies แล้วครับ
```

```
</body>
```

```
</html>
```

ถ้าคุณเรียกไฟล์นี้ แล้วไปเรียกไฟล์ showcookies.asp คุณจะไม่เห็นค่าของ cookies เนื่องจากถูกลบแล้วถ้าต้องการแสดงค่าอีกให้เรียกไฟล์ makecookies.asp ก่อน 1 ครั้งแล้วจึงเรียก showcookies.asp

จะเห็นค่า cookies เป็น valueisThaiTOP รายละเอียดการใช้ที่สูงขึ้นอ่านได้จาก road map นะครับ

#### 4. Session

session คือค่าตัวแปรที่จะสร้างขึ้นให้กับผู้ใช้หรือผู้เรียกไฟล์ แล้วอยู่ไปตลอดจนกว่า session จะหมดอายุ(Time out) แนวทางการนำมาใช้ก็เช่น ถ้าผู้ใช้ยังไม่ได้อ่านไฟล์ใดที่เรากำหนด ก็ไม่สามารถไป อ่านไฟล์อื่นๆหรือ

download โปรแกรมที่เรากำหนดได้ หรือจะป้องกันการเรียกใช้ไฟล์โดยไม่ผ่านหน้าหลักของเว็บไซต์ หรือจะประยุกต์ใช้อื่นๆ ก็ได้ตามแต่ความสามารถและโอกาส ให้เข้าใจง่ายๆก็คือจำตัวแปรหรือค่าใดค่าหนึ่งหรือหลายๆค่าไว้ โดยสามารถเรียกค่านั้นมาใช้ได้ ซึ่งหลักการนี้ก็จะคล้ายๆกับการใช้ cookies

ในบทเรียนนี้จะยกตัวอย่างของการสร้าง session และการตรวจสอบ โดยมีเงื่อนไขว่าถ้าผู้ที่ต้องการ download ไฟล์ๆหนึ่งยังไม่อ่านข้อความหนึ่งจากเจ้าของ จะไม่อนุญาตให้ download ไฟล์นั้น ทำดังนี้

```
[/asp/creatsession.asp]
```

```
' สร้าง session ชื่อ candownload โดยมีค่าเท่ากับ yes
```

```
<% session("candownload") = "yes" %>
```

```
<html>
```

```
<body>
```

ถ้าสนใจ download อ่านข้อตกลงต่อไปนี่ก่อนครับ<br>

1. ถ้าพบข้อผิดพลาดกรุณาแจ้งเราที่ อีเมล someone@domain.com<br>

2. ....<br>

3. ....<br>

<br> ถ้าเห็นด้วยเลือกสิ่งนี้เพื่อ<a href="/asp/dl.asp">ไปหน้า download</a> ได้

```
</body>
```

```
</html>
```

```
[/asp/dl.asp]
```

```
<%
```

```
'ส่วนต่อไปนี้จะตรวจสอบว่า ถ้า session ที่ชื่อว่า candownload ไม่เท่ากับ (<>) yes จะส่งกลับไป
```

ที่

```
/asp/creatsession.asp เพื่อให้ไปอ่านข้อตกลงก่อน เพราะฉะนั้นถ้าใครเรียกไฟล์นี้โดยตรง
```

โดยไม่

```
'ได้อ่านไฟล์ creatsession.asp ก่อน จะต้องถูกส่งกลับไปอ่าน creatsession.asp เสมอไป
```

```
if session("candownload") <> "yes" then
```

```
'ต่อไปนี้เป็นคำสั่งการย้ายผู้เรียกไฟล์ให้ไปยังไฟล์หรือ url ที่กำหนดในวงเล็บ
```

```
response.redirect("/asp/creatsession.asp")
```

```
end if
```

```
%>
```

```
<html>
```

```
<body>
```

```
คุณได้อ่านข้อตกลงแล้ว เลือกคลิกเพื่อ download ไฟล์ได้เลยครับ

```

```
download somefile
```

```
</body>
```

```
</html>
```

### 2.7.5 ตรวจสอบประเภทและรุ่นของ บราวเซอร์

เนื่องจาก browser แต่ละยี่ห้อมีความสามารถไม่เหมือนกัน หรือยี่ห้อเดียวกันแต่ต่างรุ่นความสามารถก็ต่างกันผู้พัฒนาเว็บไซต์จึงต้องมีวิธีตรวจสอบยี่ห้อและรุ่น เพื่อที่จะส่งหน้าที่เหมาะสมไปให้ผู้ใช้ browser ต่างๆกันนั้นเช่น Netscape ไม่สามารถใช้ ActiveX ได้ แต่ IE ใช้ได้ ดังนั้นเมื่อมีใครเข้ามาในเว็บไซต์ของเรา จึงตั้งกำหนดหน้าที่เหมาะสมให้กับ Netscape (หน้าที่ไม่มี ActiveX) และส่งหน้าธรรมดาหรือหน้าที่มี ActiveX ให้กับ IE เป็นต้น ต่อไปนี้เป็นตัวอย่างการเรียกใช้และตรวจสอบ

```
[/asp/browsercap.asp]
```

```
<%
```

' เปิดเครื่องมือช่วยในการตรวจสอบ browser (เรียกใช้ ASP component) จะต้องเปิดแบบนี้ทุกครั้งถ้าต้องการใช้

' แต่อาจจะเปลี่ยน คำว่า bc ให้เป็นตัวแปรอื่นๆได้ แต่ถ้าเปลี่ยนแล้วส่วนอื่นๆที่เกี่ยวข้องจะต้องเปลี่ยนไปSet bc = Server.CreateObject("MSWC.BrowserType")

'ตั้งตัวแปร bname ให้แทนชื่อของ browser ผลที่ออกมาจะเป็น unknown, Netscape , IE เป็นต้น

```
bname=bc.browser
```

'ตั้งตัวแปร brver ให้แทน version ของ browser ผลที่ออกมาจะเป็นตัวเลขเช่น 1 2 3 4  
5brver= CINT(bc.version)

'ตัวแปรต่อไปนี้แสดงความสามารถของ browser ถ้ามีค่าเท่ากับ 1 แสดงว่าสามารถใช้ได้ แต่ถ้าออกมาเป็น 0 แสดงว่าไม่สนับสนุน

'เช่น ถ้า frames = 0 หมายความว่า browser ไม่สามารถใช้ Frame ได้ แต่ถ้าเท่ากับ 1 จะสามารถใช้ Frame ได้ เป็นต้น

```
ActiveX = bc.ActiveXControls
```

```
javaapplets = bc.javaapplets
```

```
javascript = bc.javascript
```

```
vbscript = bc.vbscript
```

```
frames = bc.frames
```

```
cookies = bc.cookies
```

```
backgroundsounds = bc.backgroundsounds
```

'platform คือระบบปฏิบัติการของผู้ใช้ ค่าที่ส่งออกมาจะเป็น win95 เป็นต้น

```
platform = bc.platform
```

' ต่อไปนี้เป็นตัวอย่างการประยุกต์ใช้ โดยมีหลักการคือ เราจะส่งผู้เข้าเว็บไซต์ของเราไปยังหน้าหลักที่เหมาะสมกับ

```
' browser ที่เขาใช้ โดยใช้ ภาษา vb script select - case เข้ามาช่วย
```

```
select case bname
```

```
case "IE"
```

'ถ้าเป็น IE ก็มาเลือก version ต่อแล้วจึงย้าย ผู้เยี่ยมชมเว็บไซต์ไปตาม version ของ browser ที่เขาใช้

```
select case brver
```

```
case "4"
```

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

```

case "3"
response.redirect("/asp/indexie3.asp")
case else
response.redirect("/asp/indexie.asp")
end select

```

```

case "Netscape"

```

'ในทำนองเดียวกันกับ IE ถ้าเป็น Netscape ก็ย้ายไปตาม version ที่เหมาะสมเช่นกัน

```

select case brver

```

```

case "4"
response.redirect("/asp/indexnet4.asp")

```

```

case "3"
response.redirect("/asp/indexnet3.asp")

```

```

case else
response.redirect("/asp/indexnet.asp")

```

```

end select

```

'ถ้าไม่ใช่ทั้ง IE หรือ Netscape ก็ย้ายไปที่ไฟล์ index.asp

```

case else
response.redirect("/asp/index.asp")

```

```

end select

```

```

%>

```

```

[/asp/activex_check.asp]

```

```

<%

```

'ตัวอย่างการตรวจสอบเฉพาะสิ่งที่ต้องการแล้ว redirect ไปยังไฟล์ที่กำหนด

```

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

```

```

ActiveX = bc.ActiveXControls

```

```

if ActiveX = 1 then

```

```

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

```

```

else

```

```

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

```

```

end if

```

```

%>
[/asp/activex_check2.asp]
<%
'ตัวอย่างการตรวจสอบเฉพาะ ActiveX แล้วแทรก ActiveX ในไฟล์ หรือไม่แทรกถ้า browser
ไม่สามารถใช้ได้

Set bc = Server.CreateObject("MSWC.BrowserType")
ActiveX = bc.ActiveXControls

%>
<html>
<body>
<% if ActiveX = 1 then %>
สามารถใช้ ActiveX ได้ ให้เริ่มแทรก ActiveX Code ได้เลย
<% else %>
ไม่สามารถใช้ ActiveX ได้ ให้ใช้วิธีอื่นในการทำงานแทน ActiveX
<% end if %>
</body>
</html>

```

### 2.7.6 ไฟล์ Global.asa

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

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

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

'เปิดไฟล์ Text สำหรับบันทึกจำนวนผู้เข้าเยี่ยมชมเว็บไซต์ การเรียกใช้มีคำอธิบายอยู่ด้านล่าง  
ในส่วนของ Sub Application\_OnStart นี้สามารถใส่ application อื่นๆ ได้อีกตามต้องการ

แต่สิ่งที่ใส่ในส่วนนี้ควรเป็น โปรแกรมที่ต้องการ share หรือใช้ตัวแปรนี้ร่วมกันทุกคนที่เข้ามาในเว็บไซต์

เช่น จำนวน visitor ในตัวอย่างนี้ หรืออาจจะนำมานับจำนวนคนที่กำลัง connect กับเว็บไซต์ของคุณ'อยู่ก็ได้

```
Sub Application_OnStart
```

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

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

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

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

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

```
End Sub
```

'ถ้าเปิด Application ใดๆใน Application\_OnStart ควรต้องมาปิดในนี้ด้วย

```
Sub Application_OnEnd
```

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

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

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

```
Out.Close
```

```
End Sub
```

'Sub Session\_OnStart นี้ใช้สำหรับส่งหรือบันทึกตัวแปรไว้เพื่อให้ผู้ใช้เว็บไซต์แต่ละคน

```
Sub Session_OnStart
```

'ดูตัวอย่างการนำ session สองตัวแปรนี้ไปใช้ในไฟล์ .asp ได้ที่ตัวอย่างด้านล่าง

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

```
link=#0000ee vlink=#551A8B
```

```
alink=#ff0000"
```

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

'เขียนจำนวนผู้ใช้เว็บเพิ่มอีก 1 คน

```
Application.Lock
```

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

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

```
Application.Unlock
```

```

Session("VisitorID") = t_visitors
If t_visitors MOD 15 = 0 Then
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
Set Out= FileObject.CreateTextFile (Application("VisitorCountFilename"), True, False)
Application.Lock
Out.WriteLine(t_visitors)
Application.Unlock
End If
End Sub
'ลบตัวแปรที่กำหนดไว้ใน Sub Session_OnStart
Sub Session_OnEnd
Session("PageTheme") = Nothing
Session("PageTitle") = Nothing
End Sub
</SCRIPT>

```

การเรียกใช้ตัวแปรและ Application ที่กำหนดไว้ใน global.asa

1. Application ในการนับจำนวนผู้ใช้เว็บไซต์การใช้คุณจะต้อง สร้างไฟล์ text ว่างๆชื่อ visitors.txt ไว้ใน /cgi-bin ก่อนจากนั้นให้เรียกใช้เว็บไซต์จากส่วนใดก็ได้ที่มีนามสกุลไฟล์ .asp ซึ่งจะ ทำให้ webserver อ่านไฟล์ global.asa จากนั้น ASP จะเขียนจำนวนตัวเลขไว้ในไฟล์ text ดังกล่าวใน ครั้งต่อไปที่ท่านเรียกไฟล์ .asp ใดๆในเว็บไซต์อีก จำนวนตัวเลขที่อยู่ในไฟล์นี้ก็จะเพิ่มขึ้นมาอีก 1 ส่วนการนำตัวเลขในไฟล์ visit rs.txt มาแสดงในเว็บไซต์ ก็เพียงแค่แทรกcode ต่อไปนี้ในไฟล์.asp ใน ตำแหน่งที่ต้องการ You are visitors No. <%=Application("visitors")%> Since mm/dd/yy

2. การนำ session PageTheme และ Session PageTitle ไปใช้ในไฟล์ .asp ใดๆในเว็บไซต์ให้ คุณสร้างไฟล์ .asp อะไรก็ได้ขึ้นมา 1 ไฟล์ แต่ในส่วนหัวของไฟล์ให้ใส่ code ต่อไปนี้ไว้จากนั้นลอง เรียกไฟล์เพื่อดูผล ไฟล์ทุกไฟล์ในเว็บไซต์ ถ้าใส่ code นี้ไว้จะทำให้มี Title และรูปแบบสีของลิงค์ ตัวอักษร . ภาพพื้นหลังเหมือนกัน ถ้าต้องการเปลี่ยนส่วนใดส่วนหนึ่งก็เพียงไปเปลี่ยนที่ไฟล์ global.asa เท่านั้น แล้ว restart IIS หรือ PWS ใหม่ รูปแบบก็จะเปลี่ยนไปทันที โดยไม่ต้องไปเสียเวลา แก้ไฟล์แต่ละไฟล์เหมือนกับที่ใช้ .htm หรือ .html ธรรมดา

```
<html>
```

```
<head>
```

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

## 2.7.7 การเขียนข้อมูลลงไฟล์ Text ด้วย ASP

สำหรับบทนี้มีตัวอย่างให้ลองของจริง แต่ก่อนเรียกไฟล์ 2 ไฟล์นี้ให้คุณสร้างไฟล์ text ว่างๆ ชื่อ textthere.txt ไว้ใน /asp ก่อนนะครั้นการนำไปใช้งานแล้วแต่จะประยุกต์ใช้

[ /asp/write2text.asp ]

```
<% if request.form("message") = "" then %>
<html>
<body>
<form action="write2text.asp" method="post">
กรอกข้อมูลที่คุณต้องการเขียนลงไฟล์ text

Name: <input type=text name=message size=30>
<input type=submit value=Writ2Text>
</form>
</body>
</html>
<%
else
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
MsgFile = Server.MapPath ("textthere.txt")
Set OutStream= FileObject.CreateTextFile (MsgFile, True)
OutStream.WriteLine Request.Form("message")
Set OutStream = Nothing
Set MessageStr =Nothing
%>
<html>
<body>
เขียนร้อยแล้วครับ เรียกไฟล์ readall.asp ดูได้เลยครับ
</body>
```

```

</html>
<% end if %>
[/asp/readall.asp]
<html>
<body>
อ่าน ไฟล์ text<hr>
<%
Set FileObject = Server.CreateObject("Scripting.FileSystemObject")
MsgFile = Server.MapPath ("textthere.txt")
On Error Resume Next
Set InStream= FileObject.OpenTextFile (MSGFile, 1, False, False)
Response.Write InStream.ReadALL & "
"
Set InStream=Nothing
%>
</body>
</html>

```

## 2.7.8 การใช้ ASP เรียกใช้ฐานข้อมูล

คุณสามารถสร้างไฟล์ฐานข้อมูลใดๆที่สนับสนุน MS ODBC แล้วนำไปไว้ใน web server ของคุณ

จากนั้นทำการ Add DSN แล่นี่คุณก็พร้อมที่จะ add edit delete view search ฐานข้อมูลของคุณได้ด้วย ASP

ในบทเรียนนี้ผมจะสร้างไฟล์ asp อย่างง่ายที่ทำหน้าที่ทั้ง 5 ประการข้างต้นให้ไว้เป็นตัวอย่าง คุณสามารถนำไปดัดแปลงเพื่อประยุกต์ใช้ในงานของคุณได้ตามความต้องการ เพราะงานทั้ง 5 นี้ถือเป็นพื้นฐานของการใช้ฐานข้อมูลทุกชนิด

ขั้นตอนแรกให้คุณสร้างไฟล์ชื่อ email.mdb โดยใช้ MS Access ภายในมี 1 table ชื่อ email เช่นกัน

มี 3 field ดังนี้คือ id ชนิด Autonumber

name ชนิด text

email ชนิด text

date ชนิด date/time

เมื่อสร้างเสร็จแล้วให้นำไปไว้ใน /database จากนั้น Add DSN โดยใช้ชื่อ DSN เป็น "themail"

งานที่ 1 "Add" นำข้อมูลจากแบบฟอร์มใส่ในฐานข้อมูล

งานที่ 2 "Edit" แก้ไขข้อมูลในฐานข้อมูล

งานที่ 3 "Delete" ลบข้อมูลในฐานข้อมูล

งานที่ 4 "View" ดูข้อมูลในฐานข้อมูล

งานที่ 5 "Search" ค้นหาข้อมูลในฐานข้อมูล

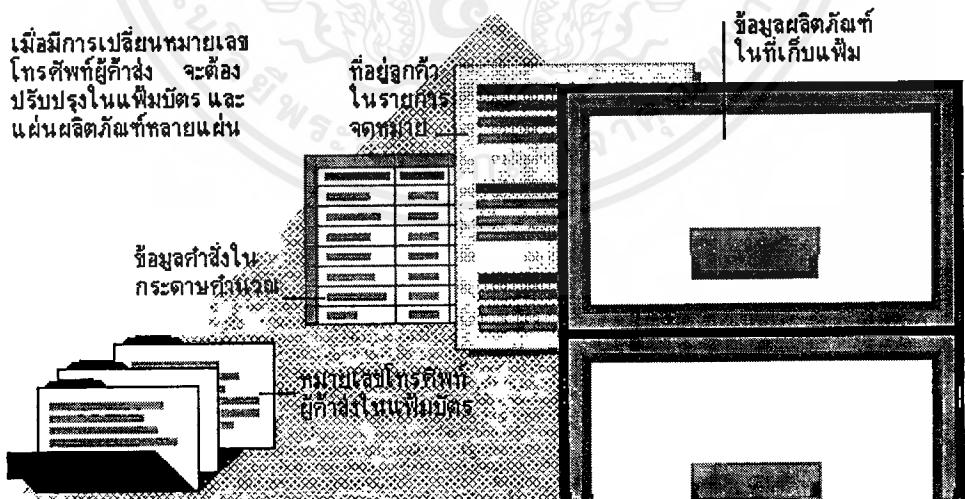
ข้อมูลการใช้ รายละเอียดต่างๆ อ่านเพิ่มเติมได้จาก Road map นะครับ ผมก็เขียนขึ้นมาโดยใช้

Road map อ้างอิง ค่อยๆศึกษาไปไม่ยากหรอกครับ

## 2.8 ข้อมูล(DATA BASE)

### 2.8.1 ฐานข้อมูลคืออะไรและทำงานอย่างไร

ฐานข้อมูลคือ การรวบรวมข้อมูลที่มีความสัมพันธ์กัน โดยอยู่ภายใต้หัวข้อหรือจุดประสงค์อย่างใดอย่างหนึ่ง เช่น การติดตามคำสั่งของลูกค้า หรือการรวบรวมข้อมูลเกี่ยวกับการดูแลเรื่องคนตรี ถ้าฐานข้อมูลของคุณไม่ได้เก็บไว้ในคอมพิวเตอร์ หรือมีเพียงบางส่วนที่เก็บไว้ คุณอาจจะต้องติดตามข้อมูลจากหลายแหล่งซึ่งคุณจะต้องรวบรวมและจัดการกับข้อมูลเหล่านั้นด้วยตนเอง



รูปที่ 2.10

## เกี่ยวกับการออกแบบฐานข้อมูล

ก่อนที่คุณจะใช้ Microsoft Access สร้างตาราง แบบฟอร์ม และวัตถุอื่นที่จะสร้างฐานข้อมูลของคุณ เวลาเป็นสิ่งสำคัญในการออกแบบฐานข้อมูลของคุณ การออกแบบฐานข้อมูลที่ดียิ่งคือ หลักสำคัญ เพื่อสร้างฐานข้อมูลที่คุณต้องการทำให้เกิดประโยชน์ มีความถูกต้อง และประสิทธิภาพ สำหรับภาพรวมของอะไรที่จะเป็นฐานข้อมูลและทำงานอย่างไร

### ขั้นตอนในการออกแบบฐานข้อมูล

1. กำหนดจุดประสงค์ฐานข้อมูลของคุณ
2. กำหนดตารางที่คุณต้องการในฐานข้อมูล
3. กำหนดเขตข้อมูลที่คุณต้องการในตาราง
4. ระบุเขตข้อมูลด้วยค่าไม่ซ้ำซ้อน
5. กำหนดความสัมพันธ์ระหว่างตารางต่างๆ
6. แก้ไขการออกแบบของคุณ
7. เพิ่มข้อมูล และสร้างวัตถุฐานข้อมูลอื่น
8. ใช้ Microsoft Access วิเคราะห์เครื่องมือ

#### 1. กำหนดวัตถุประสงค์ของฐานข้อมูลของคุณ

ขั้นแรกในการออกแบบฐานข้อมูล Microsoft Access จะต้องกำหนดจุดประสงค์ของฐานข้อมูล และมีใช้ฐานข้อมูลอย่างไร คุณจำเป็นต้องรู้ว่าข้อมูลอะไรที่คุณต้องการจากฐานข้อมูล จากนั้นคุณสามารถกำหนดเนื้อเรื่องอะไรที่คุณจำเป็นต้องจัดเก็บจริงเกี่ยวกับ (ตาราง) และความจริงอะไรที่คุณจำเป็นต้องจัดเก็บเกี่ยวกับแต่ละเนื้อเรื่อง (เขตข้อมูลในตาราง)

ให้ปรึกษากับผู้ที่ใช้ฐานข้อมูล ระดมสมอง เกี่ยวกับคำถามที่คุณอยากให้ฐานข้อมูลตอบ ร่าง รายงานที่คุณอยากให้ฐานข้อมูลสร้าง รวมแบบฟอร์มที่คุณใช้ระเบียบข้อมูลปัจจุบันของคุณ ตรวจสอบฐานข้อมูลที่ออกแบบอย่างดีคล้ายกับข้อมูลที่คุณกำลังออกแบบอยู่

#### 2. กำหนดตารางต่างๆ ที่คุณต้องการ

การกำหนดตารางสามารถเป็นขั้นตอนที่ลึกลับที่สุดในกระบวนการการออกแบบฐานข้อมูล นั่นเป็นเพราะว่าผลลัพธ์ที่คุณต้องการจากฐานข้อมูลของคุณ — รายงานที่คุณต้องพิมพ์ แบบฟอร์มที่คุณต้องการใช้ คำถามที่คุณต้องการให้ตอบ — ไม่จำเป็นต้องให้เงื่อนไขเกี่ยวกับโครงสร้างของตารางที่สร้างคุณไม่จำเป็นต้องออกแบบตารางของคุณโดยการ ใช้ Microsoft Access จริงๆ แล้วให้ร่างและทำงานการออกแบบของคุณบนกระดาษก่อน เมื่อคุณออกแบบตารางของคุณ ให้แบ่งเป็นข้อมูลส่วนๆ โดยให้การเก็บหลักการสำคัญของการออกแบบไว้

2.1 ตารางไม่ควรมีข้อมูลแบบคัดลอก และไม่ควรเป็นข้อมูลคัดลอกระหว่างตาราง

2.2 เมื่อมีการจัดเก็บข้อมูลแต่ละส่วนไว้ในตารางเดียวกัน ให้คุณปรับปรุงข้อมูลในที่หนึ่งวิธีนี้เป็นวิธีที่มีประสิทธิภาพมากขึ้นและยังเป็นการกำจัดความเป็นไปได้ของการป้อนซ้ำในข้อมูลที่แตกต่างกัน เช่น คุณต้องการจัดเก็บที่อยู่และหมายเลขโทรศัพท์ของลูกค้าแต่ละคนไว้ในหนึ่งตาราง

2.3 แต่ละตารางควรมีข้อมูลในหนึ่งเนื้อเรื่อง

2.4 เมื่อแต่ละตารางมีความจริงในหนึ่งเรื่องเท่านั้น คุณสามารถเก็บแต่ละหัวเรื่องของข้อมูลได้อย่างอิสระจากหัวเรื่องอื่น เช่น ให้คุณจัดเก็บที่อยู่ของลูกค้าในตารางที่ต่างกันจากใบสั่งซื้อของลูกค้า เพื่อคุณสามารถจะลบใบสั่งซื้อหนึ่งและยังคงเก็บข้อมูลของลูกค้าไว้

### 3. กำหนดเขตข้อมูลต่างๆ ที่คุณต้องใช้

แต่ละตารางมีข้อมูลเกี่ยวกับเนื้อเรื่องเดียวกัน และแต่ละ เขตข้อมูล ในตารางมีข้อมูลส่วนบุคคลที่เป็นจริง เกี่ยวกับเนื้อเรื่องของตารางนั้น เช่น ตารางลูกค้าอาจรวมชื่อบริษัท ที่อยู่ เมือง รัฐ และเขตหมายเลขโทรศัพท์ เมื่อกำลังร่าง เขตข้อมูลในแต่ละตาราง ให้ทำตามคำแนะนำนี้

3.1 แต่ละเขตข้อมูลจะมีความสัมพันธ์กับเนื้อเรื่องของตาราง

3.2 ไม่รวมข้อมูลสืบทอด หรือข้อมูลคำนวณ (ข้อมูลที่เป็นผลลัพธ์ของนิพจน์)

3.3 รวมข้อมูลทั้งหมดที่คุณต้องการ

3.4 จัดเก็บข้อมูลในส่วนที่เป็นตรรกะเล็กที่สุด (เช่น ชื่อแรก และชื่อสุดท้าย มากกว่าชื่อ)

### 4. แยกแยะเขตข้อมูลต่างๆ ด้วยค่าที่ไม่ซ้ำกัน

ในการให้ Microsoft Access เชื่อมต่อกับข้อมูลที่ได้จัดเก็บไว้ในตารางที่แยกส่วน เช่น เชื่อมต่อกับลูกค้าด้วยการสั่งซื้อของลูกค้าทั้งหมด แต่ละตารางในฐานะข้อมูลจะต้องรวมเขตข้อมูล หรือตั้งค่าเขตข้อมูลที่ระบุระเบียบส่วนบุคคลในแต่ละตาราง เช่น มีการเรียกเขตข้อมูลนั้น หรือตั้งค่าเขตข้อมูลว่าคีย์หลัก

### 5. กำหนดความสัมพันธ์ระหว่างตาราง

ขณะที่คุณได้แบ่งข้อมูลของคุณลงในตาราง และระบุเขตข้อมูลคีย์หลัก คุณจำเป็นจะต้องบอก Microsoft Access เกี่ยวกับการนำข้อมูลที่มีความสัมพันธ์กันกลับมาใช้อีกครั้งในวิธีที่เป็นความหมายเดียวกัน วิธีนี้คุณกำหนดความสัมพันธ์ระหว่างตาราง

คุณจะพบว่า เป็นมุมมองที่มีประโยชน์ต่อความสัมพันธ์ในฐานะข้อมูลที่ออกแบบไว้เป็นอย่างดีคงมีอยู่ ในการดูความสัมพันธ์ของฐานข้อมูลตัวอย่างใน Northwind ให้เปิด Northwind.mdb และคลิกความสัมพันธ์บนเมนูเครื่องมือ

### 6. จัดเกลาการออกแบบ

หลังจากที่คุณได้ออกแบบตาราง เขตข้อมูล และความสัมพันธ์ที่คุณต้องการ เป็นเวลาที่ต้องศึกษาการออกแบบและป้องกันความบกพร่องที่อาจยังมีอยู่ เป็นวิธีง่ายต่อการเปลี่ยนแปลงการออกแบบฐานข้อมูลของคุณ ในขณะที่มากกว่าการที่คุณได้เติมข้อมูลในตารางแล้ว

ให้ใช้ Microsoft Access สร้างตารางของคุณ กำหนดความสัมพันธ์ระหว่างตาราง และป้อนระเบียบฐานข้อมูลบางตัวลงในแต่ละตาราง เพื่อว่าคุณสามารถใช้ฐานข้อมูลให้คำตอบที่คุณต้องการ สร้างแบบร่างอย่างหยาบของแบบฟอร์ม และรายงานของคุณ และดูว่าจะมีการแสดงข้อมูลที่คิดไว้หรือไม่ ค้นหาข้อมูลซ้ำที่ไม่จำเป็นและตัดข้อมูลเหล่านั้นทิ้ง

## 7. ป้อนข้อมูล และสร้างวัตถุฐานข้อมูลอื่นๆ

เมื่อคุณต้องการ โครงสร้างของตารางตรงกับเป้าหมายการออกแบบที่ได้อธิบายไว้นี้ ให้คุณทำต่อไปและเพิ่มข้อมูลที่มีอยู่ของคุณที่ตาราง คุณสามารถสร้างได้ทุกการสอบถาม, แบบฟอร์ม, รายงาน, แมโคร, และ โมดูล (modules) ที่คุณต้องการ

## 8. ใช้เครื่องมือการวิเคราะห์ของ Microsoft Access

Microsoft Access จะรวมเครื่องมือสองอย่างที่สามารถช่วยให้คุณแก้ไขการออกแบบฐานข้อมูลของคุณ ตาราง วิเคราะห์ตัวช่วยสร้าง สามารถวิเคราะห์การออกแบบของตารางในหนึ่งครั้ง สามารถเสนอโครงสร้างตารางใหม่\_และความสัมพันธ์ ถ้าเหมาะสม และสามารถสร้างโครงสร้างตารางใหม่ลงในตารางที่สัมพันธ์กัน ถ้ามีเหตุผล

ตัววิเคราะห์ประสิทธิภาพ สามารถวิเคราะห์ฐานข้อมูลทั้งหมด สร้างคำแนะนำ และข้อเสนอแนะในการปรับปรุง ตัวช่วยสร้างยังสามารถสนับสนุนคำแนะนำและข้อเสนอแนะได้อีกด้วย

### 2.8.2 การสร้างฐานข้อมูล

Microsoft Access จะจัดการให้มีสองวิธีในการสร้างฐานข้อมูล คุณสามารถสร้างฐานข้อมูลเปล่าและเพิ่มตาราง ฟอร์ม รายงาน และวัตถุอื่นได้ทีหลัง นี่เป็นวิธีที่ยืดหยุ่นที่สุด แต่เป็นวิธีที่ต้องการให้คุณกำหนดองค์ประกอบแต่ละฐานข้อมูลให้แยกออกจากกัน หรือคุณสามารถใช้ตัวช่วยสร้างฐานข้อมูลเพื่อการสร้างในหนึ่งการปฏิบัติการที่ต้องการตาราง ฟอร์ม และรายงานสำหรับชนิดของฐานข้อมูลที่คุณเลือกนี้เป็นทางที่ง่ายที่สุดในการเริ่มสร้างฐานข้อมูลของคุณ ในทางเดียวกัน คุณสามารถปรับเปลี่ยนและต่อฐานข้อมูลของคุณได้ทุกเวลาหลังจากที่ฐานข้อมูลได้สร้างแล้ว

คุณต้องการจะทำสิ่งใด

#### 1. สร้างฐานข้อมูลด้วยการใช้ตัวช่วยสร้างฐานข้อมูล

1.1 เมื่อ Microsoft Access เริ่มต้นเป็นครั้งแรก กล่องโต้ตอบจะแสดงด้วยตัวเลือกในการสร้างฐานข้อมูลใหม่ หรือเปิดฐานข้อมูลที่มีอยู่แล้วอย่างอัตโนมัติ ถ้ากล่องโต้ตอบนี้ได้แสดงคลิก ตัวช่วยสร้างฐานข้อมูล และคลิกตกลง

ถ้าคุณมีฐานข้อมูลที่เปิดอยู่แล้ว หรือกล่องโต้ตอบที่ปิดจะแสดง เมื่อ เริ่มต้น

Microsoft Access คลิก สร้างฐานข้อมูล บนแถบเครื่องมือ

1.2 บนแท็บ ฐานข้อมูล คลิกสองครั้งที่ไอคอนสำหรับชนิดของฐานข้อมูลที่คุณต้องการจะสร้าง

### 1.3 ระบุชื่อและตำแหน่งสำหรับฐานข้อมูล

1.4 คลิก สร้าง เพื่อเริ่มกำหนดฐานข้อมูลใหม่ของคุณ

## 2.สร้างฐานข้อมูลโดยไม่ใช้ตัวช่วยสร้างฐานข้อมูล

2.1 เมื่อ Microsoft Access เริ่มต้นเป็นครั้งแรก กล่องโต้ตอบจะแสดงด้วยตัวเลือกในการสร้างฐานข้อมูลใหม่ หรือเปิดฐานข้อมูลที่มีอยู่แล้วอย่างอัตโนมัติ ถ้ากล่องโต้ตอบนี้จะแสดงคลิกฐานข้อมูลเปล่า และคลิก ตกลง

ถ้าคุณมีฐานข้อมูลที่เปิดอยู่แล้ว หรือกล่องโต้ตอบที่เปิดจะแสดงเมื่อ เริ่มต้น Microsoft Access คลิก สร้างฐานข้อมูล บนแถบเครื่องมือ และคลิกสองครั้งที่ไอคอนฐานข้อมูลเปล่าบน แท็บ ทั่วไป

2.2 ระบุชื่อและตำแหน่งสำหรับฐานข้อมูล และคลิก สร้างหลังจากคุณสร้างฐานข้อมูลเปล่า คุณจะต้องแสดงการเพิ่มขึ้นตอนในการกำหนดวัตถุประสงค์ที่จะตกแต่งฐานข้อมูลของคุณ

3.ฐานข้อมูล: คืออะไรและทำงานอย่างไร(2.8.1)

4.เกี่ยวกับการออกแบบฐานข้อมูล

### 2.8.3 การแสดง หรือการเปลี่ยนแปลงคุณสมบัติทั่วไปของวัตถุฐานข้อมูล

1.ในหน้าต่างฐานข้อมูล คลิกวัตถุที่คุณต้องการเปลี่ยนแปลงคุณสมบัติ

2.คลิก คุณสมบัติ (คุณสมบัติ (เมนูมุมมอง)แสดงแผ่นคุณสมบัติสำหรับรายการที่เลือก เช่นเขตข้อมูลตาราง หรือตัวควบคุมฟอร์ม ถ้าไม่มีสิ่งใดถูกเลือก ให้แสดงแผ่นคุณสมบัติของวัตถุที่ใช้อยู่บนแถบเครื่องมือ

### 2.8.4 การเปลี่ยนชื่อ การลบ การคัดลอก หรือ การย้ายฐานข้อมูล

การเปลี่ยนชื่อฐานข้อมูลใน Microsoft Windows 95 หรือ Microsoft Windows NT Workstation 4.0

การเปลี่ยนชื่อฐานข้อมูล

1.ปิดฐานข้อมูลนั้น ในสภาพแวดล้อมหลายผู้ใช้ ให้แน่ใจว่าผู้ใช้ทุกคนได้ปิดฐานข้อมูลนั้นจากนั้น

2.ใน My Computer หรือ Windows Explorer ให้หาเพิ่มฐานข้อมูลนั้น

3.คลิกที่ชื่อของแฟ้ม (ไม่ใช่ที่ไอคอน) จากนั้นคลิกอีกครั้งเพื่อแก้ไขชื่อ

4.พิมพ์ชื่อใหม่เข้าไป ให้รวมนามสกุล .mdb เข้าไปด้วย เมื่อคุณได้ตั้งค่าให้ Microsoft Windows แสดงนามสกุลแฟ้มแบบ MS-DOS เท่านั้น จากนั้นกด ENTER ชื่อแฟ้มสามารถใช้อักขระได้มากถึง 255 อักขระ รวมทั้งช่องว่างด้วย

หมายเหตุ เมื่อคุณเปลี่ยนชื่อให้ฐานข้อมูล รหัสในฐานข้อมูลจะถูกแปลไปด้วย การแปลรหัสใหม่อีกครั้ง และการบันทึกโมดูล (Module) ทั้งหมดในสภาพที่แปลจากนั้น ให้เปิดฐานข้อมูลนั้น เปิดโมดูลในมุมมองการออกแบบ จากนั้นคลิก คอมไพล์โมดูลทั้งหมด บนเมนู แก์จุดบกพร่อง จากนั้นคลิก คอมไพล์และบันทึกโมดูลทั้งหมด บนเมนู แก์จุดบกพร่อง

สิ่งสำคัญ ถ้าคุณเปลี่ยนชื่อ ลบ คัดลอก หรือย้ายฐานข้อมูลหนึ่ง และฐานข้อมูลอื่นๆ ที่มีวัตถุเชื่อมจากฐานข้อมูลนั้นอยู่ ให้ใช้ ตัวจัดการเชื่อมตาราง ในฐานข้อมูลอื่นๆ เหล่านั้นเพื่อปรับปรุงข้อมูลที่เชื่อม มิฉะนั้น Microsoft Access จะสร้างข้อผิดพลาด เมื่อคุณพยายามใช้วัตถุเชื่อมเหล่านั้น สำหรับข้อมูลเพิ่มเติม

### 2.8.5 ความรู้และทฤษฎีพื้นฐานโมเดลเชิงสัมพันธ์ (Relation Model)

โมเดลเชิงสัมพันธ์เป็นโมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกเก็บด้วยระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ซึ่งเป็นผลงานของ ดร.คอดด์ เสนอผลงานวิจัยให้ชาวโลกรู้จักในปี พ.ศ. 2513 โดยมีแนวความคิดในการพัฒนามาจากทฤษฎีทางคณิตศาสตร์ นั่นคือคิดว่าการแก้ปัญหาถ้าใช้ทฤษฎีทางคณิตศาสตร์ที่ถูกต้องแล้วไม่ว่าจะหาคำตอบมาจากวิธีการโดยอ้อมจะต้องได้ผลลัพธ์ที่เหมือนกัน ในปัจจุบันนี้มีการนำโมเดลนี้มาใช้กับเครื่องระดับเมนเฟรมลงไปถึงเครื่องระดับไมโครคอมพิวเตอร์ด้วย และก็เป็นที่ยอมรับกันแล้วว่า บรรดาผู้ใช้ระบบฐานข้อมูลจะมีความคุ้นเคยกับโมเดลเชิงสัมพันธ์นี้มากกว่าอีก 2 โมเดล คือ โมเดลเชิงแตกสาขา (Hierarchy Model) และโมเดลเชิงโครงข่าย (Network Model) ที่มีมาก่อนหน้านี้

#### การเปรียบเทียบระหว่างโมเดลเชิงสัมพันธ์กับโมเดลอื่น ๆ

นอกเหนือจากความแพร่หลายของโมเดลเชิงสัมพันธ์นี้แล้ว ข้อดีของโมเดลเชิงสัมพันธ์ที่มีมากกว่าอีก 2 โมเดลมีดังนี้

1. โมเดลเชิงสัมพันธ์เป็น โมเดลที่สร้างความเข้าใจได้ง่ายกว่า เพราะภาพลักษณ์ของข้อมูลที่เก็บโดยโมเดลเชิงสัมพันธ์จะมาจากมุมมองของผู้ใช้ ซึ่งจะมีความซับซ้อนน้อยกว่าภาพลักษณ์ของข้อมูลที่เก็บ โดยอีก 2 โมเดล

2. ระบบส่วนใหญ่ที่ใช้โมเดลเชิงสัมพันธ์นี้ มักจะมีเครื่องมือที่ช่วยให้ผู้ใช้ สามารถจัดการกับข้อมูลที่จัดเก็บด้วยโมเดลแบบอื่น

3. โมเดลเชิงสัมพันธ์นี้ มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถค้นพบปัญหา ที่เกิดขึ้นในการออกแบบระบบฐานข้อมูลได้โดยง่ายและยังง่ายในการแก้ไขการออกแบบที่ผิดพลาดนั้นด้วย

4. โมเดลเชิงสัมพันธ์ เป็นโมเดลที่มีความสอดคล้อง กับหลักการฐานข้อมูล ผู้ใช้ไม่ต้องพะวงกับรายละเอียดของการจัดเก็บข้อมูลเหมือนกับการจัดข้อมูลของโมเดลอื่น

5. ภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ (ภาษา SQL ; Structur Query Language) เป็นภาษาแบบเซต โอเรียนต์ (set oriented) ซึ่งจะต่างกับภาษาที่ใช้

ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลของโมเดลอื่นที่เป็นภาษาแบบ record-at-a-time

แม้ว่าโมเดลเชิงสัมพันธ์จะมีข้อดีหลายประการดังได้กล่าวไปแล้ว แต่ในปัจจุบันก็ยังมีจุดอ่อนที่มีการอ้างอิงถึงเสมอ คือ ระบบจัดการฐานข้อมูลแบบโมเดลเชิงสัมพันธ์นี้มักจะมีประสิทธิภาพในการใช้งานสู่อีก 2 โมเดลไม่ได้ โดยเฉพาะในการประยุกต์ใช้งานขององค์กรขนาดใหญ่ จุดอ่อนนี้ก็ได้มีการแย้งกลับมาในแง่ที่ว่า โมเดลเชิงสัมพันธ์ก็ยังมีจำนวนระดับชั้นที่ได้พัฒนาไปแล้วน้อยกว่าอีก 2 โมเดล ดังนั้นหากต้องการเปรียบเทียบที่ระดับจำนวนชั้นการพัฒนาที่เท่ากันจึงจะสมเหตุผล

### ศัพท์เฉพาะของโมเดลเชิงสัมพันธ์

ในหัวข้อนี้จะกล่าวถึง โมเดลเชิงสัมพันธ์ โดยกำหนดนิยามและกล่าวถึงคำศัพท์ต่าง ๆ ที่เกี่ยวข้องกับ โมเดลนี้

จากการเก็บข้อมูลที่เก็บด้วย โมเดลเชิงสัมพันธ์จะถูกเก็บไว้ในตารางที่ถูกเรียกว่า “รีเลชัน” โดยที่รีเลชันทุกรีเลชันอาจแสดงในรูปของตาราง แต่ตารางบางตารางอาจไม่เป็นรีเลชันก็ได้ ตารางที่มีลักษณะเป็นรีเลชันต้องมีคุณสมบัติดังนี้

1. แต่ละช่องของตารางจะบรรจุข้อมูลได้เพียงค่าเดียว
2. ชื่อหัวข้อในแต่ละคอลัมน์มีความแตกต่างกัน อันเป็นชื่อของแอตทริบิวของเอนทิตี
3. ค่าของข้อมูลในแต่ละคอลัมน์คือ ค่าของแอตทริบิวของเอนทิตี
4. การเรียงลำดับคอลัมน์ไม่ถือว่ามีความสำคัญ
5. ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
6. การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

ตารางที่มีคุณสมบัติดังกล่าวจะเรียกว่ารีเลชัน ดังนั้นเราจะได้นิยามของฐานข้อมูลเชิงสัมพันธ์ คือ ฐานข้อมูลที่เกิดจากการรวมรีเลชันต่าง ๆ ที่มีความสัมพันธ์ระหว่างกัน

เราจะเรียกข้อมูลแต่ละแถวในเนวอนของรีเลชันว่า ทัพเพิล (Tuple) และเรียกข้อมูลในแต่ละแถวในแนวตั้งหรือแนวคอลัมน์ว่า แอททริบิว (Attribute) โดยที่คำว่า คีย์ (key) จะหมายถึงข้อมูลที่เกิดจากแอตทริบิว 1 ตัวหรือหลายตัวก็ได้

แต่ละรีเลชันจะต้องมีสิ่งที่เรียกว่า คีย์หลัก (Primary Key) คือ คีย์คู่แข่ง (Candidate Key) ตัวที่เลือกขึ้นมา ซึ่งมักจะเลือกตัวที่สั้นกว่า กระทั่งครั้งที่สุด เพื่อประโยชน์ในการอ้างอิงและโอกาสป้อนข้อมูลผิดพลาดน้อยลง

คีย์คู่แข่ง หรือคีย์สำรอง (Alternate Key) คือกลุ่มของแอตทริบิวที่มีคุณสมบัติดังนี้

1. Uniqueness คือมีค่าไม่ซ้ำ 2 แถวใด ๆ
2. minimally คือกลุ่มของแอตทริบิวที่มีจำนวนน้อยที่สุด ที่จะทำให้ได้คุณสมบัติ Uniqueness

หรือถ้าจะกล่าวให้ง่ายก็คือ แอททริบิวต์ 1 ตัวหรือมากกว่า 1 ตัวก็ได้ ที่สามารถใช้เป็นตัวบอกได้ว่าเมื่อเราอ้างอิงถึงแอททริบิวต์นี้แล้วเราจะได้ผลลัพธ์ที่ไม่ซ้ำกัน และแอททริบิวต์อื่น ๆ ที่เหลือที่มีได้เป็นคีย์หลัก และไม่ได้เป็นส่วนหนึ่งส่วนใดของคีย์หลักก็จะถูกเรียกว่าเป็น Nonkey Attribute

ค่าว่าง (Null Value) คือสิ่งที่มีลักษณะดังนี้

1. เป็นแอททริบิวต์ที่เกี่ยวข้องแต่ไม่ทราบค่า
2. เป็นแอททริบิวต์ที่ไม่เกี่ยวข้องกับแอททริบิวต์ตัวอื่นในทพเพิลนั้นเลย

ส่วนคีย์ที่เป็นแอททริบิวต์ของรีเลชันอื่น ที่ชี้กับแอททริบิวต์ที่เป็นคีย์หลักของรีเลชันหนึ่งจำเรียกว่าคีย์นอก (Foreign Key)

คำว่า โดเมน (Domain) จะหมายถึงกรอบของค่าต่าง ๆ ที่เป็นไปได้ เช่น โดเมนของแอททริบิวต์วันที่จะหมายถึงค่าของวันที่ที่เป็นไปได้คือมีค่าเท่ากับ 1 ถึง 31

แต่ในการเก็บค่าข้อมูลลงในรีเลชันนั้น บางกรณีที่เรามีการกำหนดโดเมนให้แอททริบิวต์แล้ว แต่ข้อมูลที่จะถูกเข้าไปอาจถูกบรรจุเข้าไปในภายหลัง ลักษณะนี้จะทำให้เกิด ค่าว่างขึ้นขณะก่อนที่จะมีการบรรจุค่าข้อมูลที่อยู่ในโดเมนที่กำหนดไว้เข้าไป

เมื่อมีการจัดเก็บในมาตรฐานข้อมูลใด ๆ แล้วข้อมูลจะถูกแยกออกเป็นกลุ่มของข้อมูลเป็นชุดที่ประกอบด้วยแอททริบิวต์ต่าง ๆ ที่มีความสัมพันธ์กัน เช่นการเก็บข้อมูลของบุคคลากรในโรงเรียนก็อาจแยกเก็บเป็นกลุ่มของข้อมูลนักเรียน, กลุ่มข้อมูลของอาจารย์ และกลุ่มของนักการภารโรงเป็นต้น กลุ่มข้อมูลแต่ละกลุ่มนี้จะเรียกว่า เอนติตี้ (Entity) ซึ่งแต่ละเอนติตี้จะประกอบไปด้วยแอททริบิวต์ต่าง ๆ ที่มีความสัมพันธ์ เช่น เอนติตี้ของนักเรียนก็จะประกอบไปด้วย ชื่อ, ที่อยู่, ชั้นเรียน เป็นต้น

จากการแยกจัดเก็บข้อมูลออกเป็นเอนติตี้ แต่ละเอนติตี้ก็จะมีความสัมพันธ์กัน และสามารถเปรียบเทียบคำศัพท์ที่ใช้ในเชิงวิชาการกับคำศัพท์ที่ใช้ทั่ว ๆ ไป ได้ดังนี้

ศัพท์เชิงวิชาการ

ศัพท์เชิงธุรกิจ

relation

table ที่มี row ไม่ซ้ำกัน

tuple

row หรือ record

candinality

จำนวนของ row ในแต่ละ table

attribute

column หรือ field

degree

จำนวน column ในแต่ละ table

primary key

column ที่อาจจะเป็น 1 column หรือมากกว่า ที่เมื่อระบุค่าข้อมูลที่เป็น primary key 1 ค่าจะได้ข้อมูลใน table นั้นออกมาเพียง 1 row

domain

ค่าของข้อมูลที่เป็นไปได้ทั้งหมดในแต่ละ

## column

ความสัมพันธ์ระหว่างเอนทิตีแบ่งออกเป็น 3 ชนิดคือ

### 1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One)

ความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างเอนทิตี หมายถึง เมื่อเอนทิตีหนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ค่าของข้อมูลดังกล่าวก็จะมีความสัมพันธ์กับค่าข้อมูลคีย์หลักของอีกเอนทิตีหนึ่ง เพียงค่าเดียวเท่านั้น

### 2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many)

ความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างเอนทิตี หมายถึง เมื่อเอนทิตีหนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ค่าของข้อมูลดังกล่าวก็จะมีความสัมพันธ์กับค่าข้อมูลคีย์หลักของอีกเอนทิตีหนึ่ง ได้หลายค่า

### 3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to Many)

ความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างเอนทิตี หมายถึง เมื่อเอนทิตีหนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ที่ต่างกันอาจอ้างอิงถึงค่าข้อมูลหลักของอีกเอนทิตีหนึ่ง ได้ค่าเดียวกันหรือหลายค่าก็ได้

นอกจากนี้ เอนทิตียังมีเอนทิตีบางประเภทที่การอ้างอิงถึงตัวมัน ได้อย่างสมบูรณ์นั้นจะต้องอ้างอิงถึงเอนทิตีอื่นเสมอเช่น หากระบบฐานข้อมูลของบุคคลากรในโรงเรียนมีเอนทิตีของลูกนักรการการโรงค้วยแล้ว การอ้างอิงถึงเอนทิตีนี้ได้จะต้องอ้างอิงถึงเอนทิตีนักการการโรงค้วยแล้ว การอ้างอิงถึงเอนทิตีนี้ได้จะต้องอ้างอิงถึงเอนทิตีนักการการโรงค้วยเสมอ เอนทิตีประเภทนี้เราเรียกว่าเอนทิตีชนิดอ่อน (weak entity)

### ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relation Database Management System)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์เป็นซอฟต์แวร์ สำหรับจัดการข้อมูลและความสัมพันธ์ของข้อมูลที่เป็นแบบเชิงสัมพันธ์ รวมทั้งมีภาษาสำหรับกำหนดโครงสร้างของข้อมูลในระบบต่าง ๆ ความสถาปัตยกรรมมาตรฐานของระบบฐานข้อมูล มีภาษาสำหรับการค้นหา และแก้ไขข้อมูลรวมทั้งการจัดการความปลอดภัยของข้อมูล การเข้าถึงข้อมูลและการติดต่อกับผู้ใช้ และ โปรแกรมของผู้ใช้ที่เขียนด้วยโปรแกรมอื่น ดังมีรายละเอียดของประโยชน์ของการใช้ระบบการจัดการข้อมูลเชิงสัมพันธ์ ดังนี้

#### - ภาษาเอสคิวแอล (SQL : Structure Query Language)

ช่วยจัดการการทำงานเกี่ยวกับข้อมูล สามารถแบ่งตามลักษณะการทำงานได้ 2 ส่วนคือ

1. การจัดการเกี่ยวกับการกำหนด โครงสร้างข้อมูล (Data Definition Language : DDL) มีหน้าที่ในการกำหนดโครงสร้างข้อมูลที่ใช้แต่ละคนมองเห็น โครงสร้างข้อมูลนั้นก่อกอบแบบฐานข้อมูลมองเห็นและ โครงสร้างข้อมูลที่จัดเก็บในอุปกรณ์เก็บข้อมูลซึ่งผลของการแปล DDL จะเก็บในไฟล์พิเศษที่เรียกว่าพจนานุกรมของข้อมูล (Data Dictionary)

2. การจัดการเกี่ยวกับข้อมูล (Data Manipulation Language : DML) มีหน้าที่ในด้านจัดการ การเข้าถึงข้อมูล ได้แก่ การสอบถามหรือการเรียกค้นข้อมูลที่อยู่ในฐานข้อมูลการเพิ่มเติมข้อมูลใหม่ เข้าไปในฐานข้อมูล การลบข้อมูลออกจากฐานข้อมูล การเปลี่ยนแปลงแก้ไขข้อมูลที่อยู่ในฐานข้อมูล

- ประสิทธิภาพสูง และสะดวกรวดเร็ว ในการพัฒนาระบบงาน (Productivity)

เนื่องจากเมื่อต้องการข้อมูลจะอ่านข้อมูลจะอ่านขึ้นมาเป็นเซตของข้อมูล ทำให้สามารถ ทำงานได้สะดวกรวดเร็วและสามารถพัฒนาระบบงานได้รวดเร็วขึ้น นอกจากนี้โครงสร้างการทำงาน เป็นลักษณะที่รองรับหลายคน ดังนั้นจึงไม่ต้องเขียนโปรแกรมที่ควบคุมการให้ผู้ใช้สามารถทำงานได้ พร้อม ๆ กันหลายคน ควบคุมด้านความปลอดภัยของข้อมูล ควบคุมด้านอำนาจสิทธิของผู้ใช้แต่ละ คน นอกจากนี้ยังสามารถกู้ข้อมูลได้โดยอัตโนมัติเมื่อระบบงานเกิดความเสียหาย

- การจัดการด้านความเป็นอิสระของข้อมูล (Data Independence)

เมื่อมีการเปลี่ยนแปลงแก้ไขโครงสร้างข้อมูล ในนิยามข้อมูลระดับแนวความคิดหรือใน นิยามข้อมูล ระดับภายในนั้นจะไม่มีผลกระทบต่อโปรแกรมที่ให้ผู้ใช้งานในนิยามข้อมูลระดับภายใน นอก เนื่องจากระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ จะช่วยจัดการให้ผู้ใช้งานมองเห็น โครงสร้างข้อมูล ในนิยามข้อมูลระดับภายนอกเหมือนเดิมและสามารถใช้งานได้ตามปกติ ซึ่งการจัดการด้านความเป็น อิสระของข้อมูลแบ่งออกเป็น 2 ลักษณะคือ

1. ความเป็นอิสระของข้อมูลในระดับตรรกะ (Logic Data Independence) เมื่อมีการแก้ไข โครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในนิยามข้อมูล ระดับภายนอกที่ผู้ใช้ใช้งานกันอยู่เช่น การเปลี่ยนแปลงขนาดคอลัมน์ หรือเพิ่มคอลัมน์เข้าไป

2. ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical Data Independence) เมื่อมีการแก้ไข โครงสร้างข้อมูลในนิยามข้อมูลระดับภายใน จะไม่มีผลกระทบต่อโครงสร้างข้อมูลระดับแนวคิดเช่น มีการเปลี่ยนแปลงวิธีการเข้าถึงข้อมูลในนิยามข้อมูลระดับภายใน โดยการเปลี่ยนแปลงลักษณะการ จัดเก็บจาก ซีไอแอม (C-ISAM) เป็นบีทรี (Btree) หรือแม้กระทั่งเปลี่ยนแปลงอุปกรณ์ที่ใช้จัดเก็บข้อมูลอย่างเช่น ฮาร์ดดิสก์ (Harddisk) หรือการอัปเกรด (Upgrade) ระบบปฏิบัติการ

- ความปลอดภัยของข้อมูล (Data Security)

ระบบการจัดการฐานข้อมูลได้มีระบบการสอบถามรายชื่อพร้อมทั้งรหัสลับของผู้เข้ามาใช้ งานในระบบงานเพื่อให้เข้าทำงานในส่วนที่ผู้ใช้เกี่ยวข้องกับเท่านั้น นอกจากนี้ข้อมูลที่เก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์นั้นจะถูกทำการเข้ารหัส (Encryption) เพื่อป้องกันการเข้าถึงข้อมูลด้วยภาษาอื่น ๆ โดย ไม่ผ่านระบบการจัดการฐานข้อมูลเชิงสัมพันธ์

### - การจัดอำนาจในการใช้ข้อมูล (Data Authority)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ สามารถทำการกำหนดสิทธิอำนาจการใช้งานข้อมูลให้แก่ผู้ใช้ต่าง ๆ ได้โดยการใช้คำสั่ง GRANT หรือเพิกถอนสิทธิการใช้โดยใช้คำสั่ง REVOKE นอกจากนี้ยังใช้ VIEW เพื่อช่วยกำหนดสิทธิการใช้ข้อมูลของผู้ใช้แต่ละคนในนิยามข้อมูลระดับภายนอกได้อีกด้วย

### - ระบบการกู้ข้อมูลโดยอัตโนมัติเมื่อระบบเกิดความเสียหาย (Recovery Control)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ ได้เตรียมการรองรับการทำการประมวลผลกลุ่มงาน (Transaction Processing) โดยกลุ่มงาน (Transaction) นี้มีความหมายถึงกลุ่มของงานในทางตรรกะ (Logic Unit of Work) ซึ่งยอมให้มีการละเมิดกฎสำหรับควบคุมความถูกต้องของข้อมูลได้ภายในกลุ่มของงานและการทำงานทั้งหมดในกลุ่มงานนี้ ถ้าทำงานเสร็จตามความต้องการจะต้องเสร็จด้วยกันทั้งหมด ถ้าเกิดความบกพร่องขึ้นภายในกลุ่มงานจะต้องยกเลิกทั้งหมดโดยที่ Commit คือคำสั่งยืนยันการทำงานสำเร็จ ส่วน Rollback คือคำสั่งยกเลิกการทำงานของกลุ่มงานนั้น ดังนั้นถ้า Commit แล้วระบบเกิดความเสียหายขึ้น ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะรับรองว่าข้อมูลที่ Commit แล้วจะไม่สูญหายไปทางกลับกันกรณีของ Rollback และการทำงานที่ยังไม่ Commit นั้นระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะรับรองข้อมูลเดิมก่อนการทำงานยังไม่สูญหายไปไหน

### - การดูแลผู้ใช้หลายคนให้สามารถทำงานได้พร้อม ๆ กัน (Concurrency Control)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ จะมีการรองรับการทำงานพร้อมกันของผู้ใช้หลายคนได้โดยอาจมีปัญหากเกิดขึ้นได้ 3 ประเภทใหญ่ ๆ คือ

#### 1. ปัญหการสูญเสียการเปลี่ยนแปลงข้อมูล (The Lost Update Problem)

เกิดจากการอ่านข้อมูลตัวเดียวกัน ไปใช้งาน โดยที่มีการเปลี่ยนแปลงข้อมูลตัวนั้นในเวลาที่ไม่พร้อมกัน

2. ปัญหาที่เกิดจากการยกเลิกการเปลี่ยนแปลงข้อมูล (The Uncommitted Dependency Problem) เกิดจากมีการนำเอาข้อมูลไปใช้ในขณะที่ยังไม่เสร็จและมีการยกเลิกการประมวลผลกลุ่มงานนั้น

#### 3. ปัญหาที่เกิดจากการนำเอาข้อมูลไปประมวลผล และมีการยอมรับความเปลี่ยนแปลงนั้น

โดยที่การเปลี่ยนแปลงที่เกิดขึ้นมีผลต่ออีกการทำงานหนึ่ง ซึ่งปัญหาต่าง ๆ เหล่านี้สามารถป้องกันได้โดยวิธีการจองข้อมูล (Lock) ก่อนการใช้งาน

### ทฤษฎีการออกแบบงาน

แนวทางการออกแบบระบบงานวิธีหนึ่งที่นิยมกันอย่างแพร่หลาย และสามารถแสดงความสัมพันธ์ในรายละเอียดได้ชัดเจนคือ วิธีการออกแบบดาต้าโฟลว์ไดอะแกรม (DFD : Data Flow Diagram) ซึ่งวิธีนี้จะใช้สัญลักษณ์

ที่แสดงความสัมพันธ์เพียง 4 สัญลักษณ์เท่านั้น และจะเริ่มต้นพิจารณาความสัมพันธ์ของระบบอย่างกว้าง ๆ เป็นอันดับแรกก่อน หลังจากนั้นจะพิจารณาในรายละเอียดในแต่ละส่วนงานเพิ่มขึ้นเป็นอันดับต่อไป

### ทฤษฎีการออกแบบฐานข้อมูลเชิงสัมพันธ์

แนวทางการออกแบบฐานข้อมูลสามารถแบ่งแนวทางเป็น 2 แนวทาง คือ

#### 1. แบบวิเคราะห์ (Analysis หรือ Decomposition)

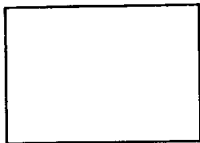
มีแนวทางการออกแบบคือ พิจารณาแอทธิบิวและความสัมพันธ์ในระบบงานทั้งหมด จากนั้นจะทำการแบ่งแยกออกเป็นกลุ่มของแอทธิบิวที่มีความสัมพันธ์ในกลุ่มที่เล็กลงมา ตัวอย่างของวิธีการออกแบบฐานข้อมูลแนวทางนี้คือ วิธีนอร์มอลไลเซชัน (Normalization)

#### 2. แบบสังเคราะห์ (Synthesis หรือ Reconstruction) มีแนวทางการออกแบบคือ พิจารณาแต่

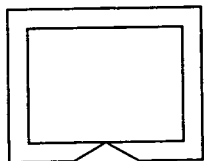
ละแอทธิบิวว่ามีความสัมพันธ์กับแอทธิบิวตัวอื่นอย่างไร แล้วเขียนแผนภาพแสดงความสัมพันธ์เมื่อสามารถกำหนดความสัมพันธ์ให้กับแต่ละแอทธิบิวในระบบงานได้ทั้งหมดแล้วก็จะทำการสร้าง (rebuild) กลุ่มของความสัมพันธ์โดยใช้คีย์หลักร่วมกัน (common Primary Key) ตัวเดียวกัน ตัวอย่างของวิธีการออกแบบฐานข้อมูลแนวทางนี้คือวิธีอีอาร์ไออะแกรม (ER : Entity Relationship Model) เป็นเครื่องมือในการออกแบบฐานข้อมูลเชิงสัมพันธ์ในระดับตรรกะเนื่องจากแนวความคิดที่ให้ Conceptual Schema มีพื้นฐานมาจากโครงสร้างธรรมชาติ ใช้รูปประโยคที่มีประธาน กริยา กรรม วิธีแสดงรูปแบบความสัมพันธ์ระหว่างข้อมูลเป็นแบบจำลองที่มีความหมายและมีเครื่องหมายแสดงความสัมพันธ์ของข้อมูลและข้อจำกัดของข้อมูลได้อย่างชัดเจนนอกจากนั้นยังสามารถแปลง Conceptual Schema เป็น Relational Database Schema ซึ่งจะอยู่ในรูป Third Normal Form และทำการนอร์มอลไลเซชัน (Normalization) ต่อจนถึง Fifth Normal Form ซึ่งวิธีการแบบอีอาร์ไออะแกรมนี้จะใช้สัญลักษณ์แสดงความสัมพันธ์ของข้อมูลในระบบงานแสดงข้อกำหนดต่าง ๆ เพื่อช่วยแสดงความสัมพันธ์ให้ชัดเจนยิ่งขึ้น ซึ่งมีสัญลักษณ์ต่าง ๆ ได้ดังนี้

สัญลักษณ์

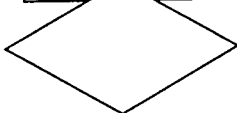
ความหมาย



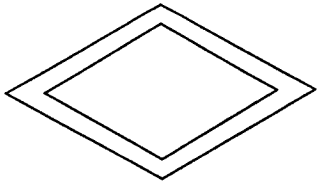
ชนิดของเอนติตี้ (Entity Type)



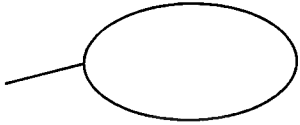
เอนติตี้ชนิดอ่อน (Weak Entity Type)



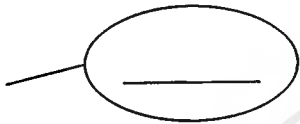
ความสัมพันธ์ (Relationship Type)



ความสัมพันธ์ระหว่างเอนทิตีกับเอนทิตีชนิด  
อื่น (Identifying Relationship Type)



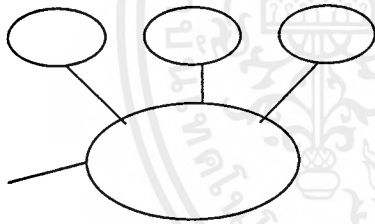
แอททริบิว (Attribute)



คีย์แอททริบิว (Key Attribute)



แอททริบิวที่มีหลายค่า (Multivalued Attribute)

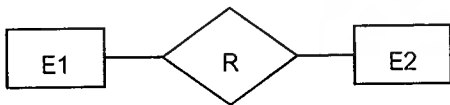


แอททริบิวที่ประกอบด้วยหลายแอททริบิวย่อย  
(Composite Attribute)

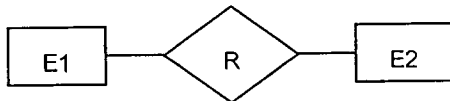
แอททริบิวที่เกิดจากความสัมพันธ์ระหว่างแอททริบิวด้วยกันเอง (Derived Attribute)

สัญลักษณ์

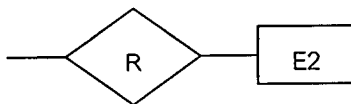
ความหมาย



การขึ้นต่อกันอย่างสมบูรณ์ของ E2 ใน R



อัตราส่วนความสัมพันธ์ 1 : N สำหรับ  
E1 : E2 ใน R



ค่าคงที่ของโครงสร้าง (min, max)  
บนการขึ้นต่อกันของ E ใน R

รูปที่ 2.11

## บทที่ 3 การสร้าง

### 1. ระบบสืบค้นรายชื่อนักศึกษา

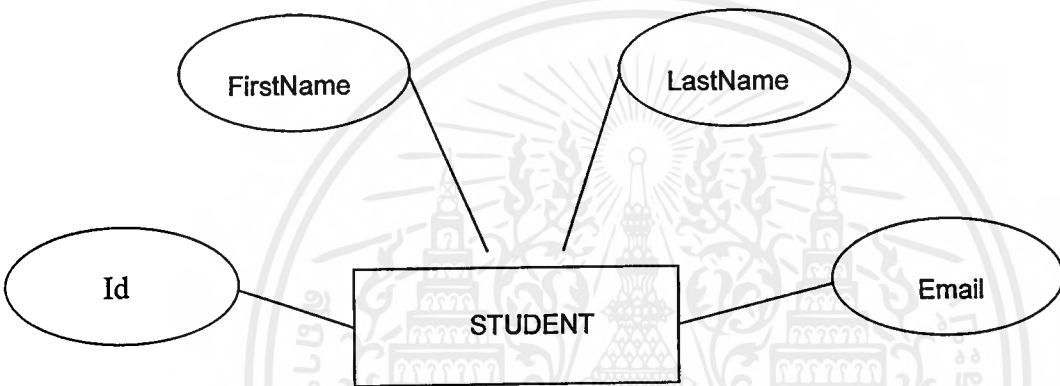
#### 1.1 วัตถุประสงค์ของระบบสืบค้นรายชื่อนักศึกษา

1.1.1 เพื่อเป็นการสร้างโปรแกรมต้นแบบระบบสืบค้นฐานข้อมูล

1.1.2 เพื่อสืบค้นรายชื่อนักศึกษา

#### 1.2 ส่วนประกอบของระบบสืบค้นรายชื่อนักศึกษา

1.2.1 ส่วนของฐานข้อมูล



รูปที่ 2.12

ฐานข้อมูลประกอบไปด้วยตารางบรรจุกความสัมพันธ์ของรหัสนักศึกษา รายชื่อนักศึกษาและข้อมูลต่าง ๆ ของนักศึกษา จาก ER Diagram สามารถแมปเป็นตารางที่ใช้ในการจัดเก็บข้อมูลได้

STUDENT

Id	FirstName	LastName	Email
----	-----------	----------	-------

1.2.2 ส่วนตั้งการติดต่อจากผู้ใช้ Index.asp ซึ่งประกอบไปด้วยฟอร์มที่บรรจุ

METHOD = "Post" ซึ่งเป็นการส่งข้อมูลมาที่เครื่องให้บริการ

ACTION = "search.asp?status=INIT" เป็นการกำหนดการติดต่อหลังจากได้กดปุ่ม search ซึ่งประกอบไปด้วยไฟล์ที่จะติดต่อ และสถานะซึ่งเป็นการกำหนดสถานะเริ่มต้นของ search.asp ในรูปแบบของ Query String

```
<HTML>
<HEAD>
<FORM NAME="MyForm" METHOD="POST" ACTION="search.asp?status=INIT">
<P>
Word: <INPUT TYPE="Text" NAME="search_text" SIZE="20" MAXLENGTH="100" VALUE=""
>
<P>
INPUT TYPE=SUBMIT><INPUT TYPE=RESET>
</FORM>
</BODY>
</HTML>
```

### 1.2.3 ส่วนติดต่อกับฐานข้อมูลและแสดงผล

เป็นไฟล์ข้อมูลในรูปแบบของ ASP ซึ่งได้รับข้อมูลมาจาก index.asp ประกอบไปด้วย 3 ส่วน

คือ

#### 1. ส่วนตรวจสอบสถานะและสร้างคำสั่งภาษา SQL

```
<%
'CHECK STATUS
If Request.QueryString("status") = "" Then
Response.Redirect "index.asp"
Else
'CHECK status = INIT
If Request.QueryString("status") = "INIT" Then
If Request.Form("search_text") <> "" Then
'READ DATA FROM FORM
search_text = Request.Form("search_text")
'OPEN NEW ARRAY
```

```

Session("MyArray") = ""

'SPLIT DATA
MyArray = SPLIT(search_text, " ")
For z=0 To UBOUND(MyArray)
MyArray(z) = TRIM(MyArray(z))
If MyArray(z) <> " " Then
If Session("MyArray") = "" Then
Session("MyArray") = " (" & Session("MyArray") & " Id LIKE '%" & REPLACE(MyArray(z), "'",
''''') & "%' "
ParenClose = "True"
Else
If AndCase = "True" Then
Session("MyArray") = Session("MyArray") & " AND (Id LIKE '%" & REPLACE(MyArray(z), "'",
''''') & "%' "
AndCase = "False"
ParenClose = "True"
Else
Session("MyArray") = Session("MyArray") & " OR Id LIKE '%" & REPLACE(MyArray(z), "'", ''''')
& "%' "
End If
End If
End If
Next

If Session("MyArray") = "" Then Response.Redirect "index.asp"
If ParenClose = "True" Then Session("MyArray") = Session("MyArray") & ") "
Session("CurrentPage") = 1
Else
Response.Redirect "index.asp"
End If

```

Else

'CHECK status = TRAV

If Request.QueryString("status") = "TRAV" Then

If Request.QueryString("MOVE") = "" Then

If Session("MyArray") <> "" Then

Session("CurrentPage") = 1

Else

Response.Redirect "index.asp"

End If

Else

'CHECK status = PREV

If Request.QueryString("MOVE") = "PREV" Then

If Session("MyArray") <> "" Then

If Session("CurrentPage") = "" Then

Session("CurrentPage") = 1

Else

If Session("CurrentPage") > 1 Then

Session("CurrentPage") = Session("CurrentPage") - 1

End If

End If

Else

Response.Redirect "index.asp"

End If

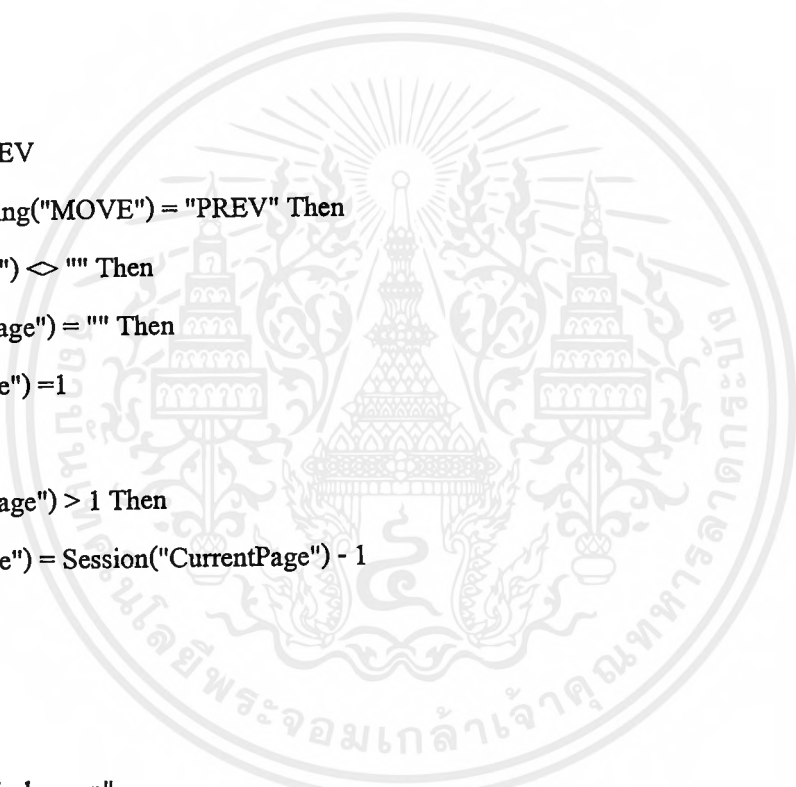
Else

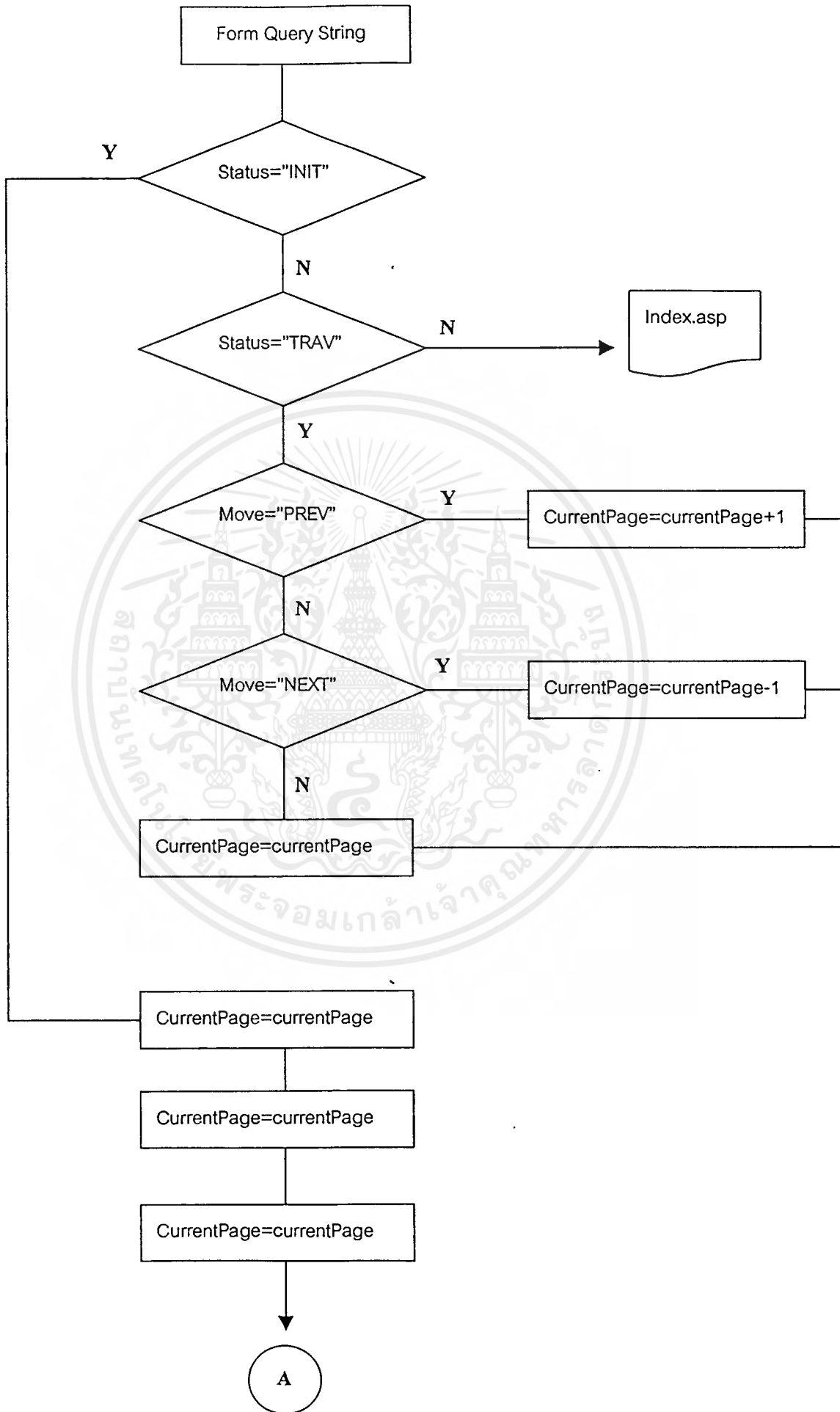
'CHECK status = NEXT

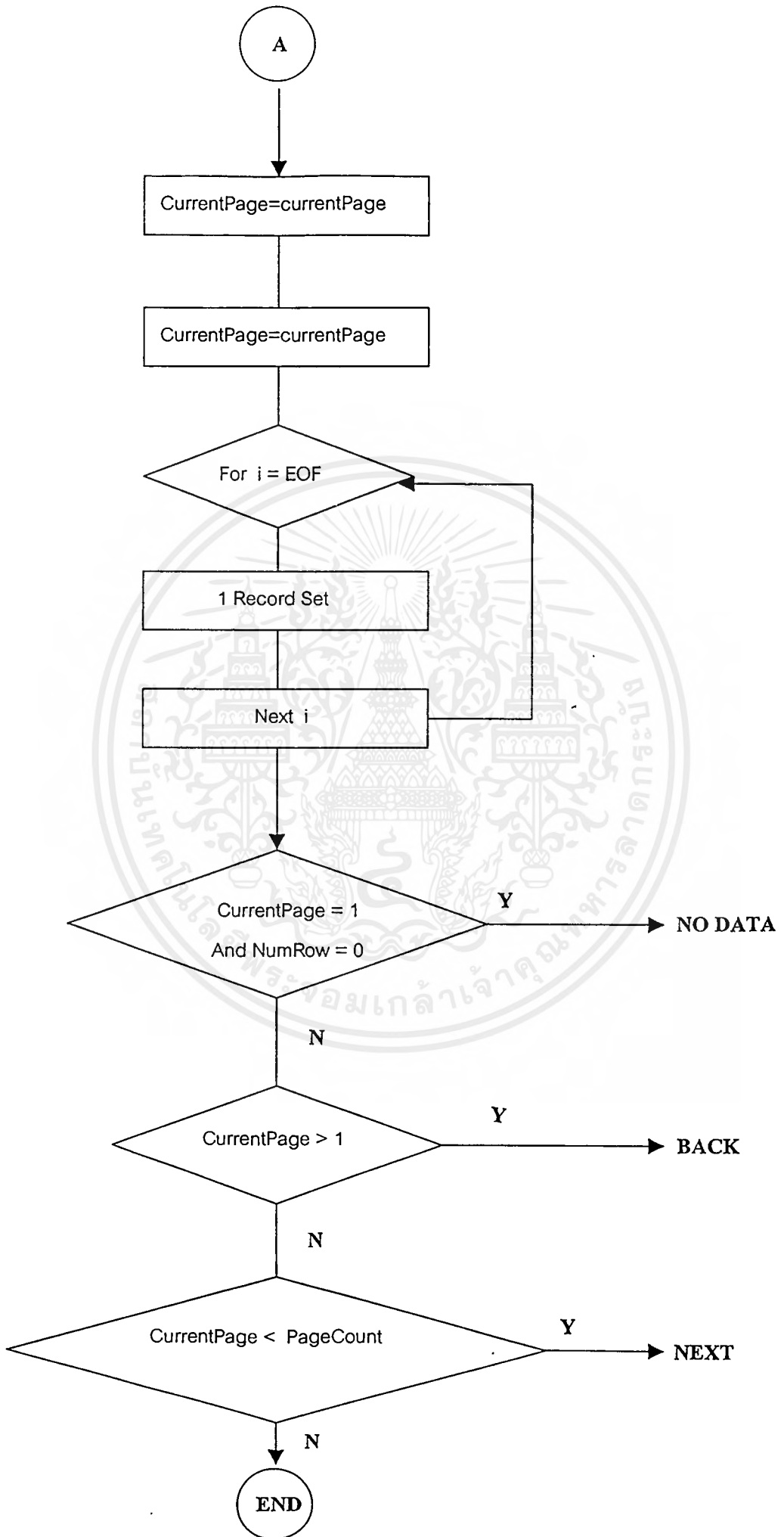
If Request.QueryString("MOVE") = "NEXT" Then

If Session("MyArray") <> "" Then

If Session("CurrentPage") = "" Then







## 2. ส่วนติดต่อกับฐานข้อมูลและแสดงผล

เป็นส่วนที่รับคำสั่งภาษา SQL มาจากส่วนแรก และสร้างออบเจกต์ (object) การติดต่อฐานข้อมูลผ่านทาง ODBC และนำผลที่ได้จากฐานข้อมูลมาสร้างไฟล์ asp ใหม่เพื่อส่งกลับไปยังเครื่องปลายทาง

```

<% .
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "KMITL","",""
Set Session("KMITL_conn") = conn
sql = "SELECT * FROM [Student] WHERE "& Session("MyArray")
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open sql, conn, 3, 3
On Error Resume Next
%>
<%
'set number of reccords in a page
RS.PageSize = 10
'set the current page
If NOT RS.BOF AND NOT RS.EOF Then
RS.AbsolutePage = Session("CurrentPage")
End If
NumRows=0
Toggle = "True"
%>
<CENTER>
<TABLE BGCOLOR="#8F7F6F" BORDER=1 WIDTH="90%" >
<TR BGCOLOR="#30406F">
</TR>
<TR BGCOLOR="#D0D0C0">
<TD VALIGN=TOP>

```

```

<TABLE>
<%
Do While not RS.EOF AND NumRows < RS.PageSize
If Toggle = "True" Then%>
<tr bgcolor="#000000">
<%
Toggle = "False"
Else
%>
<tr bgcolor = "#fcfcfc">
<%
Toggle = "True"
End If
%>
<TR>
<TD>
<TD NOWRAP><%=rs("id")%></TD>
<TD><%=rs("FirstName")%>
</TD>
<TD><%=rs("LastName")%>
</TD>
<TD><%=rs("Account")%>@chaokhun.kmitl.ac.th
</TD></TR>
<%
RS.MoveNext
Numrows = Numrows + 1
Loop
%>
</table>
<TR BGCOLOR="#30406F">
<TD>

Faculty of Engineering kmitl

```

```
</TD></TR>
```

```
</TABLE>
```

```
<%=sql%>
```

### 3. ส่วนสร้างการติดต่อกลับ

เป็นการนำข้อมูลจากฐานข้อมูลมาประมวลผล เพื่อสร้างการติดต่อกับผู้ใช้ปลายทางในครั้งต่อไป

```
ไป
```

```
<%
```

```
Content = ""
```

```
If Session("CurrentPage") = 1 AND Numrows = 0 Then
```

```
Content = Content & " <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=" &
```

```
DQ & "100%" & DQ & " BGCOLOR=" & DQ & "#FFFFFFC6" & DQ & ">" & LN
```

```
Content = Content & " <TR>" & LN
```

```
Content = Content & " <TD ALIGN=CENTER COLSPAN=2>No
```

```
DATA" & LN
```

```
Content = Content & " </TD>" & LN
```

```
+Content = Content & " </TR>" & LN
```

```
Content = Content & " </TABLE>" & LN
```

```
Response.Write(Content)
```

```
Else
```

```
%>
```

```
<hr>
```

```
<% If Session("CurrentPage") > 1 Then %>
```

```
[PREV]
```

```
<% End If %>
```

```
<% If Session("CurrentPage") < RS.PageCount Then %>
```

```
[NEXT]
```

```
<% Session("PageCount") = RS.PageCount %>
```

```
<% End If %>
```

```
<%
```

RS.Close  
Conn.Close  
End If  
%>



## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 เกี่ยวกับโปรแกรมการสร้างระบบผู้ให้บริการ

รายละเอียดที่ต้องศึกษา คือ HTML โดยมีการใช้ FORM Tag ทำหน้าที่ในการให้ผู้ใช้งานป้อนข้อมูล และการติดต่อกับฐานข้อมูล ใช้ ODBC เป็นเครื่องมือในการติดต่อ ซึ่งจะต้องมีการเขียน CGI Script ขึ้น เพื่อนำเอาข้อมูลจาก Browser ที่ผู้ใช้ระบบป้อนไว้แล้ว นำไปเก็บ โดยจะใช้หลักการ ของ ASP ในการเขียน CGI ขึ้นมา เมื่อมี ผู้ใช้ระบบ ที่ต้องการติดต่อกับข้อมูล ก็สามารถที่จะทำได้ โดยเพียงป้อนข้อมูล ที่ต้องการทราบ CGI ก็จะทำงาน ร่วมกับ Web Server ในการนำข้อมูลมาแสดง โดยผ่าน Browser ให้ผู้ใช้ระบบทราบได้

#### 4.2 การทดลอง

- 4.2.1 นำโปรแกรม CGI Application ติดตั้งที่เครื่อง Server
- 4.2.2 ทำการ Start โปรแกรม Internet Information Server
- 4.2.3 ให้เครื่อง Client ติดต่อเข้าหาเครื่อง Server
- 4.2.4 เมื่อสามารถติดต่อกันได้แล้วให้เครื่อง Client ทำการ RUN โปรแกรม Brows
- 4.2.5 ดูเอกสารใน Server (สามารถใช้ได้ทั้ง Netscape หรือ Microsoft Internet Explorer)
- 4.2.6 ป้อน Location เพื่อดูเอกสารที่ <http://161.246.29.99> [ IP Address ของเครื่อง Server ]
- 4.2.7 จะเห็นเป็น Home Page ของการทดลอง
- 4.2.8 ทำการป้อนข้อมูล ที่ต้องการติดต่อตามเอกสาร ฟอรัม ที่มีให้
- 4.2.9 จากนั้น ทดลองดูข้อมูลที่แสดง
- 4.2.10 จะเห็นข้อมูลใหม่ที่แสดงให้เห็นที่ Browser แสดงว่าการทำงาน ปกติ

## บทที่ 5

### บทวิจารณ์ และสรุป

การศึกษาการทำงานของระบบผู้ให้บริการเว็บ นั้น จำเป็นที่จะต้องศึกษา ถึง รายละเอียดต่างๆ ที่ทำงานเกี่ยวข้อง เว็บเซิร์ฟเวอร์ ซึ่งจะประกอบไปด้วย การทำงานของระบบ ปฏิบัติการที่ใช้งานนั้น โปรแกรมที่นำมาใช้งานเป็นเว็บเซิร์ฟเวอร์ ซึ่งส่วนใหญ่จะมีบริการไว้ใน อินเทอร์เน็ต สามารถที่จะนำมาทดลองใช้งานได้ โดยที่ต้องศึกษาการทำงานของโปรแกรมนั้นๆ ให้เข้าใจ การที่จะสามารถให้เว็บที่สร้างขึ้นมีความน่าสนใจ จะขึ้นอยู่กับการใช้ภาษา เอชที เอ็มแอล นำมาในการเขียน ซึ่งถ้ามีความเข้าใจ และหมั่นทดลองเขียน ก็จะสามารเขียนโฮมเพจที่ เป็นที่น่าสนใจขึ้นมาได้ ปกติ ล่าพังโฮมเพจเองจะเป็นในลักษณะไดนามิก คือไม่มีการเคลื่อนไหว ของเอกสารที่ดูนั้น แต่เมื่อนำเอาภาษาเชิงวัตถุ คือ จาวา มาช่วยในการทำให้โฮมเพจนั้น เกิดความ น่าสนใจยิ่งขึ้น เพราะจะเกิดเป็นภาพที่เคลื่อนไหวได้ จากภาพนิ่งธรรมดา ดังนั้น การที่จะศึกษา การทำงานของแต่ขึ้นมา การที่จะทำให้ผู้ใช้บริการ ป้อนข้อมูลที่ต้องการกลับมายังผู้ให้ บริการได้นั้น จำเป็นที่จะต้องมีส่วนกลาง ในการทำงานระหว่างผู้ให้บริการ กับผู้ใช้บริการ และแหล่ง ของข้อมูล เรียกส่วนนี้ว่า ซีจีไอ

การศึกษาข้อมูลและรายละเอียดต่างๆ ต้องศึกษาจากเอกสาร และตำราอ้างอิงของต่างประเทศเป็นส่วนใหญ่ เนื่องจากวิทยาการด้านนี้ยังไม่เป็นที่แพร่หลายในประเทศไทยนัก คณะผู้จัดทำก็ได้พยายามศึกษา ทั้งจากตำราเอง และ ละครภาษา จึงจำเป็นที่จะต้องใช้เวลา และหมั่นใช้งานภาษานั้นๆ จึงจะเกิด ความชำนาญในการเขียน สอบถาม ขอบคําแนะนำจากผู้ที่มีประสบการณ์มาบ้าง จึงทำให้โครงการนี้สำเร็จด้วยดี

เมื่อได้ทำการศึกษารายละเอียดของส่วนประกอบต่างๆ ของเว็บเซิร์ฟเวอร์แล้ว จำเป็นที่จะต้อง มี โปรแกรมที่สามารถทำงานได้จากการศึกษาข้อมูลต่างๆ ขึ้นมา โครงการนี้ได้สร้างโปรแกรมผู้ให้บริการเว็บ โปรแกรมสามารถที่จะเก็บและแสดงข้อมูลที่ผู้ใช้บริการป้อนไว้ให้ได้ โดยใช้ ความรู้ที่ได้ศึกษารายละเอียดของเว็บเซิร์ฟเวอร์มา สามารถทำให้โปรแกรมทำงานได้ตามที่ต้องการ ได้ ซึ่งเป็นพื้นฐานแก่ผู้ที่สนใจนำไปพัฒนาเขียนโปรแกรมเพื่อใช้งานใน เว็บเซิร์ฟเวอร์ โดยที่ในปัจจุบันการใช้งานด้านนี้ เริ่มเป็นที่สนใจมากยิ่งขึ้น จึงสามารถที่นำไปใช้ประโยชน์ต่างๆ ได้ และใช้เป็นการทำงานทางธุรกิจได้

## ผลดีของการที่ศึกษาโครงการนี้

1. สามารถสร้างระบบอินทราเน็ต ขึ้นมาได้จากการใช้วิธีการที่ทดลอง
2. สามารถที่นำเอาวิธีการเขียน โหมดเพจด้วยภาษา เอชทีเอ็มแอล และ เอเอสพี มาใช้ งานในเชิงการศึกษา เช่น การทำการให้บริการ การสืบค้นฐานข้อมูล ได้
3. สามารถที่จะนำความรู้ที่ได้ศึกษาในโครงการนี้ ไปประยุกต์ทำงานในเชิงธุรกิจได้ เช่น เป็นผู้ติดตั้ง ระบบเครือข่าย ทั้งภายนอกและภายใน ได้



## ภาคผนวก

### เว็บไซต์ (Web Stie) รายละเอียดของข้อมูล

จากการที่ศึกษารายละเอียดของข้อมูลต่างๆ ซึ่งในประเทศไทยยังมีเอกสาร และตำรา น้อย ทำให้การที่จะศึกษาข้อมูลต่างๆ ต้องศึกษาจาก อินเทอร์เน็ต ทางคณะผู้ดำเนินการ โครงการนี้ ก็ได้ นำเอาข้อมูลเรื่องต่างๆ ที่ศึกษาใน โครงการ มาจากอินเทอร์เน็ต ที่เว็บไซต์ (Web Stie) ดังต่อไปนี้

#### 1. เรื่อง ซีจีไอ

<http://www.activeserverpages.com>

<http://www.wpi.edu/~bmorin/fnord/doc/cgi.html>

<http://www.jmarshall.com/easy/cgi>

<http://hoohoo.ncsa.uiuc.edu/cgi>

<http://www.fas.harvard.edu/~glau/httpd/>

<http://website.ora.com/wdocs/32demo/windows-cgi.html>

#### 2. เรื่อง เจฟเวอร์

<http://www.yahoo.com/>

<http://www.microsoft.com/ntserver/tools/web.htm>

<http://www.fas.harvard.edu/~glau/httpd/docs/index.html>

<http://www.wpi.edu/~bmorin/fnord>

[http://emwac.edu.ac.uk/html/internet\\_toolchest/https/](http://emwac.edu.ac.uk/html/internet_toolchest/https/)

## เอกสารอ้างอิง

1. จักรกฤษณ์ นันทพินิต จันทิวรา นาคะภากร วิรุฬ ทรัพย์ถาวรกุล, คู่มือการใช้งานไมโครซอฟท์ แอ็กเซส 2 สำหรับวินโดวส์, บ.ซีเอ็ดยูเคชั่น จำกัด, 2539, 797 หน้า
2. บทความทั่วไป, "TCP/IP", วารสาร อินเทอร์เน็ต อินทราเน็ต แมกกาซีน, ฉบับที่ 1, 2539, หน้า 37-41
3. โชคณ์ อรุณยิ่งมงคล, "การเขียน Home Page โดยใช้ HTML", วารสารอินเทอร์เน็ต อินทราเน็ต แมกกาซีน, ฉบับที่ 1, 2539, หน้า 55-64
4. ณรงค์ชัย เลิศสิทธิ์, "เส้นทาง NT สู่อินเทอร์เน็ต", วารสารคอมพิวเตอร์, ปีที่ 8, ฉบับที่ 86, 2539, หน้า 165-169
5. ยุทธนา สอนสุข, "ประจักษ์ของ WEB ที่ชื่อว่า CGI", วารสาร อินเทอร์เน็ต อินทราเน็ต แมกกาซีน, ฉบับที่ 3, 2539, หน้า 59-74
6. ยุทธนา สอนสุข, "เก็บมาเล่าวิธีการติดตั้ง Web Server", วารสาร อินเทอร์เน็ต อินทราเน็ต แมกกาซีน, ฉบับที่ 3, 2539, หน้า 111-118
7. รุจการ ลสุภาพันท์, "เปรียบเทียบสองระบบปฏิบัติการ Windows 95 และ Windows NT", วารสารคอมพิวเตอร์รีวิว, ปีที่ 12, ฉบับที่ 140, 2539, หน้า 164-169
8. สุกฤษณ์ สุ่มมาตย์, "JAVA", วารสาร อินเทอร์เน็ต อินทราเน็ต แมกกาซีน, ฉบับที่ 1, 2539, หน้า 43-53
9. สยาม สงวนรัมย์, "เทคนิคการเพิ่มประสิทธิภาพให้กับวินโดวส์", บริษัท เอ อาร์ อินโฟเมชั่น แอน พับลิเคชั่น, 2537, 59 หน้า
10. เอนก กนกอภิวัฒน์, "รู้จัก JAVA", วารสาร อินเทอร์เน็ต อินทราเน็ต แมกกาซีน, ฉบับที่ 1, 2539, หน้า 35-39
11. อรุณา พัวรัตนอรุณกร, "มาหัดเขียน HTML กันเถอะ", วารสาร BCM คอมพิวเตอร์, ปีที่ 7, ฉบับที่ 84, 2539, หน้า 205-212
12. Turbowin, "แนะนำ JAVA ภาษาสำหรับนักพัฒนายุคใหม่", Internet Magazine, บริษัทซีเอ็ดยูเคชั่น จำกัด, 2539, หน้า 29-34
13. Ed Title Mark Gaither Sebastain Hassinger Mike Eewin, Foundation Of WWW Programmning With HTML & CGI, IDG Books Inc., CA, 1995, 648 หน้า
14. Glenn Vanderburg, Trick Of the JAVA programming Gurus, Sans.net Publishing, 1996, 815 หน้า
15. Microsoft Windows NT version 4.0 Installation Guide, Microsoft Corporation, 1994, 192หน้า