

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้เว็ลด์ไวด์เว็บกับการใช้งานห้องสมุด
Internet Application Document Retrival System



โดย

นายธรรมบุญ วงศ์ธีเรศ

นายสมบัติ แก้วธรรมชัย

นายธีรเดช หรั่งเพ็ชร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เลขหมู่.....

เลขทะเบียน.....30480

วัน, เดือน, ปี.....17 ก.ค. 2541

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้เวลาด์ไวด์เว็บกับการใช้งานห้องสมุด
Internet Application Document Retrival System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

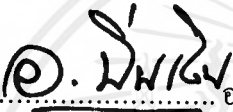
เรื่อง การประยุกต์ใช้เว็ลด์ไวด์เว็บกับการใช้งานห้องสมุด

ผู้จัดทำ

1. นายธรรมบุญ วงศ์ธีเรศ

2. นายสมบัติ แก้วธรรมชัย

3. นายธีรเดช หรั่งเพชร

..... อาจารย์ที่ปรึกษา

(ดร.เออน ปิ่นเงิน)

..... อาจารย์ที่ปรึกษา

(อ.ธนา หงษ์สุวรรณ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้เว็ลด์ไวด์เว็บกับการใช้งานห้องสมุด

ธรรมบุญ วงศ์ธีเรศ
สมบัติ แก้วธรรมชัย
ธีรเดช หรั่งเพ็ชร

ดร.เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
อ.ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการประยุกต์เว็ลด์ไวด์เว็บกับการใช้งานห้องสมุด โดยเป็นการนำอินเทอร์เน็ตเข้ามาทำการพัฒนางานของสำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ให้มีความสะดวกในให้บริการแก่ผู้ที่มาใช้บริการให้มากยิ่งขึ้น ซึ่งได้ทำการพัฒนาในส่วนโฮมเพจของสำนักหอสมุดกลาง ให้มีความสามารถในการค้นหาหนังสือของสำนักหอสมุดกลางผ่านทางเว็บเบราว์เซอร์ได้ ซึ่งใช้โปรแกรมภาษาจาวาการพัฒนาในส่วนของงานบริการของสำนักหอสมุดกลาง ได้แก่ การแนะนำบริการและหนังสือต่างๆ การเผยแพร่ข่าวสาร ซึ่งได้พัฒนาในส่วนให้ผู้ดูแลระบบสามารถควบคุมการทำงานให้ดียิ่งขึ้น จึงทำให้ผู้ดูแลไม่ต้องมีความรู้ด้านคอมพิวเตอร์มากนัก โดยในส่วนงานบริการนี้จะใช้การทำงานของ CGI โดยใช้โปรแกรมภาษา Perl ในการพัฒนา และในส่วนสุดท้ายพัฒนาส่วนการนำพระราชดำรัส และเพลงพระราชนิพนธ์ โดยใช้โปรแกรม Real Audio โดยคณะผู้จัดทำหวังว่าปริญญานิพนธ์นี้จะเป็นประโยชน์ในการศึกษาสำหรับผู้สนใจต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Internet Application Document Retrieval System

Thammanoon Wongtheeres

Sombat Kawturmachai

Teeradate Rungphat

Dr.Ouen Pinngern Advisor

Thana Hongsuwan Advisor

Abstract

This thesis presents the application of world wide web for library usage which use Internet to improve efficiency of library management in the central library of Kmitl . This application provides the used more convenience when using library by developing central library's homepage to make it's capable of searching the books in the library by passing the web browser.

In the part of service routines in the library such as service and books recommendation and broadcasting information, this application uses JAVA language in the development and makes it easy for the administer to control so the administer need not to have so much knowledge about computer. With these service tasks ,the application.

Uses CGI which have developed in Pearl language and used Real audio program in the part of presenting the speech of the throne and the songs writing by the King. Finally ,we hope this thesis will be useful for person who interested in this application next.

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	1
1.4 แนวทางในการพัฒนา	2
บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับระบบเครือข่าย	3
2.1 ระบบเครือข่าย และการพัฒนาโปรแกรมระบบเครือข่าย	3
2.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์ และอินเทอร์เน็ตเวิร์กกิง	3
2.1.2 รูปแบบของโปรแกรมบนระบบเครือข่าย	3
2.1.2.1 การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์	3
2.1.2.2 การประมวลผลแบบกระจาย	4
2.1.3 โมเดล OSI	4
2.1.4 ลักษณะของการติดต่อ	5
2.1.4.1 Connection – Oriented	5
2.1.4.2 Connectionless หรือดาต้าแกรม	5
2.1.5 แอดแควส	5
2.2 ความรู้เบื้องต้นเกี่ยวกับโปรโตคอล TCP/IP	6
2.2.1 เปรียบเทียบระหว่างโปรโตคอล TCP/IP และ OSI โมเดล	6
2.2.1.1 แอปพลิเคชันเลเยอร์	6
2.2.1.2 ทรานสปอร์ตเลเยอร์	7
2.2.1.3 อินเทอร์เน็ตเลเยอร์	7
2.2.1.4 ฟิสิคอลลเลเยอร์	7
2.2.2 รูปแบบการกำหนดแอดแควสของโปรโตคอล TCP/IP	7
2.2.2.1 การแบ่งเน็ตเวิร์กคลาส	7
2.2.2.2 การแทนด้วยเลขฐานสิบและจุด	8
2.2.2.3 การทำซับเน็ตติ้ง	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3	ชุดโปรโตคอล TCP/IP	10
2.2.3.1	โปรโตคอล IP	10
2.2.3.2	โปรโตคอล ARP	11
2.2.3.3	โปรโตคอล ICMP	11
2.2.3.4	โปรโตคอล RARP	11
2.2.3.5	โปรโตคอล UDP	11
2.2.3.6	โปรโตคอล TCP	11
2.2.4	หมายเลขพอร์ต	11
บทที่ 3	เทลเน็ต	13
3.1	ภาพโดยรวมของเทลเน็ต	13
3.1.1	ระดับชั้นของ TCP/IP ที่เทลเน็ตทำงานและ Port บริการ	13
3.1.2	หลักการการทำงานของเทลเน็ต	15
3.2	Network Virtual Terminal	15
3.2.1	โครงสร้างและหลักการของ NVT Network Virtual Terminal	15
3.2.2	การทำงานและ Option ของ NVT ในเทลเน็ต	16
3.3	Telnet Model	17
3.3.1	โครงสร้างของ Telnet Model และการทำงาน	17
3.3.2	Mode การส่งข้อมูลของ Telnet	19
3.3.2.1	Mode การส่งข้อมูลแบบ line-at-a-time	19
3.3.2.2	Mode การส่งข้อมูลแบบ Character-at-a-time	21
3.4	เทอร์มินอลเสมือน	21
3.5	คำสั่งและการควบคุมของเทลเน็ต	22
3.6	เทลเน็ตไคลเอนท์อัลกอริทึม	22
3.7	อัลกอริทึมของเทลเน็ตไคลเอนท์	23
3.8	เทอร์มินอล I/O ในยูนิกซ์	23
บทที่ 4	ซีจีไอ จาวาเบืองตันและเรียลออดีไอ	
4.1	ซีจีไอ (CGI : Common Gateway Interface)	25
4.1.1	หลักการทั่วไปของ CGI	25
4.1.1.1	ศัพท์เฉพาะของ CGI	25
4.1.2	หน้าที่ของ CGI สคริปต์	25
4.1.3	ประเภทของภาษาที่สามารถนำมาใช้เป็น CGI	27
4.1.3.1	ภาษาสคริปต์ที่นำมาใช้งานเป็น CGI	27
4.1.3.2	การบอกลักษณะข้อมูลที่ใส่ส่ง	28
4.1.3.3	มาตรฐานของตัวแปร CGI	29

4.1.3.3.1 ข้อมูลเกี่ยวกับ User	30
4.1.3.3.2 ข้อมูลเกี่ยวกับ server	31
4.1.3.3.3 ตัวอย่างการอ่านข้อมูล Environment Variables	32
4.1.3.3.4 ตัวอย่างการอ่านข้อมูล Environment Variables ใน Perl	33
4.1.3.3.5 ตัวอย่างการอ่านข้อมูล Environment Variables ใน C	34
4.1.4 การสร้างแบบฟอร์ม	35
4.1.4.1 ข้อมูลชนิด TEXT	36
4.1.4.2 ข้อมูลชนิด CHECKBOX	37
4.1.4.3 ข้อมูลชนิด RADIO	37
4.1.4.4 การส่งแบบฟอร์ม (SUBMIT)	37
4.1.4.5 การยกเลิกแบบฟอร์ม (RESET)	37
4.1.4.6 การสร้างปุ่มตัวเลือก <SELECTION>	37
4.1.4.7 ข้อมูลชนิด TEXTAREA	38
4.1.4.8 การรับข้อมูลจาก server	38
4.1.4.9 การส่งแบบฟอร์มไปยัง โปรแกรม CGI	39
4.1.4.10 รูปแบบการเข้ารหัสของข้อมูล	39
4.1.5 ประเภทของข้อมูล CGI ที่ส่งไปยังไคลเอนท์	39
4.2 จาวาเบื้องต้น	40
4.2.1 จาวาคืออะไร	40
4.2.2 คำศัพท์เกี่ยวกับโอโอพีทีที่ควรทราบ	41
4.2.3 จาวาแอปเพล็ต	43
4.3 เรียลออดิโอ (Real Audio)	44
4.3.1 การใช้งานระบบเรียล	45
4.3.2 โปรแกรมเรียล พับลิเชอร์ และ โปรแกรมเรียล เอ็น โคเดเจอร์	46
บทที่ 5 การออกแบบ	47
5.1 ระบบปฏิบัติการและเว็บเซิร์ฟเวอร์ที่ใช้	47
5.2 การออกแบบส่วนโฮมเพจหลัก	47
5.3 การออกแบบส่วนแนะนำและบริการต่างๆ	49
5.4 การออกแบบส่วนค้นหาหนังสือ	52
5.4.1 ลักษณะ โปรแกรม Telnet ของห้องสมุด	52
5.4.2 หลักการทำงานของโปรแกรม	56
5.5 การออกแบบการกระจายเสียงผ่านทางเครือข่ายอินเทอร์เน็ต	58
บทที่ 6 การทดลองและสรุปผลการทดลอง	59
6.1 การทดลองส่วนแนะนำหนังสือและบริการ	59

6.2 การทดลองส่วนของโปรแกรมเทลเนท	61
บทที่ 7 สรุปและวิจารณ์	64
ภาคผนวก ก	65
กิตติกรรมประกาศ	68
เอกสารอ้างอิง	69



สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 OSI โมเดล 7 เลเยอร์	5
รูปที่ 2.2 เลเยอร์ของโปรโตคอล TCP/IP	6
รูปที่ 2.3 แสดงหมายเลข IP แอดเดรส	7
รูปที่ 2.4 แสดงคลาส, จำนวนเครือข่าย และจำนวน โฮสต์ของแต่ละคลาส	8
รูปที่ 2.5 แสดงรูปแบบของ IP แอดเดรสในคลาสต่างๆ	8
รูปที่ 2.6 แสดงตัวอย่างหมายเลข IP แอดเดรสทั้งสองแบบ	9
รูปที่ 2.7 การทำซับเน็ตติ้ง	9
รูปที่ 2.8 ความสัมพันธ์ระหว่างชุดโปรโตคอล	10
รูปที่ 3.1 ลำดับการเอ็นแคปซูลชั้น	13
รูปที่ 3.2 แสดงรายละเอียดของ TCP message	14
รูปที่ 3.3 แสดงรายละเอียดของ IP packet	14
รูปที่ 3.4 แสดง โครงสร้าง NVT	16
รูปที่ 3.5 แสดงการทำงานเทลเน็ตโมเดล	17
รูปที่ 3.6 ขั้นตอนการ Connect	18
รูปที่ 3.7 การส่งข้อมูลแบบ line-at-a-time	20
รูปที่ 3.8 การส่งข้อมูลแบบ character-at-a-time	21
รูปที่ 3.9 เทลเน็ตไคลเอนท์อัลกอริทึม	22
รูปที่ 3.10 เทอร์มินอลดีไวซ์ไคร์เวอร์	24
รูปที่ 4.1 แสดงการติดต่อ Client-Server และ CGI	26
รูปที่ 4.2 กราฟแสดงการใช้งานภาษาต่างๆ ในงาน CGI	28
รูปที่ 4.3 แสดงการกำหนดชนิดข้อมูลใน Netscape	29
รูปที่ 4.4 โปรแกรมเรียลเพลเยอร์	45
รูปที่ 5.1 โฮมเพจหลักของสำนักหอสมุดกลาง	49
รูปที่ 5.2 Flowchart แสดงหลักการทำงานของโปรแกรมรับข้อมูลจากผู้ใช้	51
รูปที่ 5.3 Flowchart แสดงหลักการทำงานของโปรแกรมผู้ดูแลระบบ	52
รูปที่ 5.4 แสดงรูปแบบการสื่อสารของ innopac	53
รูปที่ 5.5 แสดง code ของการสื่อสารของ Telnet	53
รูปที่ 5.6 แสดง Flow chart ของโปรแกรม Telnet ที่เราทำการเขียนขึ้น	54
รูปที่ 5.7 แสดง Packet ของ Telnet	56

รูปที่	หน้า
รูปที่ 5.8 หน้าที่ทำให้ผู้ใช้บริการได้เลือกฟังระบบเรียล	58
รูปที่ 6.1 แสดงโฮมเพจส่วนรับข้อมูลจากผู้ใช้	60
รูปที่ 6.2 แสดงโฮมเพจส่วนผู้ดูแลระบบในส่วนลบคำแนะนำหนังสือ	60
รูปที่ 6.3 แสดงโฮมเพจส่วนผู้ดูแลระบบในส่วนแก้ไขคอนฟิค	61
รูปที่ 6.4 แสดงให้เห็นว่าโปรแกรมเราสามารถ connect ได้แล้ว	61
รูปที่ 6.5 แสดงหน้าจอ Telnet ที่เราทำการ Autologin เข้าไป connect	62
รูปที่ 6.6 หน้าจอหลังจากที่เราทำการ ใส่ข้อมูลรายชื่อผู้แต่งที่เราต้องการ	63



สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 4.1 ระบบรักษาความปลอดภัยของจาวา	43



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันระบบอินเทอร์เน็ตมีบทบาทอย่างกว้างขวางในการทำงานต่างๆ เป็นอย่างมาก จนถึงกับว่าอินเทอร์เน็ตนั้นเป็นส่วนหนึ่งของชีวิตประจำวันไปแล้ว เป็นที่ทราบกันดีว่าอินเทอร์เน็ตนั้นเป็นการติดต่อสื่อสารที่มีความรวดเร็วและในปัจจุบันการนำมาใช้ง่ายกว่าแต่ก่อนมาก และมีความทันสมัยมากยิ่งขึ้นด้วย จึงเป็นสิ่งที่เหมาะสมอย่างยิ่งในการนำเสนอข่าวสารหรือสิ่งต่างๆ ออกไปให้ทั่วโลกได้รับรู้

สำหรับสำนักหอสมุดกลาง สถาบันเทคโนโลยี พระจอมเกล้าลาดกระบัง ก็เช่นเดียวกัน ได้มีการนำอินเทอร์เน็ตมาใช้ในการบริการบ้างแล้วในบางส่วน แต่ยังไม่ดีเท่าที่ควรในโครงการนี้จึงเป็นการนำเอาอินเทอร์เน็ตเข้ามาพัฒนาทางด้านบริการต่างๆ ของสำนักหอสมุดกลางให้มีประสิทธิภาพในการบริการให้มากยิ่งขึ้น

1.2 วัตถุประสงค์

1. เพื่อให้สำนักหอสมุดกลาง สถาบันเทคโนโลยี พระจอมเกล้าลาดกระบัง ได้มีการนำอินเทอร์เน็ตมาใช้ปรับปรุงงานด้านบริการให้มีประสิทธิภาพมากยิ่งขึ้น
2. เพื่อเรียนรู้การสร้างโฮมเพจให้เป็นไปอย่างมีแบบแผน
3. เพื่อเรียนรู้หลักการทำงานของโปรโตคอล TCP/IP และเทลเนท
4. เพื่อเรียนรู้การเขียนโปรแกรมติดต่อบนระบบเครือข่ายด้วยภาษาจาวา
5. เพื่อเรียนรู้การเขียนโปรแกรมภาษา Perl ด้วยวิธีการทำงานของ CGI
6. เพื่อเรียนรู้การส่งเสียงและภาพแบบเรียลไทม์ ผ่านระบบเครือข่าย โดยใช้โปรแกรม

Real Audio

1.3 ขอบเขต

1. โฮมเพจใหม่ของสำนักหอสมุดกลาง ซึ่งเป็นภาษาไทย
2. โปรแกรมซึ่งเขียนโดยภาษาจาวา ทำให้สามารถค้นหาหนังสือที่มีอยู่ในสำนักหอสมุดกลางได้
3. โปรแกรมซึ่งเขียนโดยภาษา Perl ทำให้ผู้ที่มาใช้บริการโฮมเพจของทางสำนักหอสมุดกลาง ได้ทำการและนำหนังสือและบริการที่ผู้ใช้ต้องการ มาให้ยังสำนักหอสมุดกลางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โปรแกรมซึ่งเขียนโดยภาษา Perl ทำให้เจ้าหน้าที่ทางสำนักหอสมุดกลางได้ปรับปรุงข่าวสารที่จะใส่ไว้ในโฮมเพจได้ง่าย โดยไม่ต้องไปแก้ไขในไฟล์ HTML แต่จะแก้ไขในเท็กซ์ไฟล์แทน
5. พระราชดำรัสและเพลงพระราชนิพนธ์ ซึ่งจัดทำในรูปแบบของ Real Audio ให้แก่ผู้ที่สนใจทั่วไป

1.4 แนวทางการพัฒนา

โครงการนี้เป็นโครงการซึ่งถูกพัฒนาขึ้นมาโดยใช้เครื่องมือในการพัฒนาหลายชนิดมาก เพราะมีส่วนงานที่ต้องพัฒนาหลายด้าน เครื่องมือที่ใช้จึงถูกเลือกให้เหมาะสมกับงานที่พัฒนาในแต่ละด้าน ซึ่งจะใช้เครื่องมือในการพัฒนากับงานต่างๆ ดังนี้

1. Windows NT Server เป็นระบบปฏิบัติการในการติดตั้งเว็บเซิร์ฟเวอร์
2. Internet Information Server สำหรับเป็นโปรแกรมเว็บเซิร์ฟเวอร์
3. Homesite และ MS-fontpage สำหรับงานตกแต่งโฮมเพจ
4. Photoshop สำหรับงานตกแต่งรูปภาพ
5. Java Visual Café และ JDK สำหรับงานการเขียนโปรแกรมเกี่ยวกับจาวา
6. Perl for Win32 สำหรับงานแนะนำหนังสือและบริการ และงานควบคุมการแก้ไขข่าวสารของเจ้าหน้าที่
7. Real Player และ Real Audio Encoder สำหรับงานบันทึกเสียง

แนวทางการพัฒนานั้นเริ่มจากการวิเคราะห์ข้อมูลเกี่ยวกับงานของสำนักหอสมุดกลาง และวิเคราะห์ความต้องการ รวมถึงความเป็นไปได้ที่จะนำอินเทอร์เน็ตมาประยุกต์ใช้กับงานของทางสำนักหอสมุดกลาง ต่อจากนั้นเป็นการเลือกเครื่องมือในการพัฒนาดังที่ได้กล่าวไปแล้ว เมื่อได้เครื่องมือในการพัฒนาแล้วต่อไปจึงเป็นการจัดทำงานแต่ละส่วน และสุดท้ายจึงเป็นการทดลองใช้งาน ของงานที่ได้จัดทำขึ้นเพื่อไปปรับปรุงส่วนที่บกพร่อง

โครงการนี้สามารถเข้าไปเยี่ยมชมได้ที่ <http://www.lib.kmitl.ac.th/>

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับระบบเครือข่าย

2.1 ระบบเครือข่าย และการพัฒนาโปรแกรมระบบเครือข่าย

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควรในหัวข้อนี้จะอธิบายกับศัพท์พื้นฐาน และหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายดังกล่าวต่อไป

2.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์ และอินเทอร์เน็ตเวิร์กกิง

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End-System) ซึ่งระบบปลายทางเป็นระบบที่เป็นอิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นได้ ตั้งแต่ไมโครคอมพิวเตอร์ ไปจนกระทั่งซูเปอร์คอมพิวเตอร์ ขนาดใหญ่เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งปันทรัพยากรของระบบเช่น ไฟล์ ,ข้อมูล , เครื่องพิมพ์ (Printer) , โมเด็ม ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing database) อินเทอร์เน็ตเวิร์กกิง หรืออินเทอร์เน็ต คือการเชื่อมต่อของระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้น คอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์ระบบเครือข่ายอื่นๆได้ เช่น เน็ตเวิร์คโทเคนริงค์ เชื่อมกับเน็ตเวิร์คอีเทอร์เน็ต โดยมีเกตเวย์เป็นตัวเชื่อม

2.1.2 รูปแบบของโปรแกรมบนระบบเครือข่าย (Network Programming Models)

จากหัวข้อ 2.1.1 ได้ให้ความหมายของระบบเครือข่าย แสดงถึงวิธีการที่ระบบคอมพิวเตอร์ใดๆ จะทำการเชื่อมต่อกันระบบเครือข่าย แต่ว่าคอมพิวเตอร์ที่เชื่อมต่ออยู่กับระบบเครือข่ายนั้นมีวิธีการอย่างไรในการแลกเปลี่ยน ข้อมูลและแบ่งปันทรัพยากรเหล่านั้นได้ทำให้ต้องมีโปรแกรมประยุกต์ ซึ่งสามารถที่จะจัดการในสิ่งที่กล่าวมาแล้ว อย่างเหมาะสม ซึ่งในหัวข้อนี้จะกล่าวถึงรูปแบบของโปรแกรมบนระบบเครือข่าย 2 รูปแบบคือ โคลเอนต์เซิร์ฟเวอร์ และการประมวลผลแบบกระจาย (Distributed Computing)

2.1.2.1 การประมวลผลแบบโคลเอนต์เซิร์ฟเวอร์

การประมวลผลแบบโคลเอนต์เซิร์ฟเวอร์นี้การประมวลผลของโปรแกรมประยุกต์จะแบ่งออกเป็น 2 ส่วนคือ ส่วนฟรอนต์เอนด์ที่ทำงานบนโคลเอนต์ ส่วนนี้จะทำหน้าที่แสดงผลที่ได้จากการประมวลผล และรับข้อมูลจากผู้ใช้ อีกส่วนหนึ่งคือแบ็กเอนด์ทำงานบนเซิร์ฟเวอร์มีหน้าที่ในการเก็บรวบรวม และจัดการข้อมูลจากฟรอนต์เอนด์ในรูปแบบการประมวลผล โคลเอนต์เซิร์ฟเวอร์นี้ เครื่องเซิร์ฟเวอร์นี้มักจะเป็นเครื่องที่มีความสามารถสูงกว่าเครื่องโคลเอนต์ โดยปกติเครื่องเซิร์ฟเวอร์มักจะเป็นเครื่องเมน

เฟรม หรือมินิคอมพิวเตอร์และเครื่องไคลเอนต์มักจะเป็นเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งการติดต่อกันระหว่างส่วนฟรอนต์เอนด์และแบ็คเอนด์ ทำโดยผ่านระบบเครือข่ายซึ่งในทางปฏิบัติแล้ว ในส่วนของแบ็คเอนด์ที่อยู่บนเซิร์ฟเวอร์ จะเป็นฝ่ายให้บริการแก่งานของไคลเอนต์หลายงานในเวลาเดียวกัน

2.1.2.2 การประมวลผลแบบกระจาย (Distributed Computing)

การประมวลผลของโปรแกรมประยุกต์มีรูปแบบการประมวลผล 2 แบบ คือ 프리คอลเล็คชัน (Precollection) และการประมวลผลแบบขนาน (Parallel processing) ดังนี้

1. 프리คอลเล็คชันเป็นลักษณะการทำงานที่ข้อมูลที่ต้องการในการจัดเก็บ และส่งตัวต่อไปที่ระบบเครือข่ายอยู่ตลอดเวลาอย่างสม่ำเสมอ การทำงานในลักษณะนี้จะเหมาะสมกับงานบางงาน เช่น ต้องการเก็บสถานะของคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายหนึ่งๆ ทุกเครื่อง

2. การประมวลผลแบบขนาน การประมวลผลลักษณะนี้งานใดๆ จะถูกประมวลผลด้วยคอมพิวเตอร์หลายๆ เครื่องโดยเครื่องคอมพิวเตอร์เหล่านั้นสามารถจะติดต่อกัน โดยระบบเครือข่าย เช่น การทำการพัฒนาโปรแกรมประยุกต์ขนาดใหญ่ โดยทีมพัฒนาที่มีผู้พัฒนาหลายคน สามารถลดเวลาของการแปล และการรวมโมดูลต่างๆ เข้าเป็นโปรแกรมเดียวกัน โดยการแบ่งงานการแปลโมดูลเหล่านั้นแก่คอมพิวเตอร์ในระบบเครือข่ายทำการแปลในเวลาเดียวกัน

2.1.3 โมเดล OSI

OSI เป็นคำย่อที่มาจากคำว่า Open System Interconnection โดยที่เป็นมาตรฐานที่ถูกเสนอขึ้นโดย International Standards Organization ซึ่งเป็นองค์กรที่จัดตั้งขึ้นมาเพื่อดูแล และส่งเสริม ตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรม แบบระบบเปิด (Open System) เพราะมุ่งที่จะให้ระบบคอมพิวเตอร์ในหลายๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโปรโตคอลในการสื่อสารออกเป็น 7 เลเยอร์

โปรโตคอล คือชุดของกฎ หรือข้อตกลงในการติดต่อ ข้อสังเกตโมเดล OSI เป็นเพียงข้อเสนอแนะ มิใช่ข้อกำหนด และควรรู้ว่ายังไม่มียุทธศาสตร์เชื่อมต่อใดที่สร้างเหมือนกับโมเดล OSI จริงๆ ในหนึ่งชั้นของเลเยอร์ไม่ได้กำหนดว่าจะต้องมีเพียงหนึ่งโปรโตคอลเท่านั้น ที่อยู่ในระดับเลเยอร์เดียวกัน และในทางตรงกันข้าม ชุดของโปรโตคอลใดๆ อาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกับเป็นข้อกำหนดของระบบเครือข่ายเรียกว่าชุดโปรโตคอล (Protocol Suite) เช่น ชุดโปรโตคอล TCP/IP (Transmission Control/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเป็นเลเยอร์ คือการกำหนดการติดต่อระหว่างเลเยอร์ ทำได้โดยไม่ต้องคำนึงถึงการเปลี่ยนในเลเยอร์ใดๆ ที่ติดกัน

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

2.1.4 ลักษณะของการติดต่อ

การติดต่อบนเน็ตเวิร์กนั้นแบ่งได้ออกเป็น 2 ประเภท คือ

2.1.4.1 Connection – Oriented

คือ การติดต่อที่ต้องการให้มีการทำการเชื่อมโปรเซสที่จะทำการติดต่อกันก่อนที่จะมีการส่งหรือการรับข้อมูล ซึ่งในที่นี้จะสามารถใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโปรเซส ถึงแม้ว่าข้อมูลนั้นอาจจะผ่าน Packet-Switching Network บริการชนิดนี้ส่วนมากนั้นจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งขั้นตอนการทำงานออกได้เป็น

- ขั้นตอนการสร้างการติดต่อ (Connection Establishment)
- ขั้นตอนการส่งผ่านข้อมูล (Data transfer)
- ขั้นตอนยกเลิกการติดต่อ (Connection Termination)

2.1.4.2 Connectionless หรือดาต้าแกรม

คือจะไม่มีการสร้างการติดต่อ และขั้นตอนการยกเลิกการติดต่อกันก่อน แต่จะมีขั้นตอนการส่งผ่านข้อมูลอย่างเดียว โดยข้อมูลที่ส่งนั้นเรียกว่าดาต้าแกรม โดยดาต้าแกรมนั้นจะถูกส่งจากระบบหนึ่งไปยังอีกระบบหนึ่งอย่างเป็นอิสระโดยที่ไม่ขึ้นอยู่กับดาต้าแกรมอื่นๆ

2.1.5 แอดเดรส

การที่ระบบต่างๆ ในเครือข่ายนั้นสามารถที่จะติดต่อกันได้นั้นจำเป็นต้องมีสิ่งที่เป็นที่อยู่สำหรับตำแหน่งที่จะติดต่อกันซึ่งเรียกว่าแอดเดรส แอดเดรสนั้นจะเหมือนกันกับหมายเลขประจำตัวของแต่ละที่ซึ่งลำดับของแอดเดรสสามารถพิจารณาได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แต่ละเครือข่ายจะต้องมีแอดเดรสสำหรับเครือข่าย
- คอมพิวเตอร์โฮสต์แต่ละเครื่องในเครื่องในเครือข่ายจะต้องมีแอดเดรส
- แต่ละโปรเซสในโฮสต์จะต้องมีหมายเลขประจำตัว

โดยทั่วไปแอดเดรสของโฮสต์จะประกอบด้วยหมายเลขเครือข่าย (Network ID) และหมายเลขของโฮสต์ (Host ID) ส่วนแอดเดรสของโปรเซสของผู้ใช้จะอยู่ในรูปจำนวนเต็มซึ่งกำหนดโดยโปรโตคอล เช่น โปรโตคอล TCP/IP จะใช้เลขจำนวนเต็มขนาด 32 บิต ในการกำหนดหมายเลขเครือข่ายและหมายเลขของโฮสต์ โดยทั้ง TCP และ UDP นั้นใช้เลขจำนวนเต็ม 16 บิต เป็นหมายเลขพอร์ตหรือหมายเลขของโปรเซส ชุดของโปรโตคอลส่วนใหญ่จะมีการกำหนดชุดของแอดเดรสสำหรับการบริการที่เป็นที่รู้จักกันโดยทั่วไป เช่น คอมพิวเตอร์ที่ใช้โปรโตคอล TCP/IP ส่วนใหญ่จะมี FTP (File Transfer Protocol) ซึ่งไคลเอนต์สามารถติดต่อได้โดยใช้หมายเลขพอร์ตคือ 21

2.2 ความรู้เบื้องต้นเกี่ยวกับโปรโตคอล TCP/IP

2.2.1 เปรียบเทียบระหว่างโปรโตคอล TCP/IP และ OSI โมเดล

การออกแบบโปรโตคอล TCP/IP นี้ไม่ได้เป็นไปตามรูปแบบของ OSI โมเดล เนื่องด้วยการถูกออกแบบโดยองค์กรขนาดใหญ่ซึ่งใช้เวลานานในการออกแบบทดลองจนการรับรองมาตรฐานต่างกับโปรโตคอล TCP/IP ที่ถูกออกแบบด้วยความต้องการอันเร่งด่วนของรัฐบาลสหรัฐ จึงทำให้การพัฒนาโปรโตคอล TCP/IP มีเงื่อนไขของด้านความต้องการที่ต่างจาก OSI โมเดล ซึ่งหากเรามองโดยรวมแล้วจะเห็นว่าโปรโตคอล TCP/IP มีการแบ่งเป็นเลเยอร์ที่น้อยกว่า OSI คือมีเพียงแค่ 4 ชั้นเท่านั้น ดังรูปที่ 2.2 โดยจะแบ่งได้เป็น

4. Application Layer
3. Transport Layer
2. Internet Layer
1. Physical Layer

รูปที่ 2.2 เลเยอร์ของโปรโตคอล TCP/IP

2.2.1.1 แอปพลิเคชันเลเยอร์

ในเลเยอร์นี้นั้นจะประกอบไปด้วยโปรแกรมประยุกต์ที่ใช้งานเครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูล (File Transfer Protocol) และอาจจะกล่าวได้ว่าเลเยอร์นี้โปรโตคอล TCP/IP ก็คือ เลเยอร์ใน

ชั้นแอปพลิเคชันเลเยอร์ ร่วมกับชั้นพีรีเซนเตชันเลเยอร์ใน OSI นั้นเอง และในเลเยอร์ชั้นนี้ของโปรโตคอล TCP/IP จะกลืนอยู่ในตัวโปรแกรมประยุกต์

2.2.1.2 ทรานสปอร์ตเลเยอร์

ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดต้นทางไปยังยังจุดปลายทาง หากเปรียบเทียบกับ OSI โมเดล ก็สามารถเทียบได้กับชั้นเซสชันเลเยอร์ร่วมกับทรานสปอร์ตเลเยอร์นั่นเอง โดยโปรโตคอล TCP/IP มีชื่อเรียกเป็นจุดปลายทางในการสื่อสาร ซึ่งชื่อเรียกนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์ และหมายเลขพอร์ตของเครื่องที่ต้องการส่งข้อมูลไปถึง ในชั้นนี้มีการรับรองการถึงที่หมาย และลำดับของข้อมูลที่ส่งโดยไม่ซ้ำ รวมถึงความผิดพลาดของข้อมูลอีกด้วย

2.2.1.3 อินเทอร์เน็ตเลเยอร์

เลเยอร์นี้ทำหน้าที่ในการกำหนดค่าตัวแปร และทำการหาเส้นทางที่ดีที่สุดในการส่ง หน้าที่ของเลเยอร์นี้เทียบเท่ากับเน็ตเวิร์คเลเยอร์ (Network Layer) และดาต้าลิงก์ (Data Link Layer) ของ OSI โมเดลนั่นเอง

2.2.1.4 ฟิสิกอลเลเยอร์

โปรโตคอล TCP/IP ไม่ได้กำหนดรูปแบบของการเชื่อมต่อในระดับนี้ไว้ใหม่ แต่ได้ใช้มาตรฐานเดิมที่มีอยู่ก่อนแล้ว เช่น RS232, อีเทอร์เน็ต (Ethernet) เป็นต้น

2.2.2 รูปแบบการกำหนดแอดเดรสของโปรโตคอล TCP/IP

ในหัวข้อนี้จะได้อธิบายรูปแบบการกำหนดแอดเดรสของโปรโตคอล TCP/IP ซึ่งลักษณะแอดเดรสของโปรโตคอลนี้ค่าของแอดเดรสของเครื่องคอมพิวเตอร์ในระบบเครือข่ายจะไม่ซ้ำกัน โดยเลขนี้เรียกว่า IP แอดเดรส (IP Address) เป็นเลข 32 บิต ซึ่งแบ่งเป็นคลาส (Class) ตามหลักการที่จะได้พิจารณาต่อไปนี้

2.2.2.1 การแบ่งเน็ตเวิร์กคลาส

เนื่องจากหมายเลขแอดเดรสของคอมพิวเตอร์เครื่องใดๆ นั้นจะต้องสามารถบอกถึงความแตกต่างระหว่างเครื่อง ตลอดจนเครือข่ายที่คอมพิวเตอร์นั้นเชื่อมต่ออยู่ด้วยหมายเลข IP แอดเดรส จึงแยกออกเป็น 2 ส่วน ได้แก่ ส่วนที่แสดงหมายเลขของเครื่องคอมพิวเตอร์โฮสต์ และส่วนที่เป็นหมายเลขของเครือข่าย การแบ่งคลาสของแอดเดรสทำได้โดยการพิจารณาจำนวนบิตของ 2 ส่วนประกอบข้างต้นที่ได้กล่าวไป ซึ่งมีการแบ่งออกเป็น 5 คลาส แต่มีการใช้เพียง 3 คลาสแรก คือ คลาส A, คลาส B, คลาส C ส่วนคลาส D และคลาส E นั้นถูกสงวนไว้สำหรับจุดประสงค์พิเศษ จะให้เห็นดังรูป

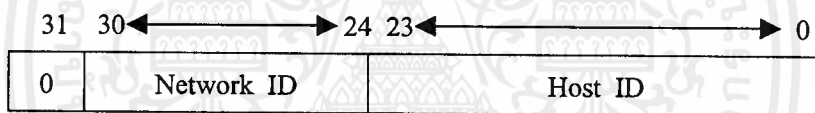
Class ID	Network ID	Host ID
----------	------------	---------

รูปที่ 2.3 หมายเลข IP แอดเดรส

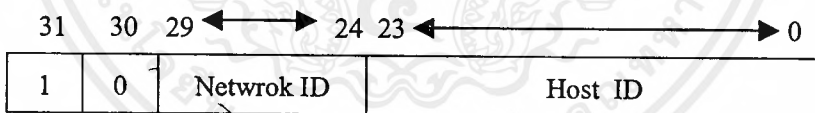
Network Class	Networks	Host per Network
A	126	16,777,214
B	16,382	65,534
C	2,097,150	254

รูปที่ 2.4 แสดงคลาส,จำนวนเครือข่าย และจำนวนโฮสต์ของแต่ละคลาส

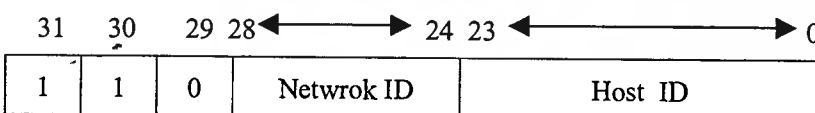
ซึ่งโดยปกติผู้พัฒนาโปรแกรมนั้นไม่จำเป็นต้องสนใจความแตกต่างระหว่างคลาสของ IP แอดเดรส



Class A IP Address format



Class B IP Address format



Class C IP Address format

รูปที่ 2.5 แสดงรูปแบบของ IP แอดเดรสในคลาสต่างๆ

2.2.2.2 การแทนด้วยเลขฐานสิบและจุด (Dotted Decimal Notation)

เนื่องจากการแทนหมายเลข IP แอดเดรสเป็นเลขฐาน 2 ซึ่งค่อนข้างอ่านไม่สะดวกจึงมีการแทนเลขฐานสองเหล่านั้นในรูปแบบเลขฐานสิบและจุด โดยเลขฐานสิบแต่ละตัวจะแทนเลขฐานสองเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวน 8 บิต โยระหว่างเลขฐานสิบแต่ละตัวจะแทรกด้วยจุดดังนั้นจะต้องใช้เลขฐานสิบ 4 ตัว ในการแทนเลข 32 บิต ที่เป็น IP แอดเดรส ดังตัวอย่างจะสังเกตว่าหมายเลข IP แอดเดรส ดังตัวอย่างจะสังเกตว่าหมายเลข IP แอดเดรสนี้จัดอยู่ในคลาส B โดยหมายเลขของเครือข่ายคือ 166.78 และหมายเลขประจำเครื่องคือ 4.139 นั่นเอง

Dotted Decimal Notation			
166.78.4.139			
Binary Representation			
10100110	01001110	00000100	10001011

รูปที่ 2.6 แสดงตัวอย่างหมายเลข IP แอดเดรสทั้งสองแบบ

2.2.2.3 การทำซับเน็ตติ้ง

การทำซับเน็ตติ้งเป็นการทำการเปลี่ยนแปลงการใช้หมายเลขของเครื่องโฮสต์และหมายเลขของเครือข่ายในระดับท้องถิ่น โดยในทางตรรก คือการเส้นแบ่งที่แยกหมายเลขเครื่อง และหมายเลขของเน็ตเวิร์กที่อยู่ในหมายเลข IP แอดเดรส โดยที่ปริมาณของหมายเลขเครื่องโฮสต์และหมายเลขเครือข่ายจะแปรผกผันกัน เช่น หากมีปริมาณของเน็ตเวิร์กมาก ก็จะทำให้เครื่องใดๆ ที่จะต่อกับระบบเครือข่ายหนึ่งๆ จะน้อยลงเป็นต้น ในทางปฏิบัติการทำซับเน็ตติ้งทำโดยการนำซับเน็ตมาร์ค (Subnet mark) คือตัวเลขจำนวน 32 บิต มาทำการกระทำทางตรรกและ (AND) กับหมายเลข IP แอดเดรส ดังตัวอย่างโดยกำหนดหมายเลข IP แอดเดรส คือ 166.78.47.139 และซับเน็ตมาร์ค คือ 255.255.255.0 ทำการกระทำตรรก ดังรูปที่ 2.7

166.78.4.139	
	AND
255.255.255.0	
166.78.4.0	

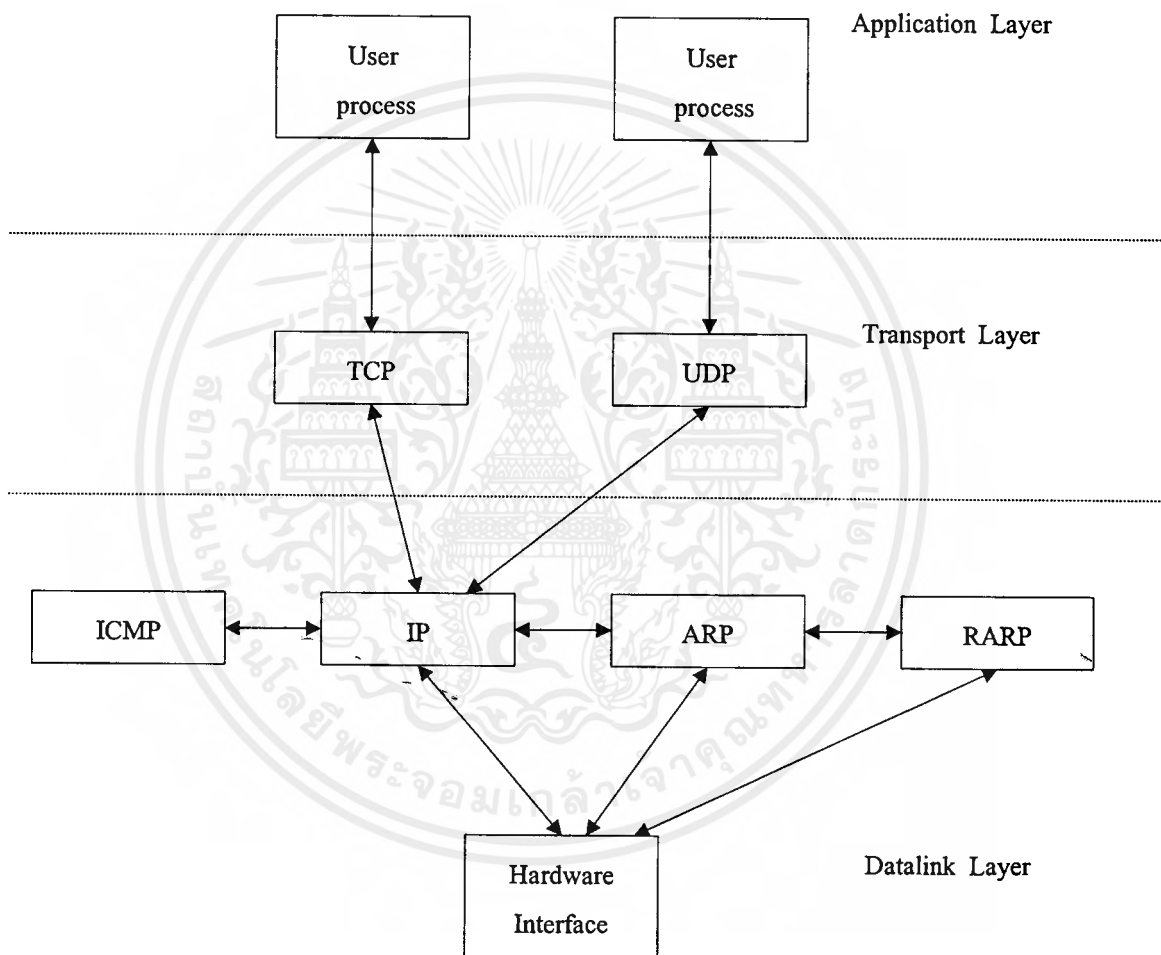
รูปที่ 2.7 การทำซับเน็ตติ้ง

จะเห็นว่าหากพิจารณาโดยไม่มีการทำซับเน็ตติ้งแล้วจะได้หมายเลขของเน็ตเวิร์กคือ 166.78 และหมายเลขประจำเครื่อง คือ 4.139 แต่ผลที่ได้จากการกระทำทางตรรกและ ในรูปข้างต้นผลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการศึกษาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิพท์ที่ได้คือ 166.78.4.0 และที่เหลือคือ 139 หากพิจารณาผลลิพท์นี้ คือหมายเลขเน็ตเวิร์กคือ 166.78.4.0 และหมายเลขเครื่อง คือ 139 หรืออาจพูดได้ว่าคอมพิวเตอร์เครื่องนี้มีหมายเลขเครื่องเท่ากับ 139 และอยู่บนเครือข่ายย่อยหมายเลข 166.78.4

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

ชุดโปรโตคอล TCP/IP นอกจากมีโปรโตคอล TCP และ IP แล้วยังมีโปรโตคอลอย่างอื่นอีก ดังภาพแสดงความสัมพันธ์ของชุดโปรโตคอลโดยแบ่งตามเลเยอร์



รูปที่ 2.8 ความสัมพันธ์ระหว่างชุดโปรโตคอล

2.2.3.1 โปรโตคอล IP

โปรโตคอล IP เป็นโปรโตคอลแบบคอนเนกชันเลสซึ่งได้กล่าวถึงลักษณะของโปรโตคอล ชนิดนี้แล้วในหัวข้อที่ 2.1.4 โดยที่โปรโตคอล IP นั้นไม่รับประกันว่าข้อมูลที่ส่งไปนั้นจะถึงยังปลายทางซึ่งแพคเกจของข้อมูลอาจจะไปถึงในลักษณะที่ผิดพลาด, ซ้ำกัน หรือไปไม่ถึงเลย โดยความน่าเชื่อถือของการส่งจะถูกควบคุมในโปรโตคอลในเลเยอร์ต่างๆ ไป การหาเส้นทางของข้อมูลจะทำในระดับของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการศึกษาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล IP นี้ โดยจะพิจารณาแต่ละแพคเกจ แยกออกจากกัน และยังมีหน้าที่ในการจัดเรียงข้อมูลใหม่ที่ปลายทางอีกด้วย

2.2.3.2 โปรโตคอล ARP (Address Resolution Protocol)

โปรโตคอลนี้ทำหน้าที่จับคู่ระหว่างหมายเลข IP แอดเดรสเข้ากับหมายเลขแอดเดรสทางฮาร์ดแวร์ โดยโปรโตคอลนี้ทำหน้าที่ในการส่งข้อความไปทั้งเครือข่ายท้องถิ่น ซึ่งข้อความนี้เป็นลักษณะข้อความที่ตรวจสอบได้ว่ามีคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกับที่ต้องการหาหรือไม่ หากคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกันนั้นได้รับข้อความนี้ก็จะตอบกลับและเป็นหน้าที่สังเกตว่าโปรโตคอลนี้ทำงานได้กับระบบเครือข่ายท้องถิ่นเท่านั้น เพราะว่าโครงสร้างหมายเลขทางฮาร์ดแวร์ของเครื่องคอมพิวเตอร์จะขึ้นอยู่กับชนิดของเครือข่าย

2.2.3.3 โปรโตคอล ICMP (Internet Control Message Protocol)

เป็นโปรโตคอลที่ทำหน้าที่ในการจัดการเกี่ยวกับข่าวสารความผิดพลาดและการควบคุมเครือข่าย และเครื่องคอมพิวเตอร์ในระบบเครือข่าย

2.2.3.4 โปรโตคอล RARP (Reverse Address Resolution Protocol)

เป็นโปรโตคอลที่ทำหน้าที่ในการจับคู่ระหว่างหมายเลขของฮาร์ดแวร์กับหมายเลข IP แอดเดรส หรือทำงานกลับกันกับโปรโตคอล ARP

2.2.3.5 โปรโตคอล UDP (User Datagram Protocol)

โปรโตคอลนี้เป็นโปรโตคอลที่อยู่ในระดับทรานสปอร์ตเลเยอร์ และมีความสำคัญเพราะว่าเป็นโปรโตคอลที่ผู้พัฒนาโปรแกรมสามารถใช้ได้โดยตรง โปรโตคอลนี้เป็นโปรโตคอลแบบคอนเนกชันเลสมีความน่าเชื่อถือต่ำ ไม่รับรองว่าข้อมูลที่ส่งไปจะถึงปลายทางหรือไม่ และอาจซ้ำซ้อน หรือผิดพลาดได้ แต่ข้อดีของโปรโตคอลนี้ คือค่าความสิ้นเปลือง (Overhead) ที่ต่ำนั่นเอง

2.2.3.6 โปรโตคอล TCP (Transmission Control Protocol)

โปรโตคอลนี้อยู่ในระดับทรานสปอร์ตเลเยอร์ เหมือนกับโปรโตคอล UDP แต่มีลักษณะที่ตรงกันข้ามกันคือ เป็นโปรโตคอลแบบคอนเนกชันออเรียนเต็ด โดยจะมีความน่าเชื่อถือในการรับ ส่งข้อมูลและลำดับของข้อมูลจะมีลำดับเหมือนกับต้นทางและเนื้อข้อมูลจะไม่มีผิดพลาด จึงทำให้เกิดความสิ้นเปลืองในการเชื่อมต่อของการส่งข้อมูลมากกว่าโปรโตคอล UDP

2.2.4 หมายเลขพอร์ต

เนื่องจากในเวลาใดๆ สามารถมีโพรเซสของผู้ใช้ UDP หรือ TCP ได้พร้อมกัน ดังนั้นจึง

ต้องมีวิธีแยกแยะว่าข้อมูลเป็นของผู้ใช้คนใด ซึ่งวิธีที่ TCP และ UDP ใช้คือการใช้หมายเลขพอร์ตเมื่อโพรเซสรับข้อมูลหรือส่งข้อมูลให้ระบบปฏิบัติการเพื่อที่จะนำข้อมูลนั้นไปส่งให้กับโพรเซสที่ต้องการ เมื่ออยู่ใต้เห็นไปจะระบุชื่อที่ทำการรับส่งข้อมูลทุกครั้งที่มีการนำไปใช้

เซสโคลเอนต์ที่ต้องการที่จะติดต่อกับเซิร์ฟเวอร์โคลเอนต์จะต้องเจาะจง เซิร์ฟเวอร์ที่ต้องการติดต่อกัน แต่ ถ้าทั้งรู้อัดเครสอินเตอร์เน็ต 32 บิตเพียงอย่างเดียวนั้นไม่เพียงพอ เพราะว่าสามารถติดต่อกับโฮสต์ได้ เพียงอย่างเดียวแต่ไม่สามารถเจาะจงโพรเซสที่จะทำการติดต่อกันได้ ดังนั้นเพื่อแก้ปัญหาที่ทั้ง TCP และ UDP ได้มีการกำหนด หมายเลขพอร์ตมาตรฐาน (well-know ports) ซึ่งเป็นที่รู้จักกันเช่น ทุกๆ ระบบ TCP/IP ที่มีเซิร์ฟเวอร์ FTP (File transfer Protocol) จะมีหมายเลขพอร์ตเป็น 21 เป็นต้น เมื่อ TCP หรือ UDP กำหนดหมายเลขพอร์ตที่ไม่ซ้ำกันให้โพรเซสของผู้ใช้ เราเรียกหมายเลข พอร์ตนี้ว่าหมายเลขพอร์ต ชั่วคราว (Ephemeral Port Numbers) เมื่อโคลเอนต์เล็กใช้หมายเลขพอร์ตนี้แล้ว สามารถกำหนดหมายเลขพอร์ตนี้ให้โคลเอนต์อื่นได้ โพรเซสที่ได้รับหมายเลขพอร์ตชั่วคราวนี้จะไม่สนใจว่ามีค่าเท่าใด แต่เป็น หน้าทีของอีกโพรเซสหนึ่งที่ต้องกันที่ตองสนใจ เพราะต้องส่งข้อมูลกลับมาที่พอร์ตนี้ ใน TCP และ UDP นั้นหมายเลขพอร์ตตั้งแต่ 1-1023 เป็นพอร์ตที่สงวนไว้สำหรับหมายเลขพอร์ตมาตรฐาน



บทที่ 3

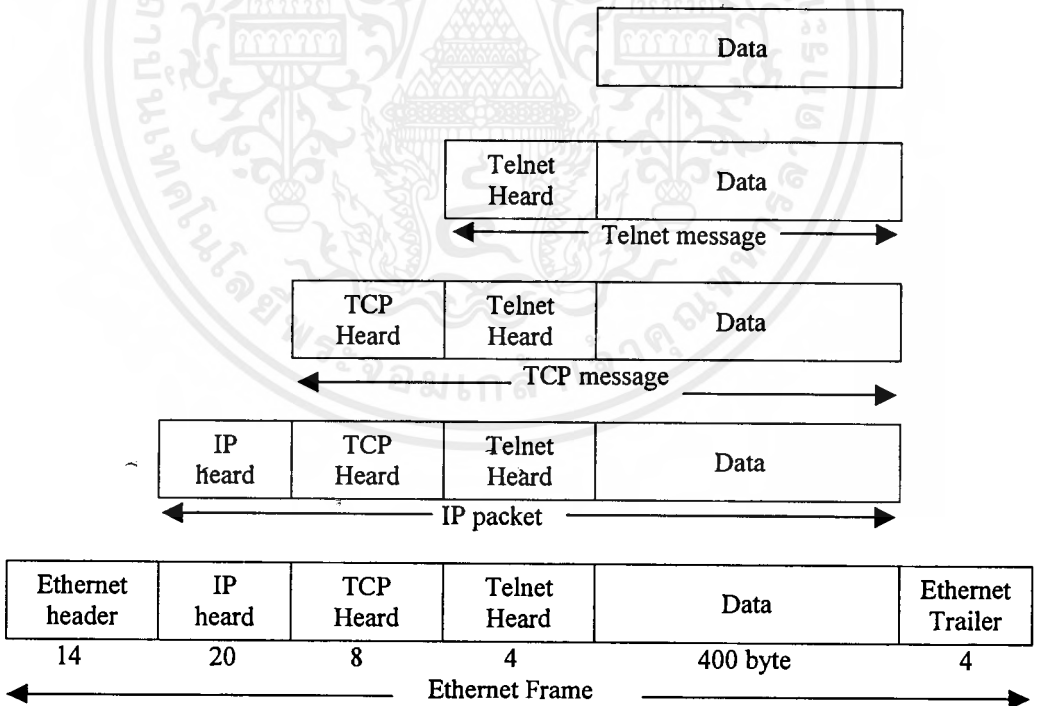
เทลเน็ต

3.1 ภาพโดยรวมของเทลเน็ต

เทลเน็ต คือ โปรแกรมประยุกต์ที่ทำให้ผู้ใช้สามารถล็อกอินผ่านเทอร์มินอลใดๆเพื่อขอใช้บริการจากโฮสต์ใดๆ ที่เป็น remote machine และใช้โปรโตคอล TCP/IP ในการติดต่อระหว่างเทอร์มินอลกับโฮสต์นั่นเอง

3.1.1 ระดับชั้นของ TCP/IP ที่เทลเน็ตทำงานและ Port บริการ

เทลเน็ตเป็นโปรแกรมประยุกต์ที่ทำงานอยู่บนชั้นแอปพลิเคชัน ของโปรโตคอล TCP/IP ดังนั้นข้อมูลจากผู้ใช้ ก่อนที่จะส่งออกไปในระบบเครือข่ายจะต้องมีการเพิ่มส่วนหัวในชั้นต่างๆ ดังรูป



รูปที่ 3.1 ลำดับการเอ็นแคปซูลเลขัน

ส่วนรายละเอียดของ TCP message มีดังนี้

16-bit Source Port number		16-bit Destination Port number	
32-bit Sequence number			
32-bit Acknowledgement number			
4-bit Header Length	6-bit Reserved	6-bit Flags	16-bit Windows Size
16-bit TCP check sum		16-bit Urgent pointer	
Option (if any) & Padding			
Data (Variable length, if any) (Telnet message)			

รูปที่ 3.2 แสดงรายละเอียดของ TCP message

ส่วนรายละเอียด IP Packet มีดังนี้

4-bit Version	4-bit Header Length	8-bit Type of Service	16-bit Total Length	
16-bit Identification		3-bit Flags	13-bit Fragment Offset	
8-bit Time to Live	8-bit Protocol	16-bit Header Checksum		
32-bit Source Address				
32-bit Destination Address				
Option (if any) & Padding				
Data (Variable length, if any) (TCP message)				

รูปที่ 3.3 แสดงรายละเอียดของ IP Packet

ส่วน พอร์ต บริการของเทลเน็ตนั้น โดยทั่วไปแล้วจะอยู่ที่พอร์ตบริการเบอร์ 23 ดังนั้น การทำคอนเนคชันจะต้องทำคอนเนคชันกับพอร์ต 23 ที่โฮสต์เสมอที่มีการใช้โปรแกรมเทลเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 หลักการทำงานของเทลเน็ต

หลังจากที่ผู้ใช้พิมพ์คำสั่ง “telnet < host id หรือ host name >” แล้วมีเหตุการณ์อะไรเกิดขึ้น ?

1. ที่เครื่องที่ผู้ใช้ (Client) จะทำการเริ่มโปรแกรมเทลเน็ต
2. เทลเน็ตจากเครื่องผู้ใช้จะพยายามคอนเนคกับเครื่องให้บริการ (Host) ถ้าคอนเนคได้เครื่องให้บริการจะมีการ ACK กลับถ้าไม่ได้ก็จะแจ้งผิดพลาดแล้ว ABORT โปรแกรมไป
3. ถ้าคอนเนคได้แล้ว เครื่องผู้ใช้ก็จะขอทำ Remote Login กับเครื่องให้บริการโดยถ้าเครื่องให้บริการพร้อมให้ Login และก็จะส่ง ACK ตอบกลับมา พร้อมทั้งส่ง Escape Character ที่จะใช้ในการทำเทอร์มินอลเสมือน
4. เครื่องผู้ใช้ ACK กลับมาแล้วพร้อมรอข้อมูลจากผู้ใช้ (Loginname, password) แล้วส่งข้อมูลเหล่านั้นไปให้เครื่องให้บริการ
5. เครื่องให้บริการตอบกลับว่า Login สำเร็จหรือไม่โดยถ้าทำสำเร็จก็จะเข้า Shell process

3.2 Network Virtual Terminal

เนื่องจากเทลเน็ตในทางปฏิบัติแล้ว เมื่อผู้ใช้ทำการล็อกอินไปยังโฮสต์ใดๆ ก็ตามเทลเน็ตจะทำการล็อกอินผ่าน NVT เพื่อให้ผู้ใช้ได้ใช้เทอร์มินอลของตน ล็อกอินเข้าไปใช้ระบบและดีไวซ์ของเครื่องโฮสต์ได้

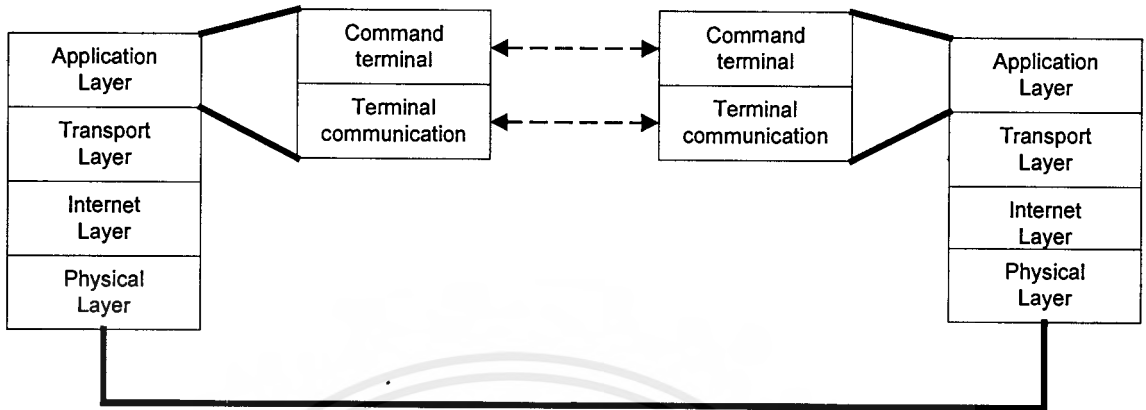
3.2.1 โครงสร้างและหลักการของ NVT Network Virtual Terminal (NVT)

NVT เป็นหลักการที่ใช้ส่งข้อมูลระหว่าง terminal application ที่อยู่ต่างๆ เครื่องกันและต่างสิ่งแวดล้อมกัน ได้แก่ ความแตกต่างในเรื่อง OS ตลอดไปจนถึงความแตกต่างในเรื่องโปรโตคอลที่ใช้ในระดับ Upper – Level ของแอปพลิเคชันที่รันอยู่ต่างเครื่องกันนั้น ซึ่งด้วยหลักการของ NVT นี้ จะทำให้แอปพลิเคชันที่รันอยู่บนเครื่องให้บริการซึ่งมีสิ่งแวดล้อมที่ต่างกัน สามารถติดต่อกันได้ซึ่งเปรียบเสมือนโปรแกรมเทลเน็ตที่รันอยู่บนเครื่องให้บริการที่ต้องติดต่อกับเทลเน็ตเวิร์กที่รันอยู่บนเครื่องให้บริการนั่นเอง

หลักการของการทำ NVT มีอยู่ว่าเราจะมองเทอร์มินอลที่รันแอปพลิเคชันทั้ง 2 ที่ต้องการติดต่อกันนั้นเป็น 2 เทอร์มินอลคือ physical Terminal กับ Logical Terminal

Logical Terminal คือเทอร์มินอลในความคิดโดยเราจะกำหนดมาตรฐานต่างๆ ในการติดต่อให้เหมือนกันทั้ง 2 เครื่อง และเมื่อ logical Terminal บนเครื่องทั้ง 2 เหมือนกันแล้วจึงทำให้ logical Terminal ของแต่ละเครื่องก็จะ MAP Data ที่ติดต่อกันนั้น ให้อยู่ในรูปแบบที่ Physical Terminal ของเครื่องตนเองเข้าใจ ดังนั้นแอปพลิเคชันที่รันอยู่บน Physical Terminal ที่ต่างกัน จึงสามารถติดต่อและแลกเปลี่ยนข้อมูลกันได้โดยไม่มีปัญหาเรื่องสภาพแวดล้อมของแต่ละเครื่องเข้ามาเกี่ยวข้องเลย

และด้วยหลักการที่กล่าวทั้งหมดนี้ในทางปฏิบัติแล้วเราจะแบ่ง Application Layer ออกเป็น 2 Sublayer ดังรูปที่ 3.4



รูปที่ 3.4 แสดงโครงสร้าง NVT

Command Terminal Sublayer เป็น Sublayer ที่รันแอปพลิเคชันนั้นๆอยู่จริงโดยที่ Sublayer นี้ จะมีสิ่งแวดล้อมต่างๆ ตามเครื่องที่รันอยู่ ซึ่ง Sublayer นี้ ก็เปรียบเสมือน Physical Terminal นั่นเอง

Terminal Communication เป็น Sublayer ที่แปลงสิ่งแวดล้อมที่ไม่เหมือนกันทั้ง 2 เครื่อง ให้เป็นมาตรฐานเดียวกัน เพื่อให้แอปพลิเคชันทั้ง 2 สามารถคุยกันได้และแปลงข้อมูลที่ติดต่อกันให้อยู่ในรูปแบบที่ Physical Terminal ของตนเองเข้าใจ ซึ่ง Sublayer นี้ก็เปรียบเสมือน Logical Terminal นั่นเอง

3.2.2 การทำงานและ Option ของ NVT ในเทลเนท

การทำงานจะเริ่มจากเมื่อผู้ใช้งานเทลเนท ที่เทอร์มินอลของตนเองที่สามารถติดต่อเข้าไปใน NVT ของเครื่องโฮสต์ที่ต้องการจะติดต่อดูได้อย่างที่ได้กล่าวมาแล้ว NVT ที่ติดต่อดูด้วยนั้น จะทำให้ user เหมือนกันได้ใช้เทอร์มินอลบนเครื่องโฮสต์นั้นจริงๆ แต่ติไวซ์ที่ผู้ใช้เห็นบนเครื่องโฮสต์นั้น จริงๆ แล้วเป็นเพียงการจำลองมาเท่านั้น

เมื่อให้เห็นภาพได้ชัดเจนยิ่งขึ้น คุณลองคิดถึงเทอร์มินอลติไวซ์จริงๆ ดูว่าอะไรเป็น ดีไวซ์ที่เป็นอินพุทและ อะไรเป็นดีไวซ์ที่เป็นเอาท์พุท (เช่น จอภาพ, เครื่องพิมพ์) ดังนั้น NVT ก็จะจำลองเทอร์มินอลติไวซ์จริงๆ เหล่านั้น ซึ่งก็มีคีย์บอร์ดเป็นอินพุทและ printer เป็น เอาท์พุทจะพิมพ์ข้อความที่ได้รับมาจากโฮสต์บางข้อความที่โฮสต์ต้อง แสดงและคีย์บอร์ดก็จะสร้างข้อความที่จะต้องส่งออกไปยังโฮสต์เสมือนกับว่าเทอร์มินอลที่ใช้อยู่เป็นเทอร์มินอลจริงๆ บนเครื่องโฮสต์นั่นเอง

และเนื่องจากในแต่ละโฮสต์มักจะมียูสเซอร์จำนวนมากดังนั้นการจะทำให้ NVT ตอบสนองกับสิ่งแวดล้อมของเทอร์มินอลของแต่ละยูสเซอร์จึงทำได้ยาก ซึ่งวิธีแก้ก็คือ ใช้วิธีการตกลง Option

ต่างๆ ที่แต่ละเทอร์มินอลแต่ละ User ต้องการ โดยจะทำการตั้งคำสั่งกันในตอนที่ทำคอนเนคชันกัน option ที่ NVT เตรียมตัวไว้มีตัวอย่างดังนี้

- Option เปลี่ยนลักษณะตัวอักษรของ NVT
- Option ที่เลือกว่าจะให้ NVT สะท้อน ตัวอักษรกลับมาให้ผู้ใช้เห็นบนจอภาพ

หรือไม่

- Option เลือก mode ว่าจะส่ง Data เป็นบรรทัดเดียว หรือจะเลือก Mode ส่ง Data

ทีละตัวอักษร

และยังมี Option อื่นๆ อีก การขอกระทำ Option ต่างๆ เหล่านี้ สามารถเลือกว่าจะใส่เพิ่มหรือถอนออกได้ ซึ่งการตกลงเจรจาที่จะทำ Option นั้น ทำได้โดยใช้ลำดับของข้อความ WILL, WON'T DO และ DON'T นั่นเอง

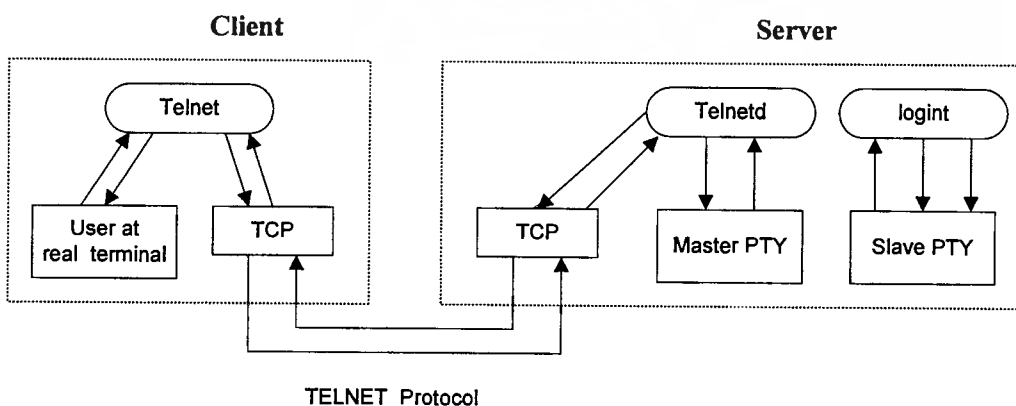
“WILL X” เป็นข้อความที่ส่งจากไชด์ใดไชด์หนึ่งโดยไชด์ที่ส่งนั้นได้บอกไชด์อื่นที่ติดต่อดัวยว่า ไชด์ตนเองจะทำ option x ซึ่งถ้าไชด์อื่นยอมรับการทำ Option X นั้นได้ ก็จะตอบกลับมาด้วย “DO X” แต่ถ้าไชด์อีกไชด์ไม่ยอมรับการทำ Option X นั้น ไชด์นั้นๆ ก็จะตอบกลับไ้ว่า “WON'T X” นั่นเอง ซึ่งตัวอย่างของ Telnet Option ได้แสดงเอาไว้ในตารางแล้ว

3.3 Telnet Model

ในหัวข้อนี้เราจะพูดถึงโครงสร้างและ Model ของเทลเน็ตว่ามีส่วนประกอบอะไรบ้าง และแต่ละส่วนทำหน้าที่อะไรบ้าง

3.3.1 โครงสร้างของ Telnet Model และการทำงาน

บนระบบปฏิบัติการยูนิกซ์ได้มีการออกแบบโครงสร้างของโปรแกรมเทลเน็ตโดยประกอบด้วย Telnet Command ซึ่งเป็นโปรแกรมที่ทำงานอยู่บนเครื่องให้บริการและ Telnet d ซึ่งเป็น process ที่ RUN อยู่บนเครื่องให้บริการซึ่งมีระบบปฏิบัติการเป็นระบบยูนิกซ์ ซึ่ง Telnet ของระบบปฏิบัติการยูนิกซ์ มีโครงสร้างดังรูปที่ 3.5



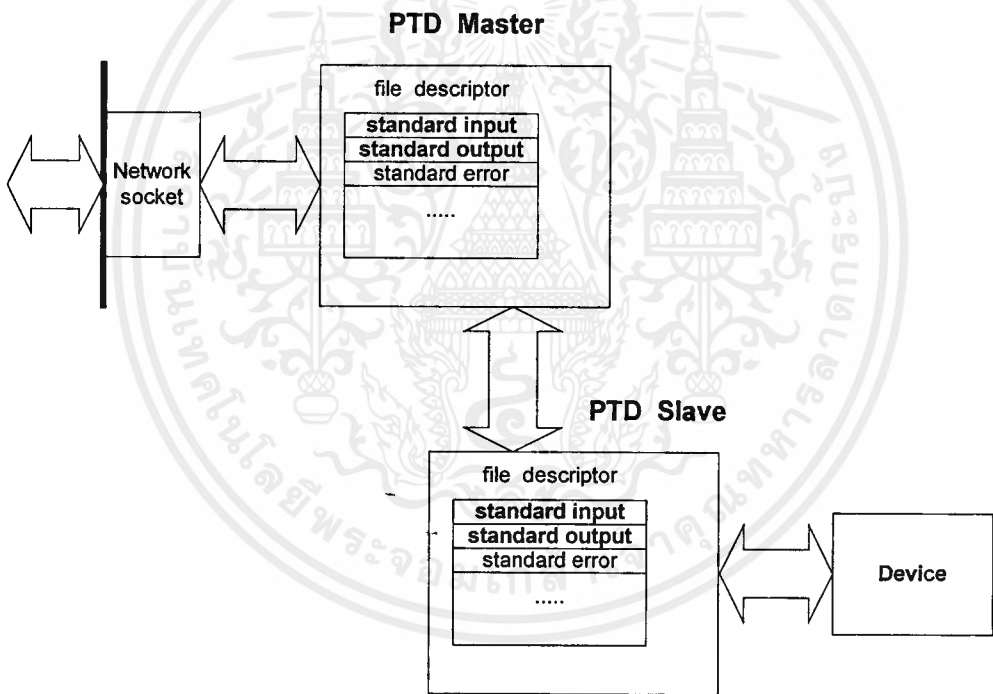
รูปที่ 3.5 แสดงการทำงานของเทลเน็ตโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปการทำงานจะเริ่มขึ้นเมื่อผู้ใช้ใส่คำสั่ง open และตามด้วยชื่อ Host name ที่ผู้ใช้ต้องการจะติดต่อด้วยให้โปรแกรม Telnet Command บนเครื่องให้บริการ โปรแกรม Telnet Command ก็จะสร้าง Connection ด้วยโปรโตคอล TCP/IP ติดต่อกับเครื่องให้บริการที่ผู้ใช้ต้องการติดต่อกับ หลังจากการสร้าง Connection กับเครื่องให้บริการสำเร็จแล้วโปรแกรม Telnet Command และ Telnetd Process ก็จะทำการตกลง Option ที่จะกระทำต่อกัน เช่น ชนิดของเทอร์มินอลบนเครื่องให้บริการและเครื่องให้บริการ , Telnet Mode ว่าเป็นการส่งที่ละบรรทัดหรือส่งข้อมูลที่จะเป็นตัวอักษร เป็นต้น และเมื่อ Telnet Command ตกลง Option กับ Telnet Process เสร็จเรียบร้อยแล้ว Telnetd Process บนเครื่องให้บริการก็จะสร้าง Pseudo-Terminal (PTD) ซึ่งเป็น device driver process ขึ้นมาสอง PTD คือ PTD Master และ PTD Slave

PTD Slave ก็คือ device ของเทอร์มินอลจริงๆ บนเครื่องที่ Telnet process รันอยู่ หรือก็คือ Physical Terminal ใน NVT นั่นเอง

ส่วน PTD Master ก็คือ process ที่แทน device บนเครื่องที่ผู้ใช้ใช้ชื่อนั้นเอง หรือก็คือ Logical Terminal ใน NVT นั่นเอง



รูปที่ 3.6 ขั้นตอนการ Connect

ซึ่งถ้าเรามาลองพิจารณา Telnet Model ดูแล้วเราก็จะพบว่า Telnet Model นั้นได้ออกแบบไว้ตามหลักการของ NVT ทุกประการ และจากหลักการดังกล่าวเวลาที่ Telnetd process จะเขียนข้อมูลลง Device แทนที่ Telnet process จะเขียนลง Device จริงๆ ของเทอร์มินอลที่ Telnet process รันอยู่เองก็เขียนลง PTV Master แทน Device ที่ใช้หลักการนี้ ก็ได้แก่ Device จอภาพ คือแทนที่จะเขียนจอภาพของเครื่องให้บริการ เราเขียนไปที่จอภาพของเทอร์มินอลของผู้ใช้แทน และนั่นทำนองเดียวกันในเวลาที่ใช้ต้องการเขียนข้อมูลบาง

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์หรือการเชิงพาณิชย์เพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลลง PTD Master PTD Master ก็จะนำข้อมูลนั้นเขียนลง PTD Slave อีกทีหนึ่ง ตัวอย่างเช่น Device คีย์บอร์ด เมื่อผู้ใช้กดคีย์บอร์ดTelnet Command ก็จะส่ง Data เขียนลงบน PTD Master แล้ว DTP Master ก็ จะเขียนข้อมูลนั้นลง PTD Slave ซึ่งก็คือคีย์บอร์ดBuffer ของเครื่องเซิร์ฟเวอร์นั่นเองซึ่งก็เสมือนกับผู้ใช้ กำลังกดคีย์บอร์ดอยู่บนเครื่องให้บริการนั้นๆ จริงๆ

เรากลับมาดูการทำงานของ Telnet d กันต่อ Telnet d จะสร้าง PTD มาเพียง process เดียว ซึ่ง process PTD ตัวแรกที่สร้างนี้ก็คือ PTD Master นั่นเอง ซึ่ง PTD process นี้จะ Fork process ลูกอีก process นั่นก็คือ PTD Slave ซึ่ง code ของ process PTD ทั้ง 2 จะเป็นเดียวกันแต่มีการทำงานที่ต่างกัน โดย ดูจาก ID process ว่า process ไหนเป็น process พ่อ (PTD Master) หรือ process ลูก (PTD Slave) โดย process พ่อ (PTD Master) จะ set file descriptors ของ Standard input Standard output และ Standard error ไปที่ซ็อกเก็ตที่ถูกสร้างขึ้นตอนทำ connection ในตอนเริ่มต้นนั่นเอง ส่วน process ลูก (PTD Slave) ก็จะ set file descriptors ของ Standard input Standard output และ Standard error ไปที่ device ต่างๆ บนเครื่อง ให้บริการดังรูปที่ 3.6 Process ลูก (PTD Slave) หลังจาก set file descriptors ต่างๆ เสร็จก็จะทำการ exec login command แล้วส่ง data จาก login command ไปให้ process พ่อ (PTD Master) อ่านเพื่อจะได้เขียนลง Standard output ของ PTD Master ซึ่งเป็นซ็อกเก็ตที่ต่อกับเครื่องให้บริการอีกทีหนึ่งที่ connect อยู่บน เครื่องข่าย แล้วที่เครื่องให้บริการของผู้ใช้ก็จะแสดง login prompt (login:) เพื่อให้ผู้ใช้ใส่ข้อมูล และเมื่อผู้ใช้ ใส่ข้อมูล Telnet command ก็จะส่งข้อมูลที่ผู้ใช้ใส่มาขึ้นไปที่ Telnet d process แล้ว PTD Master ก็จะ อ่านข้อมูลนี้ แล้วส่งไปให้ PTD Slave เพื่อเขียนลง device เพื่อให้ login command นำไปใช้ต่อไปซึ่งขั้นตอนการ execs login command นี้จะเป็นขั้นตอนแรกก่อนที่ Telnetd จะ start shell เพื่อให้ผู้ใช้เข้าใช้ระบบ ของเครื่องให้บริการต่อไป กลับมาดูที่ฝั่งเครื่องให้บริการกันบ้างTelnet command เมื่อทำ connection เสร็จ แล้ว ก็จะให้ผู้ใส่ ใส่ข้อมูลของคุณหรือที่คุณจะใส่ไปยังเครื่องให้บริการ และแสดงผลข้อมูลจากเครื่องให้ บริการ ที่ส่งกลับมายังเทอร์มินอลของคุณด้วย และด้วยรูปแบบการทำงานแบบนี้จึงทำให้ผู้ใช้สามารถอ่าน ข้อมูลจาก shell บนเครื่องให้บริการ และผู้ใช้สามารถส่งข้อมูลไปเขียนลง shell แสดงผลการเขียนข้อมูล บนระบบของผู้ใช้เองได้

3.3.2 Mode การส่งข้อมูลของ Telnet

หลังจากที่ทำ Connection ระหว่าง Telnet Command บนเครื่องให้บริการกับ Telnetd process บนเครื่องให้บริการแล้ว ก็จะมีการตกลง Option กันแล้วก็จะมีการส่งข้อมูลกันซึ่ง Mode การส่ง ข้อมูลจาก Telnet command ไปยัง Telnetd process จะมีอยู่ 2 Mode คือ Mode line-at-a-time และ Mode character-at-a-time

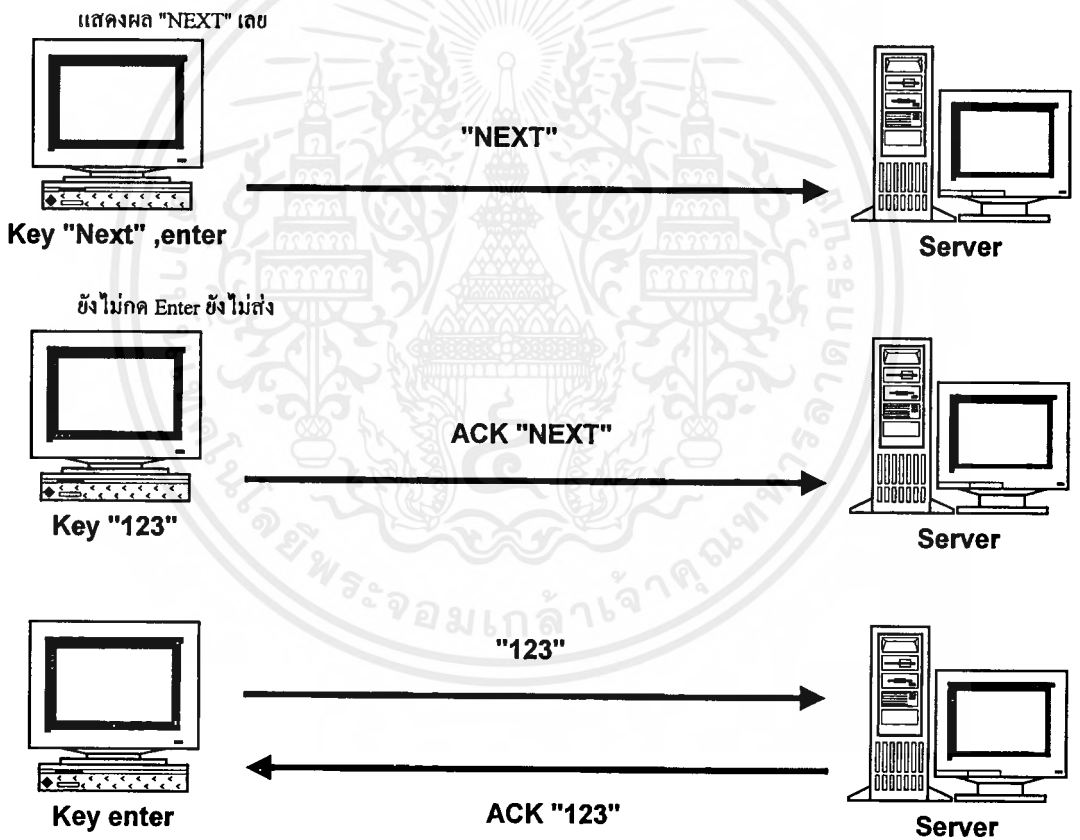
3.3.2.1 Mode การส่งข้อมูลแบบ line-at-a-time

การส่งข้อมูลใน Mode นี้ จะส่งข้อมูลที่ละบรรทัด คือ ส่งเมื่อผู้ใช้มีการกด Key 'New line' และเมื่อส่งข้อมูลจากเครื่องให้บริการไปที่เครื่องให้บริการแล้ว เครื่องให้บริการก็จะตอบรับ (ACK)

ข้อมูลที่ส่งมานั้นเท่านั้น จะไม่มีการส่ง Echo ข้อความเพื่อแสดงผลบนเครื่องให้บริการแต่อย่างใด เพราะ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปเผยแพร่โดยไม่ผ่านการพิจารณาใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลบนเครื่องใช้บริการจะถูกควบคุมด้วยตัวของเครื่องใช้บริการ เนื่องจากไม่สามารถรอการส่ง Echo ข้อความกลับมาจากเครื่องให้บริการไม่ได้ เพราะเครื่องใช้บริการเองต้องรอการกด Key "New line" จึงจะส่งข้อมูล ดังนั้นถ้ารอ Echo จากเครื่องให้บริการแล้วผู้ใช้จะไม่เห็นตัวอักษรที่ตนพิมพ์เข้าไปจนกว่าผู้ใช้จะกด Key "New line" ซึ่งถ้าเป็นอย่างนี้และจะเป็นการไม่สะดวกที่ผู้ใช้จะไม่เห็นตัวอักษรที่ตนพิมพ์เข้าไปจนกว่าผู้ใช้จะกด Key "New line" ซึ่งถ้าเป็นอย่างนี้แล้วจะเป็นการไม่สะดวกที่ผู้ใช้จะแก้ข้อความที่พิมพ์ผิด เพราะผู้ใช้อย่างนี้ไม่เห็นข้อความที่พิมพ์เข้าไป

ดังนั้นเพื่อไม่ให้เกิดเหตุการณ์ดังกล่าว การส่งข้อมูลใน Mode นี้ จึงให้เครื่องใช้บริการเป็นเครื่องที่ควบคุมการแสดงผลเอง ดังนั้นการส่งข้อมูลใน Mode นี้จึงไม่มีปัญหาในการใช้ Key พิเศษ เช่น Key Backspace หรือ Key Delete เพราะอย่างที่ได้อธิบายไปแล้ว การจัดการแสดงผลจะทำโดยเทอร์มินอลของผู้ใช้ และผู้ใช้อย่างนี้เป็นผู้กำหนดว่าจะส่งข้อมูลไปให้เครื่องให้บริการเมื่อไหร่เองอีกด้วย ซึ่งการส่งข้อมูลใน Mode นี้แสดงไว้ดังรูป

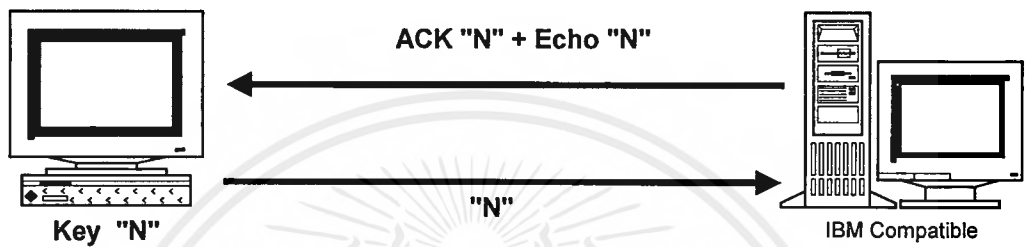


รูปที่ 3.7 การส่งข้อมูลแบบ line-at-a-time

แต่การส่งข้อมูลแบบ line-at-a-time ก็มีปัญหาคือ การส่งข้อมูลใน Mode นี้ เมื่อส่งข้อมูลไปแล้วเราจะไม่สามารถกลับไปแก้ไขข้อมูลเดิมได้อีก จึงทำให้ไม่สามารถใช้โปรแกรมประเภท Word หรือ Editor ได้ใน Telnet Mode นี้ เพราะการส่งข้อมูลแบบนี้ไม่สามารถแก้ไขข้อมูลที่ส่งไปแล้วได้นั่นเอง

3.3.2.2 Mode การส่งข้อมูลแบบ Character-at-a-time

การส่งข้อมูลใน Mode นี้ จะส่งข้อมูลที่ละตัวอักษร คือจะส่งกันทันทีที่ผู้ใช้มีการกดคีย์ใดๆ และจะส่งข้อมูลที่ละตัวอักษร ซึ่งการส่งข้อมูลใน Mode นี้ Telnet command ที่รันอยู่บนเครื่องให้บริการจะยังไม่แสดงผลข้อมูลที่ส่งไปยังเครื่องให้บริการบนเครื่องให้บริการบนจอภาพทันทีแต่จะรอให้เครื่องให้บริการตอบรับและส่ง Echo กลับมาให้จึงจะแสดงผลบนจอภาพ การส่งข้อมูลใน Mode นี้แสดงไว้ดังรูป



รูปที่ 3.8 การส่งข้อมูลแบบ character-at-a-time

ข้อดีของการส่งข้อมูลใน Mode นี้ ก็คือ ผู้ใช้สามารถแก้ไขข้อมูลที่ส่งไปแล้วได้ง่ายกว่าส่งข้อมูลใน Mode line-at-a-time เพราะข้อมูลที่ส่งไปแล้วนั้นมีขนาดเพียง 1 ตัวอักษร ดังนั้นเราจึงส่งข้อมูลไปแก้ไขข้อมูลที่ส่งไปแล้วเหล่านั้นได้ แล้วด้วยเหตุนี้จึงทำให้การส่งข้อมูลใน Mode นี้สามารถใช้งานโปรแกรมจำพวก Word หรือ Editor ได้จึงทำให้การส่งข้อมูลใน Mode นี้เป็นที่นิยมใช้กันในปัจจุบัน

แต่การส่งข้อมูลใน Mode character-at-a-time นี้ก็มีปัญหาคือ การแสดงผลบนจอภาพของการส่งข้อมูลใน Mode นี้ นั้นจะต้องแสดงจาก Echo character ที่ส่งกลับมาจากโฮสต์ดังนั้นการแสดงผลบางอย่างจึงไม่สามารถทำได้ เช่น เมื่อผู้ใช้กด Key backspace แล้วโฮสต์ส่ง ASCII ของ backspace กลับมาให้เครื่องให้บริการของผู้ใช้ แล้วเครื่องให้บริการของผู้ใช้ จะแสดงผลของ Key backspace ออกมาเป็น character ของ ASCII ของ backspace ซึ่งไม่ได้เป็นไปตามที่ผู้ใช้ ต้องการซึ่งสิ่งที่ผู้ใช้ ต้องการคือ ให้ Move Cursor กลับ ไปหนึ่งตัวอักษรและลบ Character ตัวนั้นด้วย

ดังนั้นจึงต้องมีการทำเทอร์มินอลเสมือนที่จะกล่าวถึงในหัวข้อต่อไป

3.4 เทอร์มินอลเสมือน

จากที่เคยกล่าวมาแล้วนั้นได้พูดถึงการทำเทอร์มินอลเสมือน ไว้ถึงเหตุผลในการทำเทอร์มินอลเสมือน ก็เพราะการส่งข้อมูลใน Mode character-at-a-time การแสดงผลบนจอภาพของเครื่องให้บริการของผู้ใช้จะถูกควบคุมโดยเครื่องให้บริการ แต่เนื่องจากเครื่องให้บริการแต่ละเครื่องให้บริการอยู่ห่างไกลกันการติดต่อก็ทำได้โดยผ่านเน็ตเวิร์กที่เชื่อมโยงเครื่องทั้งสอง ซึ่งการส่งข้อมูลผ่าน Network นี้เอง ทำให้เครื่องให้บริการไม่สามารถควบคุมการแสดงผลของเครื่องให้บริการด้วย Echo character เพียงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างเฉียวได้ ดังนั้นจึงต้องมีการทำเทอร์มินอลเสมือน เช่น เมื่อผู้ใช้กด Key backspace แทนที่เครื่องให้บริการ Echo character backspace กลับมาก็ส่งคำสั่งให้เครื่องให้บริการทำการย้ายตำแหน่ง Curser ถอยหลังกลับ 1 ตัวอักษรแทน เป็นต้น ซึ่งคำสั่งที่เครื่องให้บริการใช้ทำเทอร์มินอลเสมือนเหล่านี้ จะมีลักษณะเป็น Code คำสั่ง ต่างๆ โดยจะเริ่มคัมด้วย “escape” character ซึ่งจะตกลงกันเอาไว้ระหว่างเครื่องให้บริการและเครื่องใช้บริการตั้งแต่ตอนที่ติดต่อกันตอนแรก (ตอนที่ตกลง Option ต่างๆ นั่นเอง) ซึ่ง Code ที่ใช้ในการทำเทอร์มินอลเสมือนเหล่านี้ ได้มีการกำหนดไว้เป็นมาตรฐานต่างๆ ไว้มากมายมาตรฐาน แต่ที่นิยมใช้กันมากที่สุดได้แก่ มาตรฐานตระกูล VT ซึ่งในภาคผนวกได้มีตัวอย่าง Code ของ VT100 ไว้ให้แล้ว

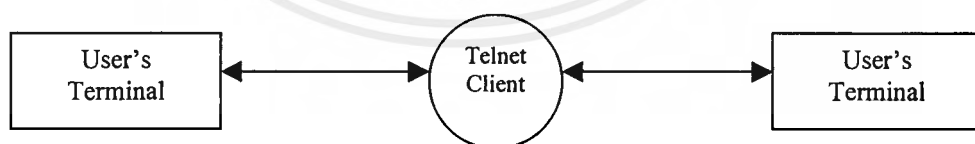
3.5 คำสั่งและการควบคุมของเทลเน็ต (Command and Control Information)

ในการรวมเอาข้อมูลที่เป็นอักขระเทลเน็ตจะยินยอมให้เครื่องใช้บริการ (Client) และเครื่องให้บริการ (Server) ทำการแลกเปลี่ยน คำสั่งหรือ ข้อมูลการควบคุม เพราะในระหว่างการติดต่อสื่อสารกันเครื่องใช้บริการและเครื่องให้บริการ ผ่านการเชื่อมต่อ TCP โพรโตคอลจะจัดการแปลคำสั่งหรือข้อมูลการควบคุมเป็นรหัส ดังนั้นทางฝ่ายผู้รับจะต้องรู้จักการแปลรหัสนั้นเป็นเหตุให้ต้องทราบจุดมุ่งหมายของโปรโตคอลจึงมีการกำหนดการส่งรหัสของคำสั่งและการรู้จักการรับข้อมูลอย่างไร

เทลเน็ตถูกกำหนดอย่างเด่นชัดระหว่างการติดต่อสื่อสารของผู้ใช้ซึ่งเป็นเทอร์มินอลและการใช้บริการจากระยะไกล โปรโตคอลถูกกำหนดความสามารถพิเศษของเทอร์มินอลในส่วนของ ติยบอร์คซึ่งผู้ใช้สามารถส่งอักษรและแสดงผลบนหน้าจอในโหมดข้อความ

ในทางปฏิบัติแล้ว ผู้ใช้สามารถตัดสินใจที่จะร้องขอความต้องการจากเครื่องใช้บริการกับทุกๆ อินพุท ในแต่ละสถานที่ติยบอร์คและเอาท์พุท ซึ่งอาจจะเป็นแสดงผล นอกจากนั้นผู้ใช้ยังเรียกร้องจากเครื่องใช้บริการ ซึ่งอยู่ในรูปของวินโดวส์

3.6 เทลเน็ตไกลเอนท์อัลกอริทึม



รูปที่ 3.9 เทลเน็ตไกลเอนท์อัลกอริทึม

รูปที่ 3.9 แนวความคิดของเทลเน็ตไกลเอนท์ผู้ขอบริการต้องการส่งตัวอักขระจากผู้ใช้โดยติยบอร์คไปยังผู้ให้บริการผ่านทางไกลและส่งข้อมูลจากผู้ให้บริการทางไกลไปยังผู้ขอใช้บริการ

3.7 อัลกอริทึมของเทลเน็ตไคลเอนท์

1. วิเคราะห์ค่าของข้อมูลที่ส่งมาและเริ่มต้นส่งโครงสร้างข้อมูล
2. เปิดการคอนเนคชันโดย TCP ไปยังพอร์ต (port) ของผู้ให้บริการระยะไกล
3. รอการรับอินพุตจาก ผู้ใช้ (user) หรือข้อมูลจะอยู่สูงถึง Connection TCP
4. ถ้าข้อมูลที่ถูกส่งจากคีย์บอร์ดและอ่านค่านำส่ง ไปผ่านกระบวนการเปลี่ยนรูปแบบ ไปยัง Network Virtual Terminal (NVT)
5. กลับไปทำข้อ 3

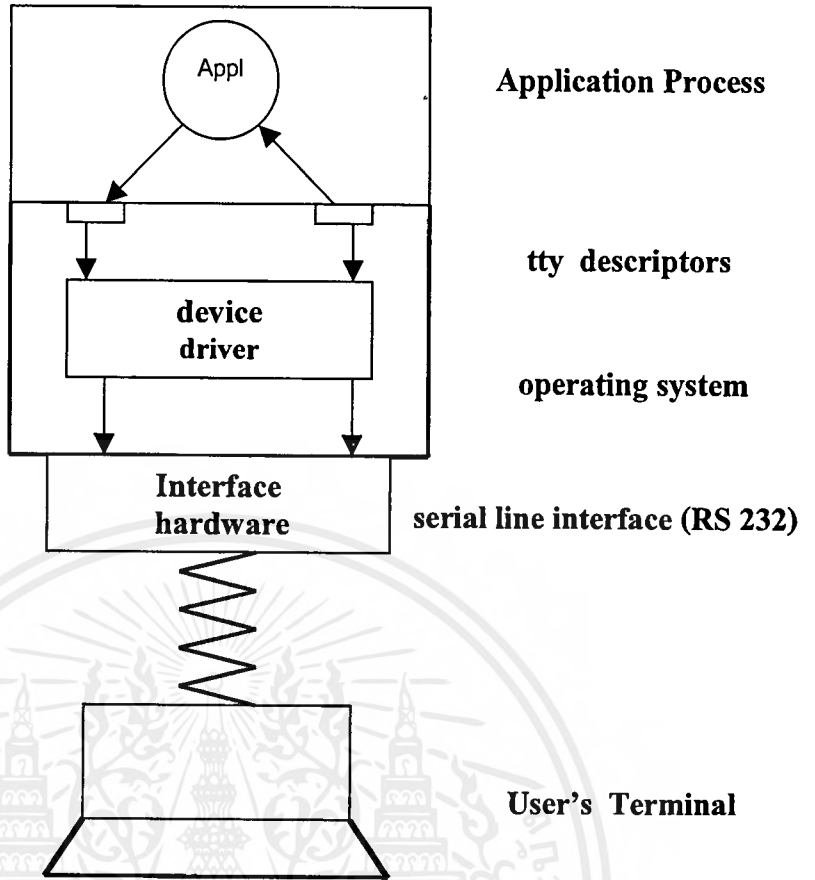
3.8 เทอร์มินอล I/O ในยูนิกซ์ (Terminal I/O ในยูนิกซ์)

แนวคิดจากเรื่องที่ผ่านมาเป็นตัวอย่างให้เห็นภาพมากที่สุด แม้กระทั่งเนื้อหาของโปรโตคอลเทลเน็ต (Telnet Protocol) และ เทอร์มินอล I/O ที่ยุ่งยากของรหัสไปจนถึงเข้าใจซอฟต์แวร์เทลเน็ตไคลเอนท์เป็นอย่างไร สลับกับจุดที่ผู้ใช้คือเทอร์มินอลขั้นแรกต้องเข้าใจในเนื้อหาของระบบปฏิบัติการที่จัดการกับเทอร์มินอล I/O ไปยังอุปกรณ์เทอร์มินอลตามเปิด-อ่าน-เขียน-ปิด (open-read-write-close) ซึ่งเป็นไคอะแกรมที่ใช้สำหรับ I/O ไปยังระบบไฟล์หรือซ็อกเก็ต โปรแกรมประยุกต์จะเรียกเปิด (open) ที่ได้จาก I/O สำหรับคีย์บอร์ดหรือแสดงผล อ่าน (read) รับข้อมูลจากผู้ใช้โดยคีย์บอร์ดและเขียน (write) ส่งข้อมูลไปแสดงผลให้แก่ผู้ใช้

ในทางปฏิบัติแล้วการประยุกต์ไม่ได้เรียกเปิดไปสร้างสำหรับผู้ใช้บริการ เพราะว่าตัวแปลคำสั่งได้เตรียมการเปิดโดยอัตโนมัติ

แม้ว่าเทอร์มินอล I/O มีลักษณะเช่นเดียวกับพื้นฐานของไฟล์ I/O และสิ่งมากมายของเนื้อหาจะมีความยุ่งยาก บางอย่างของเนื้อหาเกิดขึ้น เนื่องจากได้เริ่มต้นฮาร์ดแวร์ก่อน สำหรับในตัวอย่าง แม้ว่าผู้ใช้บริการคิดถึงในส่วนของคีย์บอร์ดและการแสดง ฮาร์ดแวร์จะแยกให้คีย์บอร์ดเป็นอินพุตการแสดงผลเป็นเอาท์พุตและแยกออกเป็น 2 อุปกรณ์ ดังนั้นฮาร์ดแวร์จะไม่ทำอัตโนมัติในการแสดงผลของแต่ละชนิด ของผู้ใช้บริการ

การประยุกต์ผู้ใช้ต้องการเห็นการเปลี่ยนแปลงอักษรในการแสดงผลทุกประเภทแต่การประยุกต์บางอย่างก็ต้องการที่จะไม่แสดงของอินพุตจากคีย์บอร์ดเช่น ในการใส่รหัสผ่านเพื่อจะต้องการรักษา รหัสผ่าน จากการเฝ้ามองของผู้ที่เฝ้ามองอยู่เป็นต้น ดังนั้นในการใช้งานทุกอย่างมีรายการข้อปลีกย่อยที่มีความเกี่ยวข้องกับเทอร์มินอล I/O และการแสดงอักษร ในระบบยูนิกซ์ได้เตรียมโปรแกรมเหล่านี้ไว้อัตโนมัติแล้ว เรียกว่า เทอร์มินอลดีไวซ์ไควร์เวอร์จะรวมอยู่ในระบบปฏิบัติการ ดังรูป



รูปที่ 3.10 เทอร์มินอลดีไวส์ไดรเวอร์

ดีไวส์ไดรเวอร์ (device driver) มีความเกี่ยวข้องกับอินพุทที่เป็น Keyboard และตรงกับการแสดงผลของเทอร์มินอลดีไวส์ไดรเวอร์สามารถโต้ตอบกันได้ หรือหยุดการโต้ตอบได้ และดีไวส์ไดรเวอร์สามารถรองรับสัญลักษณ์พิเศษ ซึ่งใช้ในการจัดจังหวะหรือหยุดการทำงานในขณะนั้น

บทที่ 4

ซีจีไอ จาวาเบื้องต้นและเรียลไทม์

4.1 ซีจีไอ (CGI : Common Gateway Interface)

4.1.1 หลักการทั่วไปของ CGI

Common Gateway Interface (CGI) เป็นมาตรฐานการเชื่อมต่อระหว่างโปรแกรมภายนอกกับศูนย์บริการข้อมูล (information server) เช่น ศูนย์บริการบนเครือข่าย (Web server) เอกสารรูปแบบ HTML นั้นเว็บเซิร์ฟเวอร์จะรับมาเหมือนกันทุกครั้ง (เพิ่มข้อความจะไม่เปลี่ยนแปลง) ทุกครั้งที่คุณเรียกโปรแกรม CGI ในขณะนั้น มันจะทำงานและส่งข้อมูลออกมาในลักษณะแตกต่างกันออกไป

การทำงานคร่าวๆ ไคลเอนต์เว็บเบราว์เซอร์ติดต่อสื่อสารกับเว็บเซิร์ฟเวอร์โดยใช้โปรโตคอล HTTP เมื่อไคลเอนต์เว็บเบราว์เซอร์ร้องขอที่อยู่ของเอกสารหรือที่เรียกว่า ยูอาร์แอล (URL) เว็บเซิร์ฟเวอร์ก็จะเริ่มติดต่อกับโปรแกรม CGI โดยรับ standard input ของเว็บเซิร์ฟเวอร์ให้กับ standard output ของโปรแกรม CGI และโปรแกรม CGI ก็จะให้บริการและติดต่อสื่อสารกับฐานข้อมูล (Data Base Engine) เมื่อฐานข้อมูลรับการร้องขอก็จะส่งข้อมูลคืนกลับไปให้โปรแกรม CGI ต่อมาโปรแกรม CGI จะเปลี่ยนรูปแบบไปแสดงผลบนเว็บเพจโดยเปลี่ยนเป็นภาษา HTML และคืนผลไปให้เว็บเซิร์ฟเวอร์ผ่าน standard output สุดท้ายเว็บเซิร์ฟเวอร์จะคืนผลเป็นภาษา HTML ไปยังไคลเอนต์เว็บเบราว์เซอร์ของผู้ใช้

4.1.1.2 ศัพท์เฉพาะของ CGI

Environment variable คือชื่อของพารามิเตอร์ที่ส่งผ่านข้อมูลจากเซิร์ฟเวอร์ไปยังไคลเอนต์

Script คือโปรแกรมที่ถูกเรียกโดยเซิร์ฟเวอร์ซึ่งในที่นี้หมายถึง โปรแกรมของ CGI นั้นเอง

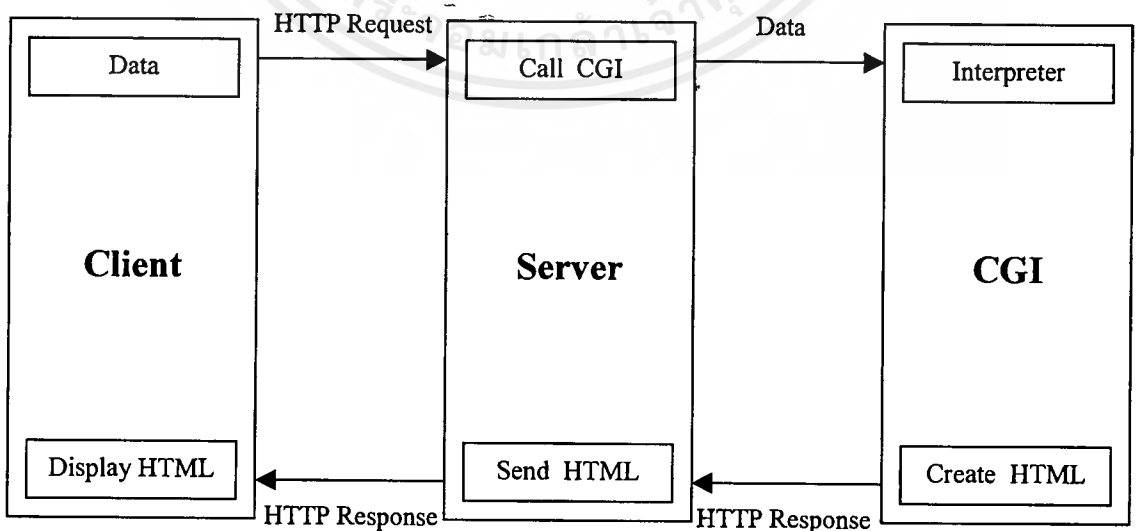
Server คือ application program ซึ่งเรียกใช้งานสคริปต์ เพื่อให้บริการตามที่ร้องขอ โปรแกรมนี้จะทำงานเป็นอิสระบนเครื่อง ในยูนิกซ์โปรแกรมนี้คือตัว daemon

4.1.2 หน้าที่ของ CGI สคริปต์ (CGI Script)

ตามปกติแล้วเว็บเบราว์เซอร์ได้ค่า URL หรือที่อยู่ของเอกสารจากการกดเลือกการที่ให้เลือก หรือใส่ที่อยู่ในรายการเมนู เครื่องก็จะทำการติดต่อไปยัง HTTP เซิร์ฟเวอร์ตาม URL ที่ระบุเอาไว้ นั้น เมื่อเครื่องเซิร์ฟเวอร์ได้รับการขอเพิ่มข้อมูลเข้ามาก็จะทำการส่งเพิ่มข้อมูลที่ต้องการกลับมาให้ แต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเครื่องบางเครื่องที่ไม่ต้องการให้ส่งเพิ่มข้อมูลจากบาง directory ให้กับผู้ใช้บางรายที่ส่งรายการขอเข้ามาก็สามารถทำได้โดยการเรียกโปรแกรมบางโปรแกรมเพื่อใช้ตรวจสอบผู้ใช้ โปรแกรมที่ทำหน้าที่เช่นนี้เรียกว่า Common Gateway Interface หรือ CGI ซึ่งสังเกตได้จากไครเรททอรีที่ใช้เก็บโปรแกรมนี้อาจจะเป็นไครเรททอรี ที่ชื่อ /cgi-bin/ แต่หน้าที่ของ CGI ไม่ได้จำกัดอยู่แค่นี้ ถ้าสังเกตให้ดีเวลาเข้าไปในเวบเพจที่มีการนับจำนวนคนที่เข้ามาเยี่ยมชมคืออาศัย CGI สคริปต์ในการนับ ในเครื่อง Workstation บางรุ่นที่ออกมาใหม่ก็มีการใช้งานของ CGI สำหรับผู้ควบคุมระบบ ด้วยเพราะโดยทั่วไปแล้วต้องใช้งานแบบ Command Line ผู้ที่ใช้งานจะต้องทราบขั้นตอนและคำสั่งต่าง ๆ ของระบบเป็นอย่างดี แต่โดยอาศัย CGI และการใช้งานบริการแบบเวลด์ไวด์เวบก็สามารถช่วยแบ่งเบาภาระส่วนนี้ได้ ส่วนที่ติดต่อกับผู้ใช้ของเวบ จะทำหน้าที่เพียงรับรายการจากผู้ใช้เท่านั้น แต่จะมีการประมวลผลก็ต่อเมื่อส่งมาถึงเครื่องที่ให้บริการแล้วเท่านั้น จากลักษณะการทำงานแบบไคลเอนท์เซิร์ฟเวอร์นี้เอง ส่วนติดต่อกับผู้ใช้หรือเวบบราวเซอร์จะทำหน้าที่เพียงรับรายการจากผู้ใช้แล้วส่งต่อไปกับเวบเซิร์ฟเวอร์ เมื่อเวบเซิร์ฟเวอร์ได้รับอินพุตคาค่า (Input Data) ซึ่งอยู่ในรูป HTTP Request ก็จะไปทำการเรียกโปรแกรม CGI โดยส่งค่าตัวแปรหรือพารามิเตอร์ผ่านต่อไปกับ CGI Application โปรแกรม CGI ก็จะทำการแปลความหมาย (Parse) รายการที่รับเข้ามา เช่น ถ้าเป็นการขออนุญาตเข้าไปใช้งานใน Directory ที่อนุญาตให้เฉพาะผู้ใช้บางคนเท่านั้นพารามิเตอร์ที่ส่งมาให้ก็จะ เป็นชื่อและรหัสผ่าน เมื่อโปรแกรม CGI รับชื่อและรหัสผ่านเข้ามาแล้วก็จะทำการตรวจสอบว่ารายการ นั้นถูกต้องหรือไม่ ถ้าถูกต้องก็จะทำการส่งเพิ่มข้อมูลที่ต้องการไปให้ หรือถ้าเป็นการสอบถามรายการ ค่าพารามิเตอร์ก็จะเป็น Keyword ของรายการที่ใช้สอบถาม โปรแกรม CGI สำหรับงานนี้ก็จะนำค่าพารามิเตอร์หรือรายการที่สอบถามที่ส่งเข้ามาค้นหารายการในฐานข้อมูล จากนั้นฐานข้อมูลจะส่งรายการกลับมาให้ CGI โปรแกรม CGI ก็จะทำการสร้างเพิ่มข้อมูลในแบบ HTML และใส่ผลลัพธ์ที่ได้จากการสอบถามจากฐานข้อมูลไว้ในเพิ่มข้อมูลนี้แล้วส่งกลับให้กับเวบเซิร์ฟเวอร์เพื่อให้เซิร์ฟเวอร์ส่งกลับมาให้ไคลเอนท์เวบบราวเซอร์แต่รายการที่ส่งกลับมามีการระบุชนิดของเพิ่มข้อมูลนั้นไว้ด้วย



รูปที่ 4.1 แสดงการติดต่อ Client-Server และ CGI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเซิร์ฟเวอร์จะใส่ content-type เพื่อบอกชนิดของแฟ้มข้อมูลนั้นไว้ในส่วน Header field หรือถ้าจะอธิบายสั้น ๆ จะได้ว่า CGI ก็ คือ ส่วนของเวปไซด์ไวด์เว็บเซิร์ฟเวอร์ที่ทำหน้าที่ให้บริการข้อมูลให้กับเวปไซด์ไวด์เว็บไคลเอนต์ดังแสดงในรูปที่ 4.1

4.1.3 ประเภทของภาษาที่สามารถนำมาใช้เป็น CGI

ประเภทของภาษาที่ใช้สำหรับการเขียนโปรแกรมสามารถจำแนกออกได้คือ

Procedural Language เป็นภาษาที่มีการทำงานเป็นขั้นตอนตามวิธีการ (Algorithm)

Object Oriented เป็นภาษาที่มีการทำงานสามารถเรียกใช้เป็นส่วนๆ ได้เหมือนสิ่งของต่างๆ

Logical เป็นภาษาที่ใช้สำหรับหาคำตอบ โดยดูจากสิ่งที่กำหนดให้ (predicate)

Functional เป็นภาษาที่การทำงานของโปรแกรมถูกกำหนดให้อยู่ในรูปของฟังก์ชัน

สำหรับภาษาที่ควรจะนำมาใช้เป็นตัวกลางในการติดต่อบริการเวปไซด์ไวด์เว็บกับโปรแกรมอื่น ๆ ควรเป็นภาษาที่สามารถใช้ประมวลผลข้อความและจัดการกับแฟ้มข้อมูลได้สะดวก ได้แก่

C

C++

Perl

Tcl/Tk

Unix Shell

Visual Basic

FORTRAN

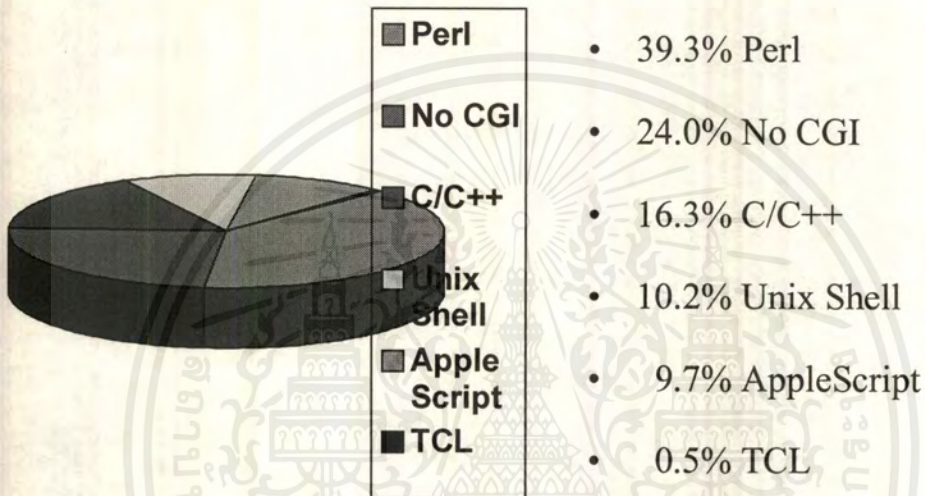
ภาษาลักษณะนี้มีลักษณะเป็น interpreter ที่มีลักษณะเป็น Procedural Language เช่นเดียวกับภาษา C, Pascal และที่สำคัญคือสามารถเรียนรู้ได้ง่าย และสามารถหารายละเอียดเพิ่มเติมได้จากที่ต่าง ๆ ในอินเทอร์เน็ตได้

4.1.3.1 ภาษาสคริปต์ที่นำมาใช้งานเป็น CGI

ภาษา Perl มีชื่อย่อมาจาก Practical Extraction and Report Language เป็นภาษาสคริปต์ภาษาหนึ่งที่เกิดขึ้นโดย Larry Wall เพื่อให้ใช้งานการจัดการกับแฟ้มข้อมูลและข้อความ ภาษานี้เหมาะที่จะใช้ในการอ่านแฟ้มข้อมูลข้อความ (textfiles) โดยทำหน้าที่ในการแยกคำและสร้างรายงาน ก่อนหน้าที่จะมีภาษานี้ในระบบปฏิบัติการแบบยูนิกซ์การใช้งานในลักษณะเดียวกันนี้จะอาศัยเชลล์สคริปต์ แต่ ภาษา Perl จะรวมเอาลักษณะเด่น ๆ ของเชลล์สคริปต์ อย่าง sed, awk และ sh เอาไว้ ภาษา Perl มีลักษณะเป็นภาษา

โครงสร้างคล้ายภาษา C แต่สามารถใช้งานได้สะดวกกว่าภาษาสคริปต์ อย่าง sed, awk และ lex ซึ่งเป็นยูนิกซ์เชลล์สคริปต์ จะเห็นได้ว่าโปรแกรมที่ใช้งาน CGI มักจะเขียนโดยใช้ภาษา Perl เป็นส่วนใหญ่

ในส่วนของ CGI เองนั้นจริง ๆ แล้วสามารถใช้ภาษาอะไรในการเขียนก็ได้ โดยพิจารณาตามความเหมาะสมหรือดูจากการใช้งานเครื่องที่ให้บริการเป็นเว็บเซิร์ฟเวอร์ว่าใช้ระบบปฏิบัติการแบบใดอยู่ มีโปรแกรมลักษณะใดให้ใช้ได้บ้าง และการใช้งานควรมีมาตรฐานความปลอดภัยของข้อมูลพอสมควรไม่ควรให้สามารถเรียกส่วนอื่น ๆ นอกจากที่ระบุไว้ได้ เพราะอาจเกิดปัญหาเรื่องความปลอดภัยของข้อมูลกับระบบได้



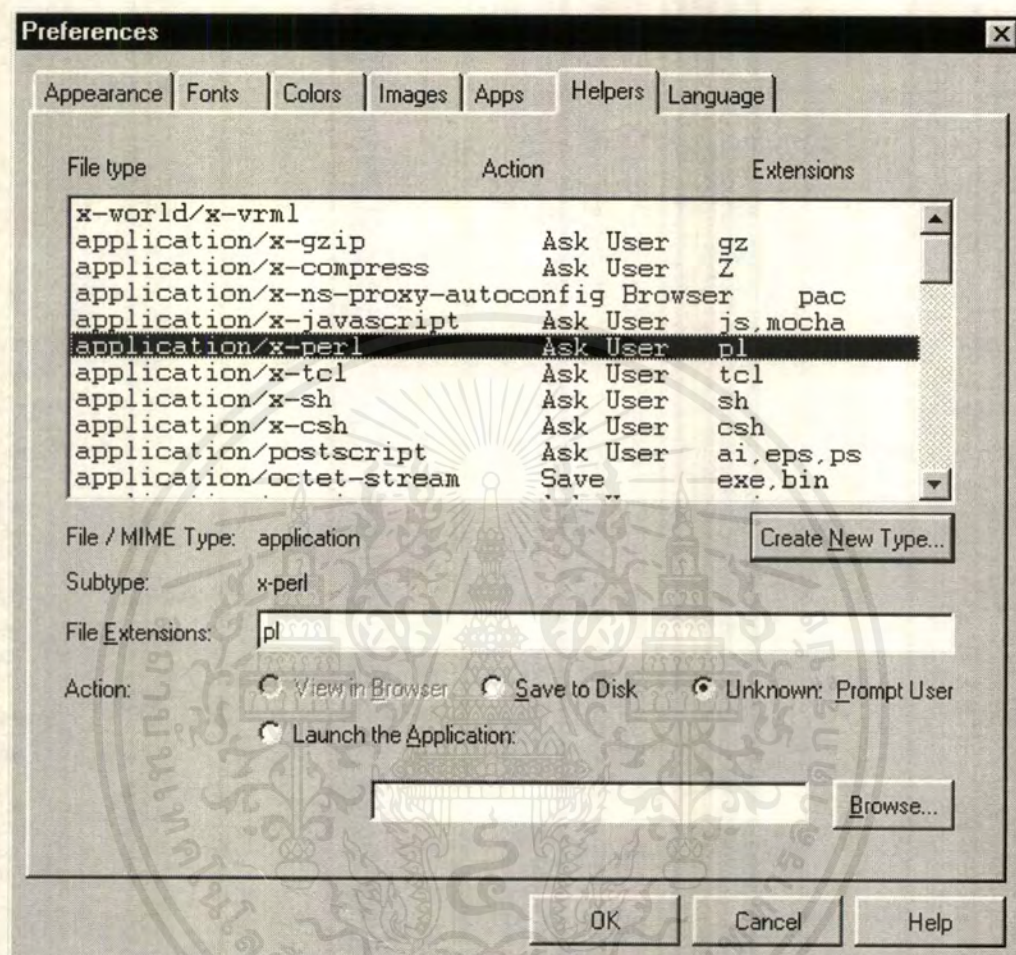
รูปที่ 4.2 กราฟแสดงการใช้งานภาษาต่างๆ ในงาน CGI

4.1.3.2 การบอกลักษณะข้อมูลที่ส่ง

ชนิดของข้อมูลที่ต้องการให้แสดงผลโดยโปรแกรมบราวเซอร์นั้น ๆ จะถูกกำหนดจากประเภทหรือชนิดของแฟ้มข้อมูลนั้น ๆ โดยในการใช้งานเว็ลด์ไวด์เว็บเองบางครั้งอาจพบกับชนิดของแฟ้มข้อมูลที่ไม่ทราบประเภทเว็บเบราว์เซอร์ก็จะถามผู้ใช้ว่าต้องการจัดเก็บหรือไม่ ซึ่งถ้าทราบประเภทเราอาจกำหนดให้เบราว์เซอร์เรียกโปรแกรมที่สามารถทำงานกับแฟ้มข้อมูลประเภทนั้นเลยก็ได้ เช่นถ้าพบแฟ้มข้อมูลที่เป็นแฟ้มเสียงก็อาจได้ยินเสียงเลย การจะทำงานในลักษณะนี้ได้จะต้องอาศัยการระบุโปรแกรมสำหรับแสดงผลแฟ้มข้อมูล ซึ่งเราจะต้องบอกลักษณะของข้อมูลในการรับการส่งด้วยว่าส่งอะไรมาให้ โดยคู่ที่ content-type ที่อยู่ในส่วนของ Header field ลักษณะการบอกชนิดของข้อมูลจะได้มา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากส่วนที่ใช้งานในการบอกประเภทข้อมูลในการส่งจดหมาย ที่เรียกว่า MIME (Multipurpose Internet Mail Extensions)



รูปที่ 4.3 แสดงการกำหนดชนิดเพิ่มข้อมูลใน Netscape

4.1.3.3 มาตรฐานของตัวแปร CGI

โปรแกรม CGI จะเริ่มต้นทำงานเมื่อมีเรียกใช้ผ่านทาง URL เซิร์ฟเวอร์จะติดต่อกับโปรแกรม CGI ผ่าน Environment variables และบางส่วนผ่านทาง command line โปรแกรม CGI จะอ่านและแปลข่าวสารที่ได้รับมา และประมวลผลเพื่อให้ได้ผลลัพธ์ที่ต้องการ และส่งข้อมูลที่ได้กลับไปยังบราวเซอร์โดยโปรแกรม CGI จะติดต่อกับเซิร์ฟเวอร์ผ่าน stdout ข้อมูลส่วนใหญ่จะส่งไปยังบราวเซอร์โดยที่เซิร์ฟเวอร์ไม่มีการแก้ไข ค่า Environment variables ที่เซิร์ฟเวอร์ส่งไปให้กับ CGI มีดังนี้

4.1.3.3.1 ข้อมูลเกี่ยวกับ User

HTTP_USER_AGENT ตัวแปรนี้จะบรรจุชื่อและเวอร์ชันของบราวเซอร์ของ user โดย มีรูปแบบคือ “name/version library/version” อาจบรรจุข้อมูลเกี่ยวกับ proxy gateway มาด้วย ตัวอย่าง

Mozilla/0.0b1J (Windows; I; 32bit)

Mozilla/1.22 (Windows; I; 16bit)

Lynx/2.6 libwww-FM/2.14

Mozilla/1.22 (Windows; I; 16 bit) via proxy gateway CERN-HTTPD/3.0 libwww/2.17

Charlotte/1.2b2 VM_ESA/1.2.2 CMS/11 via proxy gateway CERN-HTTPD/3.0pre5

HTTP_ACCEPT ประเภทของ MIME (Multipurpose Internet Mail Extensions) ที่ บราวเซอร์สามารถเข้าถึงได้ รูปแบบของ HTTP_ACCEPT คือ “type/subtype, type/subtype, [...]” ในตัวอย่างจะใส่ชื่อของบราวเซอร์ไว้ที่บรรทัดแรก และใส่ค่าของ HTTP_ACCEPT ไว้ในบรรทัดถัดไป

from Mozilla/2.0b1J --

image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*

from Lynx/2.3.7 BETA libwww/2.14 --

/, application/x-wais-source, application/html, text/plain, text/html, www/mime, application/x-ksh, application/x-sh, application/x-csh, application/x-sh

from IBM WebExplorer /v1.01 --

/; q=0.300, application/octet-stream; q=0.100, text/plain, text/html, image/bmp, image/jpeg, image/tiff, image/x-xbitmap, image/gif, application/zip, application/inf, audio/x-wav, audio/x-aiff, audio/basic, video/avs-video, video/x-msvideo, video/quicktime, video/mpeg, image/x-bitmap, application/editor

REMOTE_HOST and **REMOTE_ADDR** จะให้ข้อมูลเกี่ยวกับ IP address ของ user โดย **REMOTE_ADDR** จะบรรจุค่า IP address และ **REMOTE_HOST** จะบรรจุชื่อโฮสต์ของ address นั้น คุณจะเห็น **REMOTE_ADDR** เสมอ แต่บางครั้งอาจจะไม่มี **REMOTE_HOST** เพราะมีหลายเซิร์ฟเวอร์ที่ปิดพีเจอร์นี่เพื่อความปลอดภัยหรือด้วยเหตุผลอื่นก็ตาม ก็อาจทำให้มองไม่เห็น **REMOTE_HOST**

REMOTE_HOST = diamond.ce.kmitl.ac.th

REMOTE_ADDR = 161.246.4.3

4.1.3.3.2 ข้อมูลเกี่ยวกับ server

SERVER_SOFTWARE บรรจุนามและเวอร์ชันของเซิร์ฟเวอร์ software ในรูปแบบที่ใช้แสดงคือ “name/version”

SERVER_SOFTWARE = Apache/0.6.4b

SERVER_NAME บรรจุนามของ host server, DNS หรือ IP address ซึ่งปรากฏอยู่ใน URL Uniform Resource Locator

SERVER_NAME = personal.ce.kmitl.ac.th

SERVER_NAME = www.beamnet.com

GATEWAY_INTERFACE บรรจุ revision ของ CGI ซึ่งเซิร์ฟเวอร์ใช้ขณะนั้น รูปแบบคือ “CGI/revision”

GATEWAY_INTERFACE = CGI/1.1

QUERY_STRING เป็นวิธีที่ใช้ส่งข้อมูลผ่านไปให้โปรแกรม CGI ตัวอย่าง ถ้า URL เป็น HTTP ://www.beamnet.com/cgi/test.cgi?quick.brown.fox แล้วตัวอักษรทั้งหมดที่ตามหลัง “?” จะถูกใส่ไว้ใน QUERY_STRING ซึ่งค่าที่เก็บก็คือ “quick.brown.fox”

QUERY_STRING = GET

SCRIPT_NAME เป็นชื่อเพิ่มข้อมูลของโปรแกรม CGI

SCRIPT_NAME = /cgi/sample.pl.cgi

SERVER_PROTOCOL บรรจุนามและ revision ของ protocol ที่ร้องขอเข้ามาในรูปแบบฟอร์ม รูปแบบคือ “protocol/revision” ซึ่งเกือบจะแน่นอนที่จะเป็น “HTTP/1.0” ในตอนนี้

SERVER_PORT เป็นหมายเลขของ port ที่ร้องขอเข้ามา ซึ่งตามปกติแล้วจะเป็น “80” เพราะเป็นมาตรฐานพอร์ตของ HTTP

PATH_INFO and **PATH_TRANSLATED** แสดงเส้นทางอื่นที่ส่งผ่านข้อมูลไปยังโปรแกรม CGI ซึ่งคุณสามารถส่งผ่านเพิ่มข้อมูลไปยังโปรแกรมด้วยวิธีง่าย ๆ โดยการเพิ่มเติมมันต่อท้าย URL เช่น “http://www.beamnet.com/cgi/program.cgi/a/b/c” PATH_INFO จะบรรจุ path พิเศษนี้ (/a/b/c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น. อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ PATH_TRANSLATED จะบรรจุ PATH_INFO เพิ่มเติมเข้าไปในส่วนของ root path ของเซิร์ฟเวอร์ (/var/web/luna) ดังตัวอย่าง

```
PATH_INFO = /a/b/c
```

```
PATH_TRANSLATED = /var/web/luna/a/b/c
```

CONTENT_TYPE ชนิดของคำถามที่ติดมากับข้อมูล เช่นเมื่อมีการ post เข้ามา มันคือ MIME content-type ของข้อมูลในแบบฟอร์ม type/subtype ส่วน CONTENT_LENGTH คือจำนวนของข้อมูล

```
CONTENT_TYPE = application/x-www-form-urlencoded
```

```
CONTENT_LENGTH = 17
```

AUTH_TYPE เป็น protocol-specific authentication สำหรับจำกัดผู้ใช้

REQUEST_METHOD คือวิธีการที่ใช้ร้องขอ มันจะบอกคุณว่าข้อมูลต่าง ๆ จะถูกส่งผ่านมาได้ด้วยวิธีใด ซึ่งโดยทั่วไปจะต้องเลือกอย่างใดอย่างหนึ่งระหว่าง POST หรือ GET

4.1.3.3 ตัวอย่างการอ่านข้อมูล Environment Variables ใน sh

```
#!/bin/sh
# Filename: vars.sh.cgi
# Send the MIME header
echo Content-type: text/plain
echo

# display all the standard CGI variables
echo GATEWAY_INTERFACE = $GATEWAY_INTERFACE
echo REQUEST_METHOD = $REQUEST_METHOD
echo SCRIPT_NAME = $SCRIPT_NAME
echo QUERY_STRING = $QUERY_STRING
echo SERVER_SOFTWARE = $SERVER_SOFTWARE
echo SERVER_NAME = $SERVER_NAME
echo SERVER_PROTOCOL = $SERVER_PROTOCOL
echo HTTP_USER_AGENT = $HTTP_USER_AGENT
echo HTTP_ACCEPT = $HTTP_ACCEPT
echo PATH_INFO = $PATH_INFO
```

```

echo PATH_TRANSLATED = $PATH_TRANSLATED
echo REMOTE_HOST = $REMOTE_HOST
echo REMOTE_ADDR = $REMOTE_ADDR
echo REMOTE_USER = $REMOTE_USER
echo REMOTE_AGENT = $REMOTE_AGENT
echo AUTH_TYPE = $AUTH_TYPE
echo CONTENT_TYPE = $CONTENT_TYPE
echo CONTENT_LENGTH = $CONTENT_LENGTH

```

4.1.3.3.4 ตัวอย่างการอ่านข้อมูล Environment Variables ใน Perl

```

#!/usr/local/bin/perl
# Filename: vars.perl.cgi
# Send the MIME header
print "Content-type: text/html\n\n";

# display all the standard CGI variables
print "<B>GATEWAY_INTERFACE</B>: $ENV{'GATEWAY_INTERFACE'}
<BR>\n";
print "<B>REQUEST_METHOD</B>: $ENV{'REQUEST_METHOD'}<BR>\n";
print "<B>สคริปต์_NAME</B>: $ENV{'สคริปต์_NAME'}<BR>\n";
print "<B>QUERY_STRING</B>: $ENV{'QUERY_STRING'}<BR>\n";
print "<B>SERVER_NAME</B>: $ENV{'SERVER_NAME'}<P>\n";
print "<B>SERVER_SOFTWARE</B>: $ENV{'SERVER_SOFTWARE'}<BR>\n";
print "<B>SERVER_PROTOCOL</B>: $ENV{'SERVER_PROTOCOL'}<BR>\n";
print "<B>SERVER_PORT</B>: $ENV{'SERVER_PORT'}<BR>\n";
print "<B>HTTP_ACCEPT</B>: $ENV{'HTTP_ACCEPT'}<BR>\n";
print "<B>HTTP_USER_AGENT</B>: $ENV{'HTTP_USER_AGENT'}<P>\n";
print "<B>PATH_INFO</B>: $ENV{'PATH_INFO'}<BR>\n";
print "<B>PATH_TRANSLATED</B>: $ENV{'PATH_TRANSLATED'}<BR>\n";
print "<B>REMOTE_HOST</B>: $ENV{'REMOTE_HOST'}<BR>\n";
print "<B>REMOTE_ADDR</B>: $ENV{'REMOTE_ADDR'}<BR>\n";
print "<B>REMOTE_USER</B>: $ENV{'REMOTE_USER'}<BR>\n";
print "<B>REMOTE_IDENT</B>: $ENV{'REMOTE_IDENT'}<BR>\n";

```

```

print "<B>AUTH_TYPE</B>: $ENV{'AUTH_TYPE'}<BR>\n";
print "<B>CONTENT_TYPE</B>: $ENV{'CONTENT_TYPE'}<BR>\n";
print "<B>CONTENT_LENGTH</B>: $ENV{'CONTENT_LENGTH'}<P>\n";

```

4.1.3.3.5 ตัวอย่างการอ่านข้อมูล Environment Variables ใน C

```

/*
Filename: vars.c
*/
#include <stdio.h>
#include <stdlib.h>

/* temporary storage for the environment variable */
char * cp;
char * empty = "<empty>";
main(int argc, char ** argv)
{
/* send the mime-type header */
printf("Content-type: text/plain\n\n");

/* macro for displaying environment variables*/
#define safenv(a) ((cp = getenv(a)) ? cp : empty)

/*display all the standard CGI variables */
printf("GATEWAY_INTERFACE = %s\n", safenv("GATEWAY_INTERFACE"));
printf("REQUEST_METHOD = %s\n", safenv("REQUEST_METHOD"));
printf("SCRIPT_NAME = %s\n", safenv("SCRIPT_NAME"));
printf("QUERY_STRING = %s\n", safenv("QUERY_STRING"));
printf("SERVER_SOFTWARE = %s\n", safenv("SERVER_SOFTWARE"));
printf("SERVER_NAME = %s\n", safenv("SERVER_NAME"));
printf("SERVER_PROTOCOL = %s\n", safenv("SERVER_PROTOCOL"));
printf("HTTP_USER_AGENT = %s\n", safenv("HTTP_USER_AGENT"));

```

```

printf("HTTP_ACCEPT = %s\n", safenv("HTTP_ACCEPT"));
printf("PATH_INFO = %s\n", safenv("PATH_INFO"));
printf("PATH_TRANSLATED = %s\n", safenv("PATH_TRANSLATED"));
printf("REMOTE_HOST = %s\n", safenv("REMOTE_HOST"));
printf("REMOTE_ADDR = %s\n", safenv("REMOTE_ADDR"));
printf("REMOTE_USER = %s\n", safenv("REMOTE_USER"));
printf("REMOTE_AGENT = %s\n", safenv("REMOTE_AGENT"));
printf("AUTH_TYPE = %s\n", safenv("AUTH_TYPE"));
printf("CONTENT_TYPE = %s\n", safenv("CONTENT_TYPE"));
printf("CONTENT_LENGTH = %s\n", safenv("CONTENT_LENGTH"));

```

Tcl/Tk (Tcl:Tool Command Language,Tk:Tool Kit) เรียกย่อ ๆ ว่า (Tickle) เป็นภาษาที่มีลักษณะเป็น interpreter ภาษาหนึ่งซึ่งสามารถใช้งานร่วมกับภาษา C ได้ เพราะถูกออกแบบให้ใช้งานในลักษณะเป็น C Library พัฒนาขึ้นโดย John Ousterhout ที่มหาวิทยาลัย California ที่ Berkley Tcl จะใช้งานในลักษณะภาษาสคริปต์ ในขณะที่ Tk จะใช้งานในลักษณะกราฟฟิกในแบบ X windows ภาษานี้มีใช้ทั้งบนเครื่องที่ใช้ระบบปฏิบัติการแบบยูนิกซ์และบน Windows95

สำหรับภาษาที่ไม่ใช่ภาษาสคริปต์ อย่างเช่นภาษาเบสิกก็สามารถนำมาเขียนเป็น CGI เพื่อใช้งานกับเว็บเซิร์ฟเวอร์ได้เช่นกัน โดยจะใช้งานกับเครื่องที่ไม่ได้ใช้ระบบปฏิบัติการแบบยูนิกซ์เป็นเว็บเซิร์ฟเวอร์เช่นเครื่องที่ใช้ WindowsNT หรือ Windows95 เป็นเว็บเซิร์ฟเวอร์ เป็นต้น

สรุปก็คือว่าเราสามารถใช้งานภาษาที่มีลักษณะเป็นแบบสคริปต์ เช่น Perl, Tcl, Python หรือ ยูนิกซ์เชลล์สคริปต์ หรือภาษาที่มีโครงสร้างอย่างเช่น C หรือ C++ สำหรับใช้งานเป็น CGI ได้โดยขึ้นกับชนิดของเว็บเซิร์ฟเวอร์ที่ใช้

4.1.4 การสร้างแบบฟอร์ม

ในการสร้างแบบฟอร์มนั้น จะประกอบด้วยข้อมูล 2 ส่วน ส่วนแรกเรียกว่าส่วนเชื่อมต่อ (Interface) และส่วนควบคุม (CGI)

ส่วนเชื่อมต่อนั้น เราจะสร้างด้วยภาษา HTML ส่วน CGI จะสร้างด้วยโปรแกรมภาษา perl ส่วนเชื่อมต่อนั้นจะสร้างเก็บไว้ในแฟ้มเอกสาร .html ก็ได้ แต่สำหรับส่วน CGI ต้องเก็บไว้ที่เว็บเซิร์ฟเวอร์(Server Side CGI) เพราะว่าขณะนี้ยังไม่มียโปรแกรมเว็บเบราว์เซอร์ตัวใดสนับสนุนไคลเอนท์ไซด์ CGI ที่จะทำให้เราสามารถเขียน CGI ผนวกเข้ากับแฟ้มเอกสาร .html ได้

ในการสร้างแบบฟอร์มนั้นจะเริ่มต้นด้วยคำสั่ง <FORM> จากนั้นจึงบรรจุคำสั่งอื่น ๆ ลงไปตามที่ต้องการ แล้วปิดท้ายด้วยคำสั่ง </FORM> เหมือนชุดคำสั่งทั่ว ๆ ไป โดยมีคำสั่งย่อยที่ทำให้เกิดช่องว่างสำหรับการเติมข้อความลงไป

<FORM ACTION=CGI_URL METHOD=METHOD TYPE>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับเข้าใต้งานไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<INPUT> ...
<INPUT>
</FORM>
```

METHOD มีแอดทริบิวต์ 2 ตัว คือ GET และ POST เป็นตัวกำหนดว่าข้อมูลจะถูกส่งมาประมวลผลโดยโปรแกรมที่เซิร์ฟเวอร์อย่างไร

ACTION เป็นชื่อโปรแกรมที่ถูกเรียกขึ้นมาประมวลผลที่ผู้ใช้กรอกลงไป มีตำแหน่งของ URL ที่ไปยังโปรแกรม CGI สคริปต์ บนเว็บเซิร์ฟเวอร์เป็นตัวประมวลผลข้อมูลที่ส่งมาจากฟอร์ม ส่วนในแบบฟอร์มนั้นจะเป็นข้อมูลตามปกติ จะมีเครื่องหมายควบคุมพิเศษ เพื่อสร้างแบบฟอร์มเช่น <INPUT ...>, <TEXTAREA ...>, และ <SELECT ...>

ENCTYPE ใช้สำหรับการเข้ารหัสเพื่อรักษาความลับของข้อมูลที่ส่ง ลักษณะของ <INPUT> Tag นั้นมีด้วยกันหลายประเภท ซึ่งมีรูปแบบดังนี้

```
<INPUT TYPE = "Text" NAME = "Name" VALUE = "Int Value">
```

TYPE	เป็นชนิดของข้อมูลเข้า ประกอบไปด้วย
TEXT	ข้อมูลเข้าชนิดข้อความ
PASSWORD	ข้อมูลเข้าชนิดรหัสผ่าน ไม่ใส่อักขระขณะพิมพ์
CHECKBOX	ข้อมูลเข้าชนิดตัวเลือกแบบสามารถเลือกได้มากกว่า 1 รายการ
RADIO	ข้อมูลเข้าชนิดตัวเลือกแบบสามารถเลือกได้เพียงรายการเดียวเท่านั้น
SUBMIT	ปุ่มกดสำหรับยืนยันข้อมูลที่ป้อนเข้าไป
RESET	ปุ่มกดสำหรับยกเลิกรายการที่ป้อนเข้าไปทั้งหมด ให้กลับไปใช้ค่าเริ่มต้นใหม่
NAME	เป็นชื่อของข้อมูลเข้า จะตั้งชื่ออย่างไรก็ได้ แต่ขอให้เข้าใจง่ายก็พอ
VALUE	เป็นค่าเริ่มต้นที่กำหนดให้กับข้อมูลแต่ละตัว

4.1.4.1 ข้อมูลชนิด TEXT

TEXT เป็นข้อมูลชนิดตัวอักษรหรือข้อความ ข้อมูลชนิดนี้จะบรรจุข้อความได้เพียงบรรทัดเดียวเท่านั้น และเมื่อเลือกชนิดข้อมูลเป็น TEXT แล้ว จะต้องกำหนดค่า

SIZE="ค่าความกว้างของรายการข้อมูลนั้น ๆ" และ MAXLENGTH="จำนวนตัวอักษรสูงสุดเท่าที่รายการข้อมูลนั้นจะรับได้ รูปแบบของแอดทริบิวต์ TEXT มีดังนี้

```
<INPUT TYPE="Type" NAME="Name" VALUE="Int Value" SIZE="Width"
MAXLENGTH="Maxlength">
```

4.1.4.2 ข้อมูลชนิด CHECKBOX

ข้อมูลชนิด CHECKBOX เป็นปุ่มตัวเลือกแบบเดียวกับปุ่มวิทยุ แตกต่างกันตรงที่ช่องทำเครื่องหมายสามารถเลือกรายการได้มากกว่าหนึ่ง รูปแบบของแอตทริบิวต์ CHECKBOX มีดังนี้

```
<INPUT TYPE="CHECKBOX" NAME="Name">
```

4.1.4.3 ข้อมูลชนิด RADIO

ข้อมูลชนิด RADIO BUTTON มีลักษณะเป็นปุ่มเหมือนลูกบิดหน้าปัดวิทยุ เมื่อผู้ฟังเลือกฟังรายการวิทยุความถี่ใดก็จะเลื่อนลูกบิดไปยังรายการนั้น รายการที่เคยเลือกฟังอยู่ก็จะเปลี่ยนไป ปุ่มวิทยุของ HTML ก็ไม่แตกต่างกันเท่าใด เพียงแต่ผู้ใช้เลื่อนเมาส์ไปคลิกที่ปุ่มวิทยุใด ตัวเลือกเก่าก็จะถูกยกเลิกไป ฟังได้เพียงคลื่นความถี่เดียวเท่านั้น ไม่มีรายการคลื่นแทรก ส่วนใหญ่เราใช้ปุ่มวิทยุกับตัวเลือกที่กำหนดให้กับผู้ใช้สำหรับการตัดสินใจอย่างใดอย่างหนึ่ง รูปแบบของแอตทริบิวต์ RADIO มีดังนี้

```
<INPUT TYPE="RADIO" NAME="Name">
```

ในการกำหนดแอตทริบิวต์เป็น RADIO นี้ NAME จะต้องเป็นชื่อเดียวกัน เพราะค่าที่เลือกจะต้องเปลี่ยนไปทุกครั้งที่มีการเลือกรายการใหม่ขึ้นมา

4.1.4.4 การส่งแบบฟอร์ม (SUBMIT)

หลังจากที่สร้างแบบฟอร์มขึ้นมาตามหัวข้อที่ต้องการแล้ว ยังไม่สามารถส่งข้อมูลไปยังเครื่องเซิร์ฟเวอร์ได้ ต้องจัดเตรียมปุ่มสำหรับรายการส่งแบบฟอร์มก่อน รูปแบบของแอตทริบิวต์ SUBMIT มีดังนี้

```
<INPUT TYPE="SUBMIT" NAME="Submit">
```

4.1.4.5 การยกเลิกแบบฟอร์ม (RESET)

แบบฟอร์มที่ผู้ใช้ตอบแต่ยังไม่ถูกส่งกลับมายังเว็บเซิร์ฟเวอร์ผู้ใช้สามารถยกเลิกรายการเลือกทั้งหมดให้กลับมาเป็นเสมือนแบบฟอร์มที่ยังไม่ถูกใช้งานได้ด้วยการกดปุ่มยกเลิก รูปแบบของแอตทริบิวต์ RESET มีดังนี้

```
<INPUT TYPE="RESET" NAME="Reset">
```

4.1.4.6 การสร้างปุ่มตัวเลือก <SELECTION>

ในการสร้างแบบฟอร์ม นอกเหนือไปจากการให้ผู้ใช้ต้องป้อนข้อมูลเอง การสร้างตัวเลือกให้ผู้ใช้สามารถเลือกคำตอบได้นั้นก็นับว่าเป็นทางเลือกหนึ่งที่ถูกนำมาใช้งานอย่างมากมาย

ปุ่มตัวเลือก <SELECT>...</SELECT> จัดว่าเป็นปุ่มตัวเลือกแบบเดียวกับปุ่มวิทยุเพราะมีเพียงตัวเลือกเดียวเท่านั้นที่เลือกได้ เพียงแต่ปุ่มตัวเลือกจะมีรายการแสดงอยู่ในตารางตัวเลือก มีปุ่มสไลด์สำหรับกรณีที่มีตัวเลือกมาก ๆ ทำให้สิ้นเปลืองเนื้อที่สำหรับการแสดงผลข้อมูลอื่น ๆ ได้ การเลือกใช้ปุ่มตัวเลือกก็เป็นหนทางหนึ่งที่ทำให้สามารถใช้พื้นที่บนจอภาพสำหรับการทำงานในส่วนอื่นได้อีก

<SELECT>...</SELECT> มี NAME เป็นตัวกำหนดชื่อให้แก่ส่วนควบคุมสำหรับการประมวลผลที่เซิร์ฟเวอร์และอาจมี SIZE และ MULTIPLE ระบุลงไปด้วย

<SELECT>...</SELECT> มี <OPTION> เป็นตัวกำหนดข้อความในตัวเลือกแต่ละตัว มี SELECTED เป็นตัวกำหนดค่าเริ่มต้น รูปแบบของ <SELECT>...</SELECT> มีดังนี้

```
<SELECT NAME="Name" MULTIPLE SIZE="Size">
```

```
<OPTION>
```

```
<OPTION>
```

```
.....
```

```
</SELECT>
```

ในการเพิ่มคำว่า MULTIPLE เข้าไป จะนำรายชื่อที่กำหนดขึ้นมาแสดงบนจอภาพทั้งหมด ทำให้พื้นที่ในการแสดงข้อมูลเหลือน้อยลง ถ้าต้องการประหยัดพื้นที่บนจอภาพต้องกำหนดรายการที่แสดงบนจอภาพในแต่ละครั้งให้น้อยลงด้วยการเพิ่มคำว่า SIZE ลงไป บนจอภาพจะแสดงรายการเท่ากับตัวเลขที่กำหนดเข้าไป

4.1.4.7 ข้อมูลชนิด TEXTAREA

คำสั่ง <TEXTAREA>...</TEXTAREA> ใช้สำหรับสร้างกรอบป้อนข้อความแบบหลายบรรทัด(multiline text-entry box) และยังกำหนดตัวอักษรเริ่มต้นที่แสดงในกรอบอีกด้วย รูปแบบของคำสั่ง <TEXTAREA>...</TEXTAREA> เป็นดังนี้

```
<TEXTAREA NAME="Name" ROWS=Row COLS=Col></TEXTAREA>
```

โดยที่

ROWS เป็นจำนวนแถวที่ใช้ในการป้อนข้อมูล

COLS เป็นจำนวนคอลัมน์สูงสุดที่ใช้ป้อนข้อมูลในแต่ละแถว

4.1.4.8 การรับข้อมูลจาก server

เมื่อโปรแกรม CGI ถูกเรียกใช้งาน โปรแกรม CGI จะรับพารามิเตอร์จาก Environment variable และอินพุตมาตรฐานเท่านั้น ตัวแปรที่สำคัญนี้คือ QUERY_STRING และ PATH_INFO

QUERY_STRING คือข้อความที่เพิ่มเติมมาตามหลังเครื่องหมาย "?" ใน URL ตัวอย่างเช่น HTTP ://www.myhost.com/cgi-bin/myprog.cgi จะเรียกใช้โปรแกรมชื่อ myprog.cgi จาก directory ชื่อ cgi-bin ซึ่งเก็บไว้ที่ www.myhost.com การส่งผ่านพารามิเตอร์ทำได้โดยการเรียก

```
http://www.myhost.com/cgi-bin/myprog.cgi?mydata is here
```

parameter จะถูกเก็บไว้ในตัวแปร QUERY_STRING ในรูปแบบดังนี้

```
QUERY_STRING = mydata+is+here
```

ข้อความที่เข้ามาจะถูกเข้ารหัสโดยช่องว่างจะถูกเปลี่ยนเป็น + และอักขระพิเศษจะถูกเปลี่ยนเป็นเลข 2 หลักที่มีเครื่องหมาย % นำหน้า (%XX) ซึ่งโปรแกรม CGI จะต้องทำการถอดรหัสก่อนนำไปใช้งาน

4.1.4.9 การส่งแบบฟอร์มไปยังโปรแกรม CGI

ใช้วิธีการ GET หรือ POST เพื่อรับข้อมูลจากแบบฟอร์ม ซึ่งแต่ละวิธีจะมีการให้ข้อมูลที่แตกต่างกัน

```
<FORM ACTION=CGI_URL METHOD=GET>
```

ข่าวสารที่ไคลเอนต์ส่งไปยังโปรแกรม CGI จะเก็บไว้ในตัวแปร QUERY_STRING

```
<FORM ACTION=CGI_URL METHOD=POST>
```

ข่าวสารที่ไคลเอนต์ส่งไปยังโปรแกรม CGI จะส่งผ่าน STDIN แต่จะไม่มีรหัส EOF (End Of File) ซึ่งโปรแกรม CGI จะทราบขนาดของข้อมูลโดยตัวแปรที่ชื่อ CONTENT_LENGTH

4.1.4.10 รูปแบบการเข้ารหัสของข้อมูล

ทั้ง GET และ POST จะมีรูปแบบการเข้ารหัสที่เหมือนกัน คือข้อมูลจะเป็นคู่ของชื่อและค่า (name.value) ในรูปแบบ NAME=VALUE และจะใช้เครื่องหมาย "&" คั่นระหว่างข้อมูลแต่ละคู่ ใช้สัญลักษณ์ "+" แทนช่องว่าง และใช้ %XX แทนอักขระพิเศษ ดังตัวอย่าง

```
<FORM METHOD=POST>
```

```
<INPUT NAME="A"> (Input = "A B C")
```

```
<INPUT NAME="B"> (Input = "12345")
```

```
<INPUT NAME="C"> (Input = "DE")
```

ซึ่งโปรแกรม CGI จะรับค่า Environment variable เป็นดังนี้

```
CONTENT_LENGTH=20
```

```
STDIN: A=A+B+C&B=12345&C=DE
```

4.1.5 ประเภทของข้อมูล CGI ที่ส่งไปยังไคลเอนต์

โปรแกรม CGI สามารถส่งข้อมูลได้หลายประเภท

- An Image
- An HTML document
- An plain-text document
- An audio clip

ส่วนข้อมูลประเภทอื่นจะถูกกำหนดใน MIME type ซึ่ง CGI จะต้องบอกให้ไคลเอนท์ทราบว่าได้ส่งข้อมูลประเภทใดมาด้วยบราวเซอร์จะได้จัดแสดงได้อย่างถูกต้อง CGI จะปะส่วนหัวซึ่งเป็นข้อความ text จบด้วย line feed หรือ newline ดังตัวอย่าง

```
Content-type: text/HTML หรือ Content-type: image/gif
```

แทนที่โปรแกรม CGI จะสร้างเพิ่มข้อมูล HTML ขึ้นมา โปรแกรม CGI สามารถชี้ให้บราวเซอร์ไปดึงข้อมูลมาจากที่อื่นได้โดยใช้คำสั่งดังนี้

```
Location : HTTP ://www.myserver.com/document.html
```

4.2 จาวาเบื้องต้น

4.2.1 จาวาคืออะไร

หลายๆ คนในตอนนี้น่าเคยได้ยินเรื่องราวของจาวามาพอสมควร ในโครงการนี้ได้ใช้จาวาในการพัฒนาส่วนหนึ่งแต่ด้วยโครงการนี้ เน้นในการนำจาวามาใช้ในการเขียนโปรแกรมบนระบบเครือข่ายมากกว่าจะมาเขียนในงานทางด้านกราฟฟิกหรืองานอื่นๆ ที่จาวาทำได้และเนื่องด้วยมีผู้ที่ทำโครงการเกี่ยวกับจาวาทางด้านอื่นๆ มากแล้ว เช่น จาวาเกี่ยวกับดาต้าเบส ,จาวาเกี่ยวกับกราฟฟิก จึงอยากให้ผู้สนใจเกี่ยวกับจาวาเบื้องต้นหรือจาวาเกี่ยวกับงานดาต้าเบส หรือ กราฟฟิก ได้ศึกษาในโครงการอื่นๆ แทน ในโครงการนี้จะพูดถึงแต่การเขียนโปรแกรมจาวาบนระบบเครือข่ายเท่านั้น จึงจะขอล่าเกี่ยวกับจาวาเบื้องต้นไว้ไม่มาก

แท้ที่จริงแล้ว จาวาก็เป็นเพียงภาษาคอมพิวเตอร์ชนิดหนึ่งเช่นเดียวกับภาษาคอมพิวเตอร์ต่างๆ เพียงแต่ความสามารถของตัวเองที่ทำให้มีคนพูดถึงกันค่อนข้างมากเท่านั้น คุณสมบัติอันแรกทีพูดถึงกันบ่อยๆ ก็คือ ภาษาที่ใช้เขียนโปรแกรมภาษาแรกที่เป็นมัลติแพลตฟอร์ม พุดง่ายๆ ก็คือเขียนโปรแกรมขึ้นมาโปรแกรมเดียวเมื่อคอมไพเลอร์เสร็จคุณก็สามารถนำโปรแกรมให้ทำงานบนเว็ลด์ไวด์เว็บต้นกำเนิดของจาวานั้นเริ่มมาตั้งแต่ปี 2533 โดย Patrick Naughton ซึ่งเป็นวิศวกรซอฟต์แวร์อยู่ที่ Sun Microsystems ภาษาจาวาที่ได้สามารถสร้างแอปพลิเคชันที่ทำงานโดยลำพังและแอปเพล็ตซึ่งเป็นโปรแกรมที่ทำงานผ่านเว็บเบราว์เซอร์ หรือใช้ Applet Viewer ช่วยก็ได้ แอปพลิเคชันที่เขียนด้วยจาวานั้นแตกต่างจากแอปพลิเคชันทั่วไปตรงที่เมื่อใช้งานบนเน็ตเวิร์กและแอปพลิเคชันเหล่านี้จะต้องอยู่บนเซิร์ฟเวอร์เท่านั้น เช่น ถ้าโหลดเว็บเพจที่มีแอปเพล็ตบนเบราว์เซอร์ แอปเพล็ตนี้จะถูกส่งมาบนเครื่องคุณเพื่อทำงาน การใช้จาวาเขียนโปรแกรมจะช่วยประหยัดค่าใช้จ่ายในการพัฒนาโปรแกรม เนื่องจากคุณสมบัติมัลติแพลตฟอร์มนอกจากนี้ยังเพิ่มความสะดวกเพราะแอปพลิเคชันเก็บอยู่ที่เซิร์ฟเวอร์ส่วนกลาง ดังนั้นเมื่อมีการเปลี่ยนแปลงจึงสามารถทำได้ที่จุดเดียว ผู้ใช้ไม่ต้องมีภาระใดๆ เลย

4.2.2 คำศัพท์เกี่ยวกับโอโอพีที่ควรทราบ

□ **Object, Class และ Instance** คำศัพท์ทั้งสามนี้ คุณจะพบได้บ่อย ๆ เมื่อศึกษาภาษาที่มีลักษณะออบเจกต์โอเรียนเตด ซึ่งบางครั้งผมก็รู้สึกวุ่นวายกับกับความหมายของคำทั้งสามนี้เหมือนกัน สำหรับคลาสและออบเจกต์นั้นมีความหมายใกล้เคียงกันจนบางคนก็เหมาว่าเหมือนกัน แต่จริง ๆ แล้วทั้งสองคำนี้มีความหมายต่างกันพอสมควร ในชีวิตจริงคุณจะเห็นว่าสิ่งของมากมายอยู่บนโลกนี้ เช่นขนมเค้ก เครื่องบิน รถยนต์ เป็นต้น สิ่งของเหล่านี้มันก่อนที่จะสร้างขึ้นมาก็จะต้องได้รับการออกแบบเสียก่อน เช่น ขนมเค้กเป็นขนมที่ทำด้วยแป้งลักษณะกลมหรือสี่เหลี่ยมก็ได้ เครื่องบินจะต้องมีเครื่องยนต์ มีปีก หางเสือ และที่สำคัญบินได้ ส่วนรถยนต์ จะต้องมียี่ห้อ มีเครื่องยนต์เพื่อสร้างกำลังในการวิ่ง ต้นแบบของสิ่งของเหล่านี้เรียกว่า คลาส สิ่งของแต่ละชนิดบนโลกนี้จะสามารถจัดแบ่งเป็นกลุ่ม ๆ โดยอาศัยคลาสเป็นตัวแยก เช่นถ้าเราจะจัดกลุ่มขนมเค้ก เราสามารถบอกว่าเค้กจัดเป็นคลาสขนมอบฟรังก์ ซึ่งในกลุ่มนี้อาจมีคุกกี้พาย แต่ถ้าคุณจะบอกว่าเค้กควรจัดเป็นขนมหวานก็ไม่ผิด แต่ถ้าบอกว่าเป็นขนมหวาน ในกลุ่มนี้ก็อาจนำเอาทองหยิบ ฝอยทอง ทองหยอด ข้ามารวมในกลุ่มได้เหมือน สรุปก็คือ คลาสจะเป็นลักษณะที่ใช้บรรยายถึงสิ่งของที่มีอยู่ คุณสามารถใช้ลักษณะหรือคุณสมบัติเหล่านี้มาใช้จัดกลุ่มสิ่งของได้ คราวนี้มาดูออบเจกต์บ้างถ้าจะถามว่าออบเจกต์คืออะไรก็ต้องตอบว่าออบเจกต์คือสิ่งของแต่ละชิ้นที่มีอยู่จริงบนโลกนี้ เช่นจากที่ผมพูดถึงขนมเค้กออบเจกต์เค้กที่อยู่บนคลาสนขนมอบฟรังก์ก็มีอยู่จริง ดังนั้นจึงอาจพูดว่าคลาสเป็นสิ่งที่ไม่มีอยู่จริง เป็นลักษณะนามธรรมที่ใช้บรรยายให้เห็นถึงสิ่งของแต่ละชนิดหรือออบเจกต์ก็ได้ ส่วนอินสแตนซ์จะหมายถึงออบเจกต์แต่ละอันที่สร้างขึ้นจากคลาสที่เราบรรยายไว้ เช่นอินสแตนซ์ในกรณีก็คือ เค้กนาร่อยแต่ละชิ้นที่คุณเห็นแล้วจะรู้สึกหิวกาแพขึ้นมาทันที

□ **Encapsulation** ขอฟักเรื่องขนมเค้กไว้ก่อน คราวนี้พิจารณาถึงของแต่ละชิ้นบนโลกนี้ที่ใช้คลาสเป็นตัวบรรยายลักษณะนั้น เช่น รถยนต์โดยมากก็มีสี่ล้อ เครื่องหนึ่งเครื่องแบบไหนก็แล้วแต่ยี่ห้อ มีเกียร์มือหรือออโต้ก็ขึ้นกับราคา วิ่งได้ หยุดได้ เร่งความเร็วและเปลี่ยนเกียร์ได้ ทั้งหมดที่ผมพูดมานี้เป็นลักษณะของรถยนต์ หรือจะบอกว่าสิ่งใดก็ตามที่มีลักษณะตามนี้มักจะเรียกว่าเป็นคลาสรถยนต์ คุณจะเห็นว่าสิ่งที่จะมาประกอบกันเป็นคลาสนี้จะแบ่งออกเป็น 2 ส่วนก็คือ ส่วนที่บอกสถานะ เช่น จำนวนล้อ ประตู ประเภทเครื่องยนต์ ชนิดเกียร์ เป็นต้น และอีกส่วนหนึ่งก็คือพฤติกรรมหรือหน้าที่ เช่น วิ่ง หยุด เร่งความเร็ว หรือเปลี่ยนเกียร์ ในการเขียนโปรแกรมด้วยจาวาเราก็จะนำวิธีการสร้างคลาสนี้มาใช้เช่นกัน โดยเราจะใช้ตัวแปรแทนสถานะของคลาสนี้ และเมธอดจะใช้บรรยายถึงพฤติกรรมของคลาสนี้ ในจาวาเราเรียกลักษณะที่ทุกอย่างจะต้องบรรจุไว้ในกรณีนิยามคลาสนี้ว่าเป็นเอนแคปซูเลชัน สำหรับจาวาแล้วคุณจะเห็นว่าแตกต่างไปจากภาษาอื่น เพราะจะไม่มีฟังก์ชันที่กำหนดไว้โดยในโปรแกรม ไม่มีตัวแปรในลักษณะโกลบอลเพราะทุกสิ่งทุกอย่างต้องกำหนดไว้ในนิยามของคลาสนี้หมด

□ **Inheritance** คุณสมบัติอินสแตนซ์ในจาวานั้นก็มีให้เช่นกัน แต่มันยังค่อนข้างเคร่งครัดมากอีกด้วยในการนิยามคลาสนั้น เราสามารถนำคลาสดั้งเดิมที่มีอยู่ก่อนมาใช้กำหนดคลาสนิยามขึ้นมาได้ เช่นกรณีที่เรามีคลาสรถยนต์อยู่ก่อนแล้ว หากเราต้องการออกแบบรถแข่งซึ่งจัดอยู่ในคลาสรถยนต์เช่นกัน เราสามารถใช้คลาสรถยนต์ที่มีอยู่มาบรรยายขยายคุณสมบัติเพิ่มเติม เช่น กำหนดขนาดเครื่องยนต์ใหม่ ไม่ต้องมีประตู ลดความสูงลง เปลี่ยนยางชนิดใหม่ แต่คลาสรถยนต์ก็ยังมียุทธลักษณะของความเป็นรถ

ย่นย่อเช่นเดิม เราเรียกการถ่ายทอดคุณลักษณะของคลาสแบบนี้ว่า เป็นอินเฮอริเทนซ์ ในจาวานั้นการบรรยายคลาสใหม่โดยใช้คลาสเดิมนั้นคุณจะต้องอ้างถึงคลาสเดิมที่มีอยู่แค่เพียงชนิดเดียวเท่านั้น ต่างจากใน C++ ที่อ้างถึงได้หลายๆ คลาสพร้อมๆกัน การที่คุณจะสืบทอดคุณลักษณะของคลาสได้เพียงคลาสเดิมนั้นมีประโยชน์ที่จะช่วยลดความยุ่งยากในการเขียนโปรแกรม เพราะเวลาตรวจสอบโปรแกรมคุณจะไม่สับสนมากเกินไป

□ **Interface** เป็นกลุ่มของนิยามของเมธอดและค่าคงที่ ดูเหมือนผู้ออกแบบจาวาตั้งใจจะใช้อินเทอร์เฟซนี้มาแก้ไขปัญหาที่เราไม่สามารถคลาสที่ถ่ายทอดคุณสมบัติจากคลาสอื่นมากกว่าหนึ่งคลาสมาใช้ได้ ทั้งนี้เนื่องจากในขั้นตอนการเขียนโปรแกรมจริง ๆ บางครั้งเราจำเป็นต้องกำหนดให้ออบเจกต์บางตัวมีพฤติกรรมพิเศษที่นอกเหนือออกไป เช่น เราต้องการสร้างกล่องอิเล็กทรอนิกส์ สร้างเสียงหลาย ๆ ชนิดไม่ว่าจะเป็นเครื่องดนตรี เสียงสัตว์ เสียงลมได้คุณจะต้องออกแบบคลาสกล่องนั้นให้มีเมธอดที่รับเอาพฤติกรรมของเครื่องดนตรี เช่น กีตาร์ ไวโอลิน กลอง เปียโน แจกโซโฟน เป็นต้น มาใช้ในตัวเอง ซึ่งถ้าใช้ลักษณะอินสแตนซ์คุณจะนำเสียงเครื่องดนตรีมาใช้ได้เพียงชนิดเดียว แต่ถ้าใช้คุณสมบัติอินเทอร์เฟซ คุณจะสามารสร้างอินเทอร์เฟซที่รวมเอาเมธอดสร้างเสียงดนตรีแต่ละอย่างในตัวรวมไว้เป็นกลุ่มอินเทอร์เฟซช่วยให้เราเข้าถึงเมธอดสร้างเสียงของออบเจกต์เครื่องดนตรีจริงได้ นอกจากนี้ยังรวมเมธอดสร้างเสียงสัตว์ต่าง ๆ เข้ามาอีก ดังนั้นจะเห็นว่าสามารถใช้อินเทอร์เฟซเพื่อรวมพฤติกรรมของคลาสที่ไม่มีความสัมพันธ์กันไว้ด้วยกัน ในกรณีเป็นเครื่องดนตรีกับสัตว์แต่ละขณะที่ใช้อินเทอร์เฟซคุณต้องไม่ลืมว่าคุณไม่อาจถ่ายทอดกัวแปรจากอินเตอร์เฟซมาใช้ ความสัมพันธ์ของสิ่งต่าง ๆ ที่อยู่ในอินเทอร์เฟซไม่ได้อยู่ในโครงสร้างลำดับชั้นเหมือนในคลาส

□ **Method** หากเปรียบเทียบกับภาษารุ่นเก่า ๆ เมธอดก็คือ ฟังก์ชันที่ทำหน้าที่ต้องที่ผู้เขียนโปรแกรมต้องการ แต่ถ้านำแนวคิดของโอโอพีมาใช้ก็ต้องบอกว่าเมธอดเป็นฟังก์ชันที่ใช้แสดงพฤติกรรมของออบเจกต์ที่สร้างจากคลาส เช่น ออบเจกต์คลาสรถยนต์จะต้องมีเมธอดสำหรับสั่งให้วิ่ง เปลี่ยนเกียร์ หยุดรถ แต่ละเมธอดถูกเอนแคปซูเลทไว้ในคลาส คุณไม่จำเป็นต้องไปสนใจในรายละเอียดของแต่ละเมธอด เช่นหากต้องการเปลี่ยนเกียร์ก็ไม่ต้องรอขั้นตอนย่อย ๆ ว่าจะต้องมีการย้ายเฟืองเกียร์อย่างไร แผ่นคลัทซ์ทำหน้าที่อะไร เพียงแต่บอกกับออบเจกต์นั้นให้เปลี่ยนเกียร์อย่างเดียวบ วิธีการเรียกใช้เมธอดนี้ช่วยลดขั้นตอนการเขียน โปรแกรมลงได้ เพราะคุณไม่ต้องรับผิดชอบรายละเอียดย่อยของออบเจกต์เลย

□ **Method overriding** .ในการเขียนโปรแกรมด้วยจาวานั้นหากคุณสร้างคลาสรุ่นใหม่โดยอินเฮอริเทนซ์คลาสอื่นมาใช้ คุณสามารถนำเมธอดที่มีในคลาสบรรพบุรุษมาแก้ไขหรือเพิ่มเติมคุณสมบัติบางอย่างเข้าไปได้ แต่มีเงื่อนไขว่าต้องใช้ชื่อเมธอดเหมือนกัน คำที่คืนจากเมธอดต้องเป็นชนิดเดียวกันจำนวนและชนิดอาร์กิวเมนต์ต้องเหมือนกันด้วย เราเรียกลักษณะดังกล่าวนี้ว่าเป็นการโอเวอร์ไรด์เมธอดมาใช้ เช่น คุณต้องสร้างคลาสรถยนต์รุ่นใหม่ที่ทันสมัยกว่ารุ่นเก่า คุณก็ต้องอินเฮอริเทนซ์ลักษณะของรถยนต์มาใช้ ซึ่งแน่นอนว่าในคลาสนี้ต้องมีเมธอดสำหรับหยุดรถอยู่อย่างแน่นอน แต่สำหรับรถยนต์รุ่นใหม่คุณอาจแก้ไขวิธีหยุดรถแบบเดิมเสียใหม่ให้ใช้แบบเอปีเอสแทนก็ได้

□ **Method overloading** จากที่กล่าวถึงการแก้ไขหรือปรับปรุงเมธอดไปในข้างต้น ด้วยวิธีการโอเวอร์โหลดนั้น มีเงื่อนไขอยู่ตรงที่ว่าเมธอดที่แก้ไขใหม่จะต้องมีจำนวนและชนิดอาร์กิวเมนต์เหมือนในคลาสเดิมทุกอย่าง ซึ่งต่างจากโอเวอร์โหลดคั้งที่วิธีนี้ไม่จำเป็นต้องมีจำนวนและชนิดอาร์กิวเมนต์เหมือนเดิม ขอเพียงแต่ใช้ชื่อเมธอดเดียวกันก็พอ เช่น ในเมธอดสำหรับการหยุดรถนั้น คุณอาจแยกเป็นหยุดรถ และหยุดรถขณะเบรกแตกซึ่งทั้งสองวิธีนี้จะมีเงื่อนไขแตกต่างกันแต่หน้าที่เหมือนกันคือหยุดรถให้ได้ ดังนั้นเมธอดที่เขียนก็จะต้องแยกเป็นสองแบบที่ชื่อ `stop_car` เหมือนกัน

4.2.3 จาวาแอปเพล็ต

แอปเพล็ตเป็นโปรแกรมลักษณะหนึ่งเขียนได้ด้วยภาษาจาวา แต่แอปเพล็ตมีลักษณะพิเศษต่างจากโปรแกรมธรรมดาที่คุณจะดูการทำงานของโปรแกรมได้จะต้องสร้างไฟล์ HTML ขึ้นมาก่อน จากนั้นจึงใช้เว็บเบราว์เซอร์ หรือ Applet Viewer ดูแทนส่วนโปรแกรมอีกประเภทหนึ่งคือ แอปพลิเคชันซึ่งโปรแกรมชนิดนี้สามารถทำงานได้โดยลำพัง ไม่ต้องมีโปรแกรมมาช่วย สำหรับแอปพลิเคชันนี้มีความแตกต่างในโครงสร้างของโปรแกรมซึ่งในที่นี้จะขอเน้นเฉพาะการเขียนแอปเพล็ตเท่านั้น

ขั้นตอนในการเขียนจาวาแอปเพล็ตขึ้นมา ขั้นแรกต้องหากซอฟต์แวร์ช่วยเหลือการเขียนภาษาจาวามาก่อน เช่น Java Workshop หรือถ้าไม่มีก็สามารถดาวน์โหลดซอฟต์แวร์ที่ใช้งานได้ฟรีคือ JDK (Java Development Kit) ได้จาก <http://www.javasoft.com/> ก่อนจากนั้นที่ต้องมีอีกคือ เทกซ์เอ็ดเตอร์ในการเขียนโค้ดโปรแกรม ซึ่งในโครงการนี้ได้เลือกใช้ JDK และ Java Visual Café ในการพัฒนา

	NL	NF	AV	JA
read local file	no	no	yes	yes
write local file	no	no	yes	yes
get file information	no	no	yes	yes
delete file	no	no	no	yes
run another program	no	no	yes	yes
read the <i>user.name</i> property	no	yes	yes	yes
connect to network port on server	yes	yes	yes	yes
connect to network port on other host	no	yes	yes	yes
load Java library	no	yes	yes	yes
call <i>exit</i>	no	no	yes	yes
create a pop-up window	warning	yes	yes	yes

ตารางที่ 4.1 แสดงระบบรักษาความปลอดภัยของจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้จัดทำนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 เรียลไทม์ออดิโอ (Real Audio)

ในปัจจุบันเทคโนโลยีการเผยแพร่ข้อมูลข่าวสารผ่านทางเครือข่ายอินเทอร์เน็ตได้ถูกพัฒนาขึ้นอย่างรวดเร็วมาก โดยได้มีการนำเสนอโดยใช้ทั้งภาพและเสียงเพื่อให้การนำเสนอเป็นที่น่าสนใจมากยิ่งขึ้น และโปรแกรมที่สามารถนำมาใช้ในการนำเสนอทั้งภาพและเสียงที่มีคุณภาพเป็นที่ยอมรับของบรรดาผู้ผลิตที่ต้อง การนำเสนอข้อมูลของตนเองก็มีให้เลือกใช้อย่างมากมาย ระบบเรียล (real system) ก็เป็นโปรแกรมหนึ่งที่เป็นที่ยอมรับและถูกนำมาใช้งานอย่างแพร่หลาย

ในปัจจุบันระบบเรียลเป็นที่ยอมรับและถูกนำมาใช้งานกันอย่างแพร่หลายทั้งนี้เนื่องจากเป็นระบบที่สามารถนำเสนอข้อมูลในแบบออกอากาศสดหรือเป็นแบบออน ดีมานด์ ก็ได้ โดยใช้เทคโนโลยีแบบใหม่ที่เรียกว่า สตริม เป็นวิธีการในการส่งข้อมูลเมื่อข้อมูลได้ถูกส่งมาถึงทางฝั่งผู้รับแล้วสามารถที่จะเล่นได้ทันที โดยไม่ต้องให้ส่วนประกอบทั้งหมดของไฟล์มาถึง ซึ่งเป็นการทำงานในแบบเรียล ไทม์ ข้อมูลที่สามารถส่งได้นั้นก็จะรวมทั้งภาพและเสียง คุณลักษณะของระบบเรียล มีดังนี้

1. ลักษณะสำคัญของโปรแกรมคือสามารถที่เล่นกลับ (playback) ข้อมูลแบบสตรีม (Stream) ได้ ชนิดของข้อมูลที่สามารถนำมาเล่นกลับได้เช่น เท็กซ์, อนิเมชัน, เสียง, วีดีโอ, ภาพแบบ 2 มิติ และ 3 มิติ, ไฟล์มีดี (MIDI) และพีริเซนเดชั่น

2. สามารถทำงานได้โดยใช้โปรโตคอลที่ใช้งานอย่างแพร่หลายภายในเครือข่ายอินเทอร์เน็ต และสามารถที่จะทำงานได้ภายใต้ระบบปฏิบัติการได้หลายแบบได้แก่ ภายใต้ระบบวินโดวส์, ภายใต้ระบบแมค, ภายใต้ระบบยูนิกซ์ และสามารถที่จะทำงานได้ภายใต้บราวเซอร์ เครื่องแม่ข่ายของโปรแกรมจะทำงานภายใต้ระบบ วินโดวส์ เอ็นที หรือระบบยูนิกซ์ โดยการส่งข้อมูลระหว่างเครื่องแม่ข่ายและเครื่องที่ทำหน้าที่เป็นตัวแทนจะใช้โปรโตคอลที่ชื่อว่า เรียล ไทม์ สตริมมิ่ง โปรโตคอล (Real Time Streaming Protocol)

3. ภายใต้การทำงานแบบ โคลเอนท์-เซิร์ฟเวอร์ ชนิดของข้อมูลที่จะนำออกเผยแพร่สามารถเป็นชนิดข้อมูลใดๆ ก็ได้ เครื่องแม่ข่ายสามารถที่จะอ่านข้อมูลและกระจายข้อมูลแบบสตรีมได้ และเมื่อข้อมูลได้รับถึงเครื่องลูกข่าย ปลั๊กอินที่ติดตั้งอยู่ที่เครื่องลูกข่ายก็จะเป็นตัวบอกว่าข้อมูลที่ได้รับมานั้นจะอ่านและนำมาเล่นได้อย่างไร จึงสามารถที่จะนำมาใช้ในการออกอากาศสดได้

4. สามารถกระจายข้อมูลได้ทั้งแบบมัลติ คาสต์ และยูนิ คาสต์ โดยใช้โปรโตคอล ทีซีพี/ไอพี (TCP/IP), ยูดีพี/ไอพี (UDP/IP) และสามารถที่จะใช้งานร่วมกับระบบไฟร์ วอลล์ (firewall) ได้ การประยุกต์นำมาใช้งาน

จากคุณลักษณะดังกล่าว จะเห็นว่าระบบเรียลมีความสามารถในการทำงานสูงทีเดียว รูป 4.4 แสดงโปรแกรมเรียลเพลเยอร์



รูปที่ 4.4 โปรแกรมเรียลเพลเยอร์

ต่อไปนี้เป็นตัวอย่างการนำโปรแกรมไปใช้งานที่พบเห็นได้ทั่วไปบนอินเทอร์เน็ตในลักษณะที่พอจะสรุปได้ดังนี้

1. การกระจายข่าว อาจจะเป็นในแบบที่ออกอากาศสดๆ หรือเป็นแบบ ออน ดีมานด์ (on demand) ก็ได้ ก็จะประกอบด้วยทั้งภาพและเสียง
2. การนำเสนอผลิตภัณฑ์ อาจจะเป็นวีดีโอสาริตการใช้นผลิตภัณฑ์ที่มีทั้งภาพและเสียง
3. การไอเกะ
4. การประชุมผ่านทางเครือข่าย (Video conference) อาจจะเป็นการประชุมแบบสดหรือได้มีการบันทึกไว้ก่อนแล้วก็ได้
5. ฟรีเซนต์เดชั่น ทางด้านการศึกษา
6. การฝึกอบรมของบริษัทต่างๆ

4.3.1 การใช้งานระบบเรียล

ระบบเรียลจะประกอบด้วย 2 ส่วน

1. โปรแกรมที่ทำหน้าที่เป็นตัวสร้างไฟล์ในแบบของระบบเรียล ได้แก่ โปรแกรมเรียลพับลิเชอร์ (real publisher) และ โปรแกรมเรียล เอ็น โคคเดอร์ (real encoder)
2. โปรแกรมที่ใช้เล่นได้แก่โปรแกรมเรียลเพลเยอร์ (real player) ที่จะต้องถูกติดตั้งอยู่

จึงจะสามารถเล่นได้ สามารถดาวน์โหลดมาใช้งานได้ที่ <http://www.real.com/> เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 โปรแกรมเรียล พับลิเชอร์ และโปรแกรมเรียล เอ็นโคดเดอร์

ภายใต้การทำงานของโปรแกรมเราสามารถกำหนดได้ว่าต้องการแบนด์วิดท์เท่าไร เพื่อความเร็วในการส่งข้อมูลโดยข้อมูลที่จะนำมาใช้แบ่งได้เป็น 2 แบบ

1. ข้อมูลนั้นถูกจัดเก็บในรูปแบบของไฟล์แล้ว เช่นไฟล์เสียงก็อาจจะเป็น .wav ถ้าเป็นไฟล์ภาพก็อาจจะเป็น .avi หรือ .mov เป็นต้น

2. ข้อมูลยังไม่ได้รับการบันทึกหรือเป็นการได้มาสดๆ นั้นเอง ก็จะเป็นการนำมาออกอากาศสด เช่นจากแหล่งดังต่อไปนี้ ไมโครโฟน, กล้อง, วีดีโอ, สัญญาณจากดาวเทียม เป็นต้น

ไฟล์ที่ได้จากการใช้โปรแกรมจะแบ่งได้ดังนี้

ไฟล์เสียง (real audio file) นามสกุล .ra ถูกสร้างโดยโปรแกรมเรียล เอ็นโคดเดอร์ในเวอร์ชันเก่า

ไฟล์ภาพและเสียง (real media clip) นามสกุล .rm ถูกสร้างโดยโปรแกรมเรียล เอ็นโคดเดอร์ในเวอร์ชัน 4.0 และ 5.0 และจากโปรแกรมเรียล พับลิเชอร์ 5.0 มีได้ทั้งภาพและเสียงหรืออย่างใดอย่างหนึ่งเล่นได้ด้วยโปรแกรมเรียลเพลเยอร์

ไฟล์ภาพและเสียงแบบเมทาไฟล์ (real media metafile) นามสกุล .ram เป็นพอยเตอร์ชี้ไปยังไฟล์ใน 2 รูปแบบข้างต้น โดยภายในไฟล์จะเป็นข้อมูล URL ของไฟล์ที่ถูกชี้

ไฟล์ภาพและเสียงปลั๊กอินแบบเมทาไฟล์ (real media plug-in metafile) นามสกุล .rpm เหมือนกับไฟล์ภาพและเสียงแบบเมทาไฟล์แต่ใช้กับปลั๊กอินของบราวเซอร์

บทที่ 5

การออกแบบ

5.1 ระบบปฏิบัติการและเว็บเซิร์ฟเวอร์ที่ใช้

การออกแบบโฮมเพจนี้สิ่งที่สำคัญที่สุดส่วนหนึ่งในการออกแบบก็คือการเลือกเซิร์ฟเวอร์ และเว็บเซิร์ฟเวอร์ ที่เหมาะสมให้กับโฮมเพจที่จะจัดตั้งขึ้นมา ซึ่งมีให้เลือกอย่างมากมายมีทั้งที่ให้ใช้ฟรี และต้องเสียเงินซื้อมาใช้งาน แต่ละตัวนั้นมีความสามารถมากน้อยแตกต่างกันไป ในที่นี่เราจะเลือกระบบปฏิบัติการ Windows NT 4.00 ที่เลือกก็เพราะว่าเป็นระบบปฏิบัติการที่หามาใช้ได้ง่ายมากที่สุด และมีซอร์ฟแวร์มารองรับมากมาย ติดตั้งง่าย มีฟังก์ชันในการใช้งานครบถ้วนตามที่ต้องการ ถึงแม้ว่าในระบบปฏิบัติการตัวนี้จะไม่ใช่ตัวที่ดีที่สุดก็ตาม แต่เรียกได้ว่าเป็นตัวที่เหมาะสมที่สุดมากกว่า เมื่อได้ตัวระบบปฏิบัติการแล้ว ต่อมาก็จะถึงของเรื่องเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์นั้นคือโปรแกรมที่บริการงานทางด้านที่เกี่ยวกับอินเทอร์เน็ต ซึ่งก็มีให้เลือกหลายตัวอีกเช่นเดียวกันแต่ในที่นี้จะเลือก Internet Information Server ที่มาพร้อมกันกับ Windows NT 4.00 Server (หรือ Peer Web Server ใน Windows 4.00 Workstation) ใน IIS นั้นจะมีบริการ World Wide Web ,Gopher และ FTP พร้อมกันกับ Internet Service Manager (ISM) ซึ่งเป็นเครื่องมือในการจัดการเซิร์ฟเวอร์เหล่านี้ ทั้งจากเครื่องโกลบอลและจากเครื่องรีโมต เวอร์ชันที่ใช้ในขณะนี้คือ IIS 3.0 ใน IIS 3.0 นี้สามารถใช้ Java Virtual Machine (JVM) ได้บนฝั่งเซิร์ฟเวอร์แล้ว เป็นผลที่ทำให้ องค์ประกอบต่างๆ สามารถที่จะรันบน NT Server ได้โดยตรง

IIS 3.0 นี้จะมีระบบการรักษาความปลอดภัยมากกว่าเวอร์ชันก่อน เพราะมีการเพิ่มการสนับสนุนสำหรับ Secure Sockets Layer (SSL) 3.0 ในรูปแบบของการตรวจสอบโคลเอนด์โดยการใช้ใบอนุญาตแบบดิจิทัลซึ่งจะทำให้เซิร์ฟเวอร์ตรวจสอบหลักฐานของโคลเอนด์ที่เข้ามาใช้บริการมัน จะให้ใช้หรือไม่ให้ใช้ก็ได้โดยดูจากหลักฐานทางดิจิทัลที่เว็บเบราว์เซอร์ส่งมาให้

ใน IIS 3.0 นี้ยังมีการทำงานที่รวมมาให้หลายๆ อย่างในตัวเดียวซึ่ง เช่น Netshow 1.0 ซึ่งเป็นเซิร์ฟเวอร์ที่ให้การสนับสนุน สตรีมมิง-มีเดีย ในสภาวะแวดล้อมมัลติคาสต์ IP ดังนั้นสามารถเพิ่มเติมเนื้อหาที่มีเดียเข้าไปในเว็บไซด์ได้ โดยไม่ไปเพิ่มภาระให้แบนด์วิธของเครือข่ายเลย

5.2 การออกแบบส่วนโฮมเพจหลัก

ส่วนโฮมเพจในหน้าหลักนั้นถือว่าเป็นส่วนสำคัญที่ดึงดูดความสนใจของผู้ที่เข้ามาใช้บริการ จึงทำให้การออกแบบนั้นเน้นไปที่น่าตาความสวยงามของโฮมเพจเป็นหลัก แต่ในขณะเดียวกันก็ได้คำนึงถึงความครบถ้วนของข้อมูลด้วย จากการที่ได้ไปวิเคราะห์ข้อมูลของทางสำนักหอสมุดกลาง ซึ่งได้จากการสอบถามเจ้าหน้าที่ในหน่วยงานต่างๆ ของสำนักหอสมุดกลางแล้วนำมาวิเคราะห์จึงพบว่าควรจะนำสิ่งใดใส่ไว้ในโฮมเพจบ้าง ซึ่งสิ่งที่ควรจะนำไปใส่ไว้ในโฮมเพจได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ข่าวจากสำนักหอสมุดกลาง
2. การให้บริการของห้องสมุด
3. ประกาศต่างๆ
4. เกี่ยวกับสำนักหอสมุดกลาง
 - สถิติการใช้งานหอสมุดกลางและหอสมุดคณะต่างๆ
 - ปริมาณหนังสือ ตำรา วารสาร
 - โครงสร้างองค์การและ รายชื่อบุคลากร
 - ความเป็นมาของสำนักหอสมุดกลาง
5. ค้นหาหนังสือ
 - รายชื่อหนังสือ-ตำรา (Innopac)
 - รายชื่อวารสาร (Current content)
 - รายชื่อวิทยานิพนธ์ และบทคัดย่อไทย-อังกฤษ
6. แนะนำทรัพยากรและบริการ
7. หอจดหมายเหตุ
 - เอกสารเผยแพร่ของทางสถาบันฯ
8. หอเทิดพระเกียรติ
 - พระราชประวัติพระบาทสมเด็จพระจอมเกล้าเจ้าอยู่หัวฉบับมัลติมีเดีย
 - พระราชดำรัสพระบาทสมเด็จพระเจ้าอยู่หัวรัชการปัจจุบันและสมเด็จพระเทพฯ เนื่องในพิธีพระราชทานปริญญาบัตรแก่นักศึกษาใหม่
9. โสตทัศนศึกษา
 - รายชื่อวีดีโอเทป คาสเซ็ทเทปและอื่นๆ
10. โปรแกรมภาพยนตร์

ในส่วนเป็นหัวข้อหลักนั้นได้จัดทำไว้ในหน้าหลักของโฮมเพจแรก การได้จัดทำไว้ในหน้าหลักก็เนื่องมาจากจะได้ไม่ให้ผู้ใช้นั้นสับสนในการใช้งาน ซึ่งถ้าออกแบบให้มีหน้าย่อยมากๆ อาจจะทำให้ผู้ใช้นั้นสับสนในการใช้บริการได้ ในบางส่วนนั้นเป็นเพียงโครงการในอนาคตของทางสำนักหอสมุดกลาง ซึ่งในปัจจุบันสำนักหอสมุดกลางยังไม่มีข้อมูลในส่วนนั้นให้ หรือมีแต่ยังไม่สมบูรณ์พร้อมที่จะนำออกเผยแพร่ แต่ได้ทำรูปภาพหัวข้อต่างๆ ไว้รองรับแล้ว เพื่อให้ง่ายต่อการพัฒนาต่อไป ซึ่งจะแสดงโฮมเพจหน้าหลักไว้ดังรูปที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 โฮมเพจหลักของสำนักหอสมุดกลาง

5.3 การออกแบบส่วนแนะนำและบริการต่างๆ

ในส่วนนี้จะใช้ลักษณะการทำงานของ CGI โดยใช้ภาษา Perl มาช่วยในการพัฒนาทั้งหมด โดย Perl ที่ใช้นี้จะเป็น Perl ที่ทำงานบน Windows ซึ่งจะมีการทำงานที่เหมือนกันกับ Perl ที่ใช้บน UNIX ที่รู้จักกันเป็นอย่างดี การพัฒนานั้นจะแบ่งออกเป็นส่วนต่างๆ ได้ดังต่อไปนี้

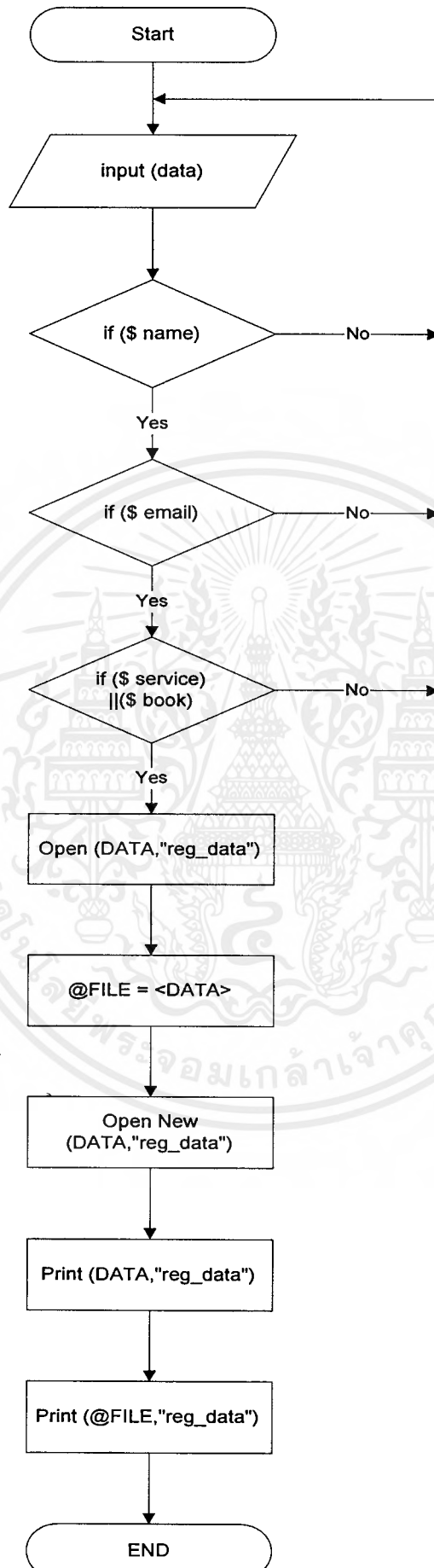
ส่วนรับข้อมูลจากผู้ใช้ แนะนำหนังสือและบริการเป็นส่วนที่ให้ผู้เข้ามาใช้งานโฮมเพจของสำนักหอสมุดกลางได้ทำการพิมพ์สิ่งที่อยากจะแนะนำหนังสือหรือบริการที่ผู้ใช้ต้องการจะได้รับ ซึ่งจะมีช่องสำหรับให้ผู้ใช้ได้ป้อนข้อความเป็นแบบอินพุทบ็อกซ์ โดยจะมีช่องให้ผู้ใช้ใส่ชื่อ, อีเมลแอดเดรส, ที่อยู่โฮมเพจ, แนะนำหนังสือ และแนะนำการบริการ ซึ่งสิ่งที่ผู้ใช้จะต้องป้อนก็คือชื่อ, อีเมลแอดเดรส และแนะนำหนังสือหรือบริการที่ต้องการ เมื่อกดที่ปุ่ม Submit จะไปเรียก CGI ขึ้นมาทำงาน โดยการทำงานของโปรแกรม คือ จะรับอินพุทเข้าที่ผู้ใช้ป้อนผ่านทางไฟล์ HTML เข้ามาเก็บยังตัวแปร แล้วก็ตรวจสอบว่ามีช่องใดที่ป้อนไม่ครบบ้าง ถ้าป้อนไม่ครบก็จะฟ้อง error แล้วให้ทำการป้อนใหม่ เมื่อได้ตัวแปรครบตามที่โปรแกรมต้องการแล้ว สิ่งที่ต้องการคือให้ข้อมูลที่ถูกรับป้อนเข้ามาล่าสุดนั้นให้แสดงผลออกมาก่อน

ส่วนแสดงผล ส่วนนี้เป็นส่วนที่ทำการแสดงผลของข้อความที่ผู้ใช้ได้แสดงความคิดเห็นมายัง สำนักหอสมุดกลาง หลักการทำงานของโปรแกรมคือ นำไฟล์ข้อมูลที่มีอยู่มาแสดงผล โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

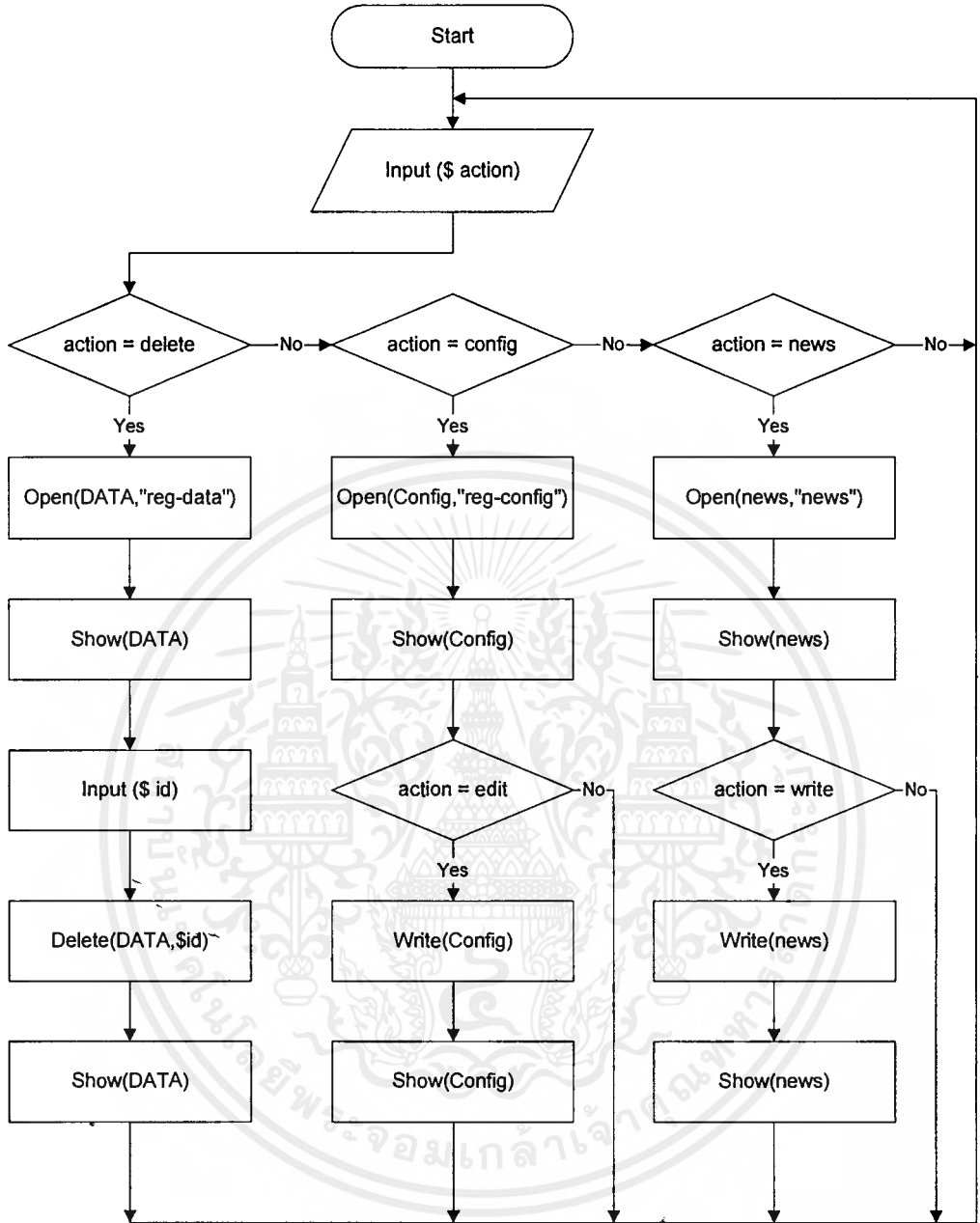
แยกเป็นแต่ละคนที่แสดงความคิดเห็นมา ที่อีเมลแอดเดรสจะทำลิงค์เพื่อส่งอีเมลกลับไปให้คนๆ นั้นได้เลย ในส่วนของโฮมเพจก็เช่นเดียวกันสามารถลิงค์ไปยังโฮมเพจนั้นได้เลย

□ ส่วนผู้ดูแลระบบ ในส่วนนี้จะถูกออกแบบมาเพื่อเจ้าหน้าที่ของทางสำนักหอสมุดกลางโดยเฉพาะ โดยเน้นที่ให้การควบคุมของเจ้าหน้าที่นั้นให้เป็นไปอย่างง่ายที่สุด เจ้าหน้าที่แก้ไขข้อมูลไม่จำเป็นต้องมีความสามารถในเรื่องคอมพิวเตอร์มากนักก็สามารถแก้ไขข้อมูลได้ หลักในการออกแบบคือจะมีไฟล์ทั้งหมดจะมีไฟล์ Config ที่เก็บตัวแปรต่างๆ เก็บไคเรคทอรีของโปรแกรม ไว้เป็นไฟล์ๆ เดียว เมื่อต้องการจะย้ายไฟล์ให้ไปเก็บในไคเรคทอรีอื่นๆ สามารถแก้ไขไฟล์ Config ได้ในเวบบราวเซอร์เลย ทำให้ไม่ต้องไปแก้ไขในตัวโปรแกรมเอง และยังสามารถแก้ไขเปลี่ยนแปลงข้อมูลต่างๆ ของข่าวสารและแนะนำหนังสือบริการได้อีกด้วย การทำงานของโปรแกรม ในขั้นแรกนั้นจะให้ผู้ใช้ได้ป้อนรหัสผ่านเพื่อเข้าไปแก้ไขโปรแกรมก่อน เมื่อเข้าไปได้แล้วนั้นจึงทำการแก้ไขส่วนต่างๆ ของข้อมูลได้ ซึ่งมีอยู่ 3 ส่วนคือ ส่วนแนะนำข่าวสารและบริการ , ส่วน Config , ส่วนข่าวสารต่างๆ ในส่วนของข่าวสารนั้นแต่เดิมที่ต้องใช้ผู้ที่เข้าใจ HTML เป็นคนที่จะแก้ไขข้อมูล ซึ่งมีเจ้าหน้าที่ของสำนักหอสมุดกลางไม่มากนักที่รู้เรื่อง HTML แต่ด้วยโปรแกรมนี้แล้วสามารถทำให้ แก้ไขข่าวสารได้บนเวบบราวเซอร์เลย ผู้ที่แก้ไขจึงไม่จำเป็นต้องรู้เรื่องเกี่ยวกับ HTML เลยก็ได้ หรือไม่ก็สามารถแก้จาก Word Processor ธรรมดาได้เพราะไฟล์ข้อมูลข่าวสารนั้นได้ถูกบันทึกเป็นแบบเท็กซ์ไฟล์

หลักการการทำงานของโปรแกรมทั้งหมดนั้นจะเป็นไปตาม Flowchart ด้านล่าง โดยจะเป็นหลักการทำงานของส่วนรับข้อมูลจากผู้ใช้และส่วนของผู้ดูแลระบบ ในส่วนแสดงผลที่ไม่มี Flowchart ก็เพราะว่าเป็นการเขียน โปรแกรมเพียงแค่อ่านข้อมูลจากไฟล์มาแสดงผลเท่านั้น



รูปที่ 5.2 Flowchart แสดงหลักการทำงาน โปรแกรมส่วนรับข้อมูลจากผู้ใช้



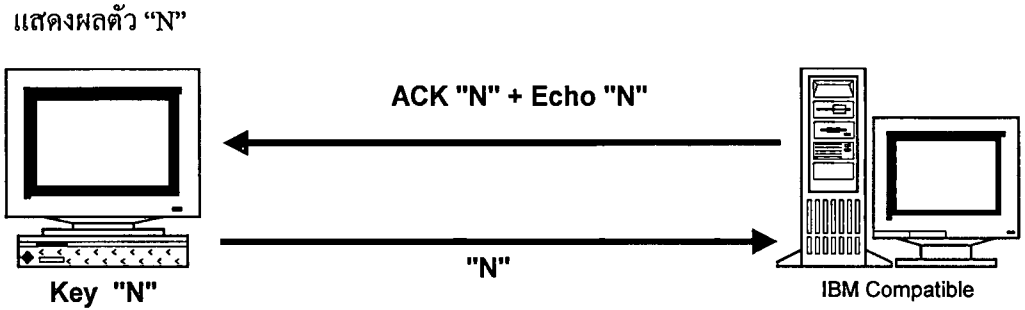
รูปที่ 5.3 Flowchart แสดงหลักการทำงานโปรแกรมส่วนผู้ดูแลระบบ

5.4 การออกแบบส่วนค้นหาหนังสือ

5.4.1 ลักษณะ โปรแกรม Telnet ของห้องสมุด

1. เป็นโปรแกรมที่ใช้การสื่อสารแบบ Character at a time คือ มีการสื่อสารแบบ การส่งตัวอักษรระหว่าง server และ Client ที่ละตัวอักษร และ ตัว Server จะตอบกลับมาว่าได้รับตัวอะไรและทำตามคำสั่งตัวอักษรตัวนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงรูปแบบการสื่อสารของ innopac

แต่การ return output ในขณะที่ Connect เรียบร้อยแล้วจะส่งข้อมูลลงบนตำแหน่งต่างๆ ของหน้าจอ ว่าจะให้แสดง Output ยังไง และตำแหน่งที่เท่าไรบนหน้าจอ โดยมีการส่ง อักขระตำแหน่งที่เท่าไร บนหน้าจอ โดยมีการส่ง อักขระพิเศษมารวมด้วยกับ Data ตามรูปข้างล่าง

```

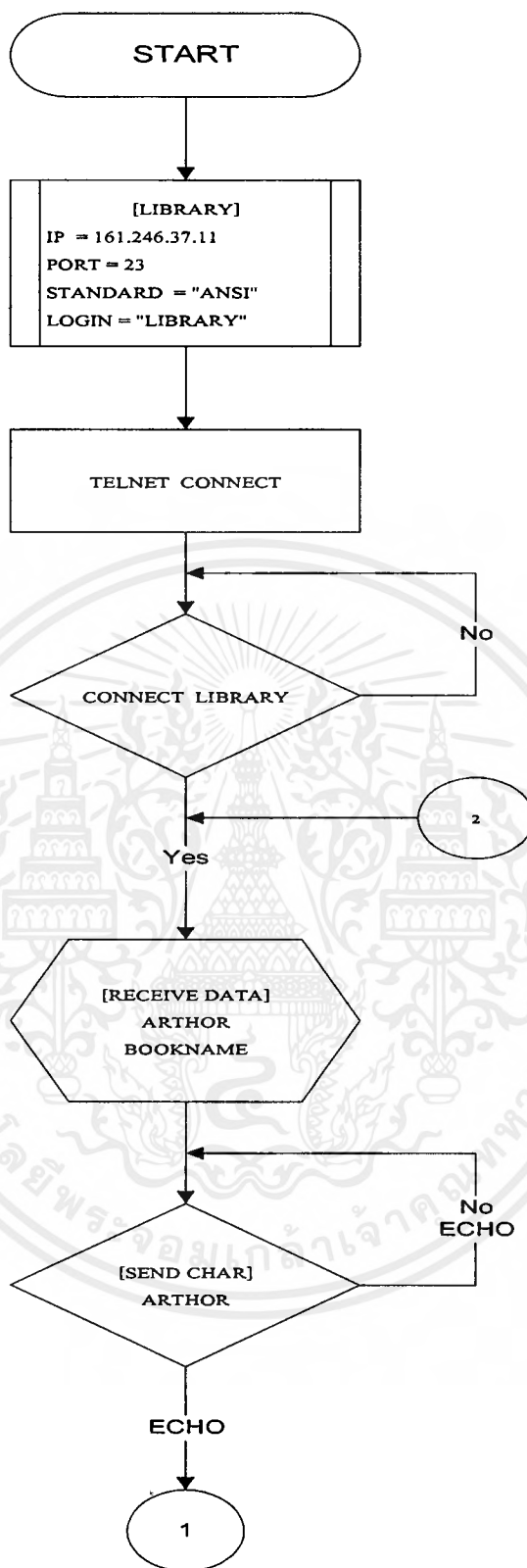
0000: 1b 5b 32 3b 31 48 1b 5b 30 3b 37 6d 0f 20 31 36 | .[2;1H.[0;7m. 16
0010: 20 41 55 54 48 4f 52 53 20 66 6f 75 6e 64 2c 20 | AUTHORS found,
0020: 77 69 74 68 20 31 36 20 65 6e 74 72 69 65 73 3b | with 16 entries;
0030: 20 41 55 54 48 4f 52 53 20 31 2d 38 20 61 72 65 | AUTHORS 1-8 are
0040: 3a 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 | :
0050: 20 20 20 20 20 20 20 20 20 20 20 20 1b 5b 30 3b 31 | .[0;1
0060: 6d 0f 0d 0a 0d 0a 20 20 20 20 20 31 20 20 20 47 | m..... 1 G
0070: 65 6f 72 67 65 2c 20 41 64 72 69 61 6e 20 56 20 | eorge, Adrian V
0080: 20 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e | .....
0090: 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e | .....
00a0: 2e 2e 2e 2e 2e 2e 2e 2e 20 20 20 20 31 20 65 6e 74 | ..... 1 ent
00b0: 72 79 0d 0a 20 20 20 20 20 32 20 20 20 47 65 6f | ry.. 2 Geo
00c0: 72 67 65 2c 20 41 6c 61 6e 20 44 20 20 2e 2e 2e | rge, Alan D ...
00d0: 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e | .....
00e0: 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e | .....
00f0: 2e 2e 2e 2e 2e 2e 20 20 20 20 31 20 65 6e 74 72 79 | ..... 1 entry
0100: 0d 0a 20 20 20 20 20 33 20 20 20 47 65 6f 72 67 | .. 3 Georg
0110: 65 2c 20 41 6c 69 73 74 61 69 72 20 20 2e 2e 2e | e, Alistair ...
0120: 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e | .....
0130: 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e | .....
0140: 2e 2e 2e 20 20 20 20 31 20 65 6e 74 72 79 0d 0a | .... 1 entry..
0150: 20 20 20 20 20 34 20 2b 20 47 65 6f 72 67 65 2c | 4 George,

```

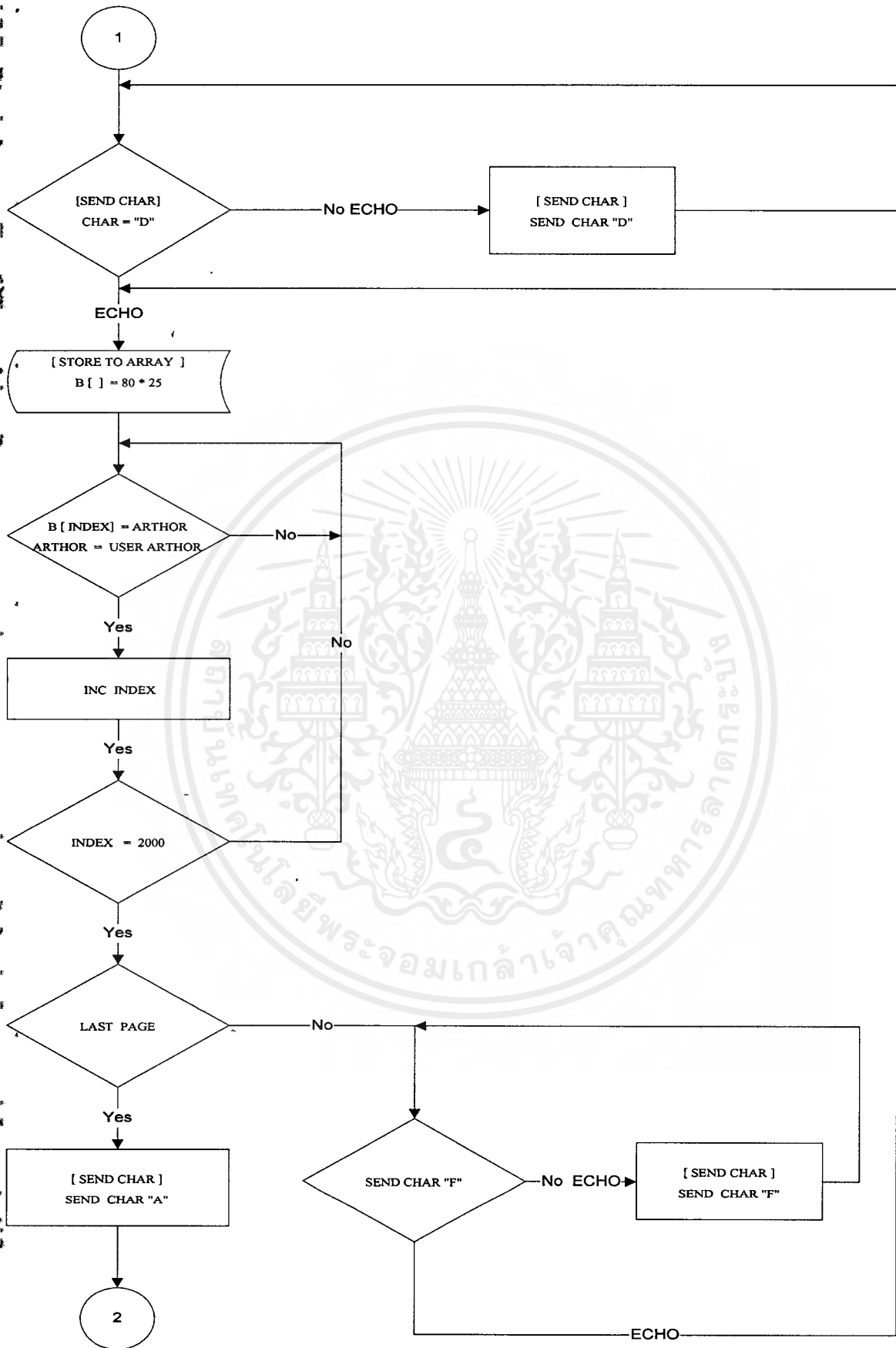
รูปที่ 5.5 แสดง code ของการสื่อสารของ Telnet

ดังนั้นการเขียนโปรแกรมให้ติดต่อกับโปรแกรมนี้จึงต้องมี class ที่ทำหน้าที่ทั้งหมด 4 Class โดยจะทำหน้าที่ต่างๆ กันไปเพื่อรองรับการ connect กับ Server และการเก็บข้อมูลที่ Server ส่งมาบนหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5. 6 แสดง Flow chart ของโปรแกรม Telnet ที่เราทำการเขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 หลักการทำงานของโปรแกรม

โปรแกรม Telnet ที่เขาเขียนขึ้นนี้ประกอบด้วย Class ทั้งหมด 4 Class

1. Class Telnet
2. Class Telnet Client
3. Class Ansi
4. Class emulation

ทำหน้าที่ในการทำงานเป็นลักษณะของ Main() โปรแกรมโดยจะทำหน้าที่ในการ Initial ค่า และการเรียกใช้โปรแกรมต่างๆ ให้ทำงานไปได้ด้วยดี โดยจะทำการกำหนด มาตราฐานในการติดต่อ และ Speed การติดต่อโดยมันจะทำการ Connect เบื้องต้นกับตัว Telnet Library โดยจะเรียกการ Connect ผ่านไปยัง Calss telnet Client โดยรับค่าจาก User ว่าจะใช้ OS แบบไหนและ มาตราฐานการติดต่อแบบ ไหนเช่นใช้ OS->NT

Standard connect -> ANSI

Login -> Library

หลังจากนั้นมันจะ connect ดั่งเองทิ้งไว้

1. Class Telnetclient จะทำหน้าที่ในการติดต่อสื่อสารระหว่าง Server กับ Client ให้สามารถติดต่อไปได้ด้วยดี โดยจะทำการ ส่งตัวอักษรไปยังตัว Server และ ทำการจัดรูปแบบการติดต่อ เมื่อตัวแม่ทำการส่ง echo char กลับมายัง Client ทำการจัดรูปแบบให้เป็นไปตามมาตรฐานของ ANSI โดยตรวจสอบว่าตัวแม่ส่งอะไรบ้าง แล้วตอบกลับไป โดยต้องมีการ Connect ตามลำดับ ก่อนหลังให้ถูกต้องด้วย ส่วนใหญ่การส่งข้อมูลจะเป็นลักษณะของ Packet

บางครั้งตัว telnet-client อาจติดต่อไม่ได้เพราะว่า packet เกิดการสูญหายระหว่างการส่ง



รูปที่ 5.7 แสดง Packet ของ Telnet

ตัว telnet client ก็จะมี Check ว่ามีการตอบกลับจากตัว Server หรือไม่เพราะถ้าไม่มีก็ จะ ส่งไปใหม่ ใน Telnet Client นั้นจะทำการ Check ว่าตัวอักษรบน key board ที่เราต้องการส่งนั้นกับ ตัวอักษรมาตรฐานของ ANSI ตรงกันหรือไม่ถ้าไม่ตรงมันก็จะทำการเปลี่ยนรหัสของตัวอักษร ให้ตรงตามมาตรฐานของ ANSI เช่น กด Enter นั้น Standard ของ Telnet ANSI นั้นจะใช้ “\r” ไม่ใช่ “\n” ตามที่ เราส่งไปถ้าไม่ทำการ Check ก่อนการ Connect จะเกิดการผิดพลาดขึ้นได้โดยโปรแกรมจะต้องตรวจสอบ ทุกตัวอักษรก่อนที่จะส่งไปให้ตัว Server

Class Emulation

เป็น Class ที่ทำหน้าที่เกี่ยวกับการแสดงผลข้อมูลออกทางหน้าจอของ Client ว่าจะแสดงเป็นอย่างไร ใช้ อะไร ขนาดเท่าไรและใช้สีต่างๆ ในการแสดงผล และ Set ค่า Display ว่าเป็น mode อะไรเป็นการใช้ Background เป็นสีพื้น Fourground เป็น สีของตัวอักษร โดยเป็น Base Class ของ การส่งข้อมูลแสดงทำหน้าที่จอ รวมทั้งทำหน้าที่ในการ Scoll Top และ Scoll Down หน้าจอ

Class ANSI

เป็น Sub Class ของ Emulation ที่ทำหน้าที่ในการดูแลอักขระพิเศษต่างๆ ของ Standard ANSI ว่า Escape code แต่ว่าตัวนั้นคืออะไรและ จะตัด Code ก็ไม่เกี่ยวข้องออก รวมทั้งการจัด code พวก Page up – Page Down, Up-arrow , etc ให้ถูกต้องตามมาตรฐานก่อนแล้วค่อยส่งกลับไปให้ตัว Server

เมื่อเริ่ม Run โปรแกรมจะ Connect ไปยัง Host ของ Library ของหอสมุดกลาง แล้วทำการ Auto login เข้าไป แล้วไปรอรับข้อมูลของ User ว่าจะให้ค้นหา อะไรหรือไม่ ถ้ามีการค้นหาก็จะรับข้อมูลเข้ามาโดยจะรับข้อมูลมา 2 ตัวคือ Arthor และ Bock_name แล้วทำการส่งข้อมูล Arthor ไปยัง Host ของหอสมุดกลางว่าต้องการข้อมูลการค้นหาหนังสือตาม Arthor โดยส่ง A แล้วตามด้วยชื่อผู้แต่ง (Arthor) เราจะได้หน้าจอของ Arthor ว่ามีกี่คนและจะแสดงให้เราทราบว่ามี 1 คนแต่งชื่อก็เล่ม หลังจากนั้นเราจะส่งตัว D เพื่อให้ได้รายละเอียดของหนังสือ อีกว่ารายละเอียดของ ชื่อหนังสืออีกว่า มีหน้าอะไรบ้าง ชื่ออะไรบ้างที่ผู้แต่งต้องการ โดยเราจะเก็บข้อมูลแต่ละครั้งคือขนาด 1 หน้าจอ คือ Array ขนาด 2000 byte 80 X 25 80 ตัวอักษร และ 25 บรรทัดต่อหน้า โดยข้อมูลทั้งหมดจะมีทั้ง Data และ อักขระพิเศษ ที่จะบอกว่าพิมพ์ที่ตำแหน่งเท่าไร โดยจะมีอักขระพิเศษ [X,y ว่าอยู่ที่พิกัดเท่าไรแล้วตามด้วยข้อความที่ต้องการพิมพ์ว่ามีอะไรบ้าง เราจะทำงานนำข้อมูลเก็บลง String-result แล้วค่อยไปตัดออกทีหลังว่ามี ชื่อผู้แต่ง (Arthor) ที่ตรงกันมีกี่คนและชื่อหนังสือที่ Arthor คนนั้นแต่งมีกี่เล่ม แล้วนำไปเปรียบเทียบกับ Book_name ว่า ชื่อหนังสือตรงกันหรือไม่ หลังจาก Check ว่า Athor ตรงกันแล้ว ถ้า String ก็เราตัดจาก Array โดยเฉพาะชื่อหนังสือที่ตัดจาก Array 2000 byte ว่าต่างกันหรือไม่ ถ้าตรงกันก็ส่งพิมพ์ออกทาง Output หลังจากนั้นก็ตรวจสอบว่า หน้าที Host ส่งมานั้นเป็นหน้าสุดท้ายหรือไม่ถ้าไม่ก็ส่งตัว Char “F” ต่อไปแล้วก็กลับไปเช็คอย่างเดิมโดยเช็คจาก Array 2000 byte ที่ได้มาใหม่ ถ้าหากว่าเรา Check ว่าหน้าจอที่ส่งมาเป็นหน้าสุดท้ายก็ทำงาน ส่ง Char “A” กลับไปรอรับ ข้อมูลจาก User ว่าต้องการหาแบบอื่นอีกหรือไม่

5.5 การออกแบบการกระจายเสียงผ่านทางเครือข่ายอินเทอร์เน็ต

การให้บริการของห้องสมุดนอกจากหนังสือที่มีอยู่มากมายแล้ว ข้อมูลอื่นๆ ที่น่าสนใจที่ทางห้องสมุดมีอยู่หากมีการนำเสนอเผยแพร่แล้ว ก็จะเป็นประโยชน์แก่ผู้ที่มีความสนใจเป็นอย่างมากเช่น บทเพลงพระราชนิพนธ์ พระราชดำรัสของพระบาทสมเด็จพระเจ้าอยู่หัวรัชกาลปัจจุบันและสมเด็จพระเทพรัตนราชสุดาฯ ที่ทรงพระราชทานแก่บัณฑิตที่เข้ารับพระราชทานปริญญาในพิธีมอบประกาศนียบัตรในปีก่อนๆ และพระราชประวัติพระบาทสมเด็จพระจอมเกล้าเจ้าอยู่หัว บรมมิ่งมุนี ซึ่งล้วนแล้วแต่มีคุณค่าและน่าที่จะนำเสนอเผยแพร่เป็นอย่างยิ่ง จึงคิดว่าน่าจะมีการจัดทำส่วนเหล่านั้นขึ้นมาโดยใช้เทคโนโลยีของระบบเรียลในการจัดทำขึ้นมาซึ่งจะมีให้ผู้ให้บริการได้เลือกดังรูปที่ 5.4



รูปที่ 5.8 หน้าให้ผู้ให้บริการได้เลือกฟังระบบเรียล

บทที่ 6

การทดลองและสรุปผลการทดลอง

6.1 การทดลองส่วนแนะนำหนังสือและบริการ

- ส่วนรับข้อมูลจากผู้ใช้และส่วนแสดงผล

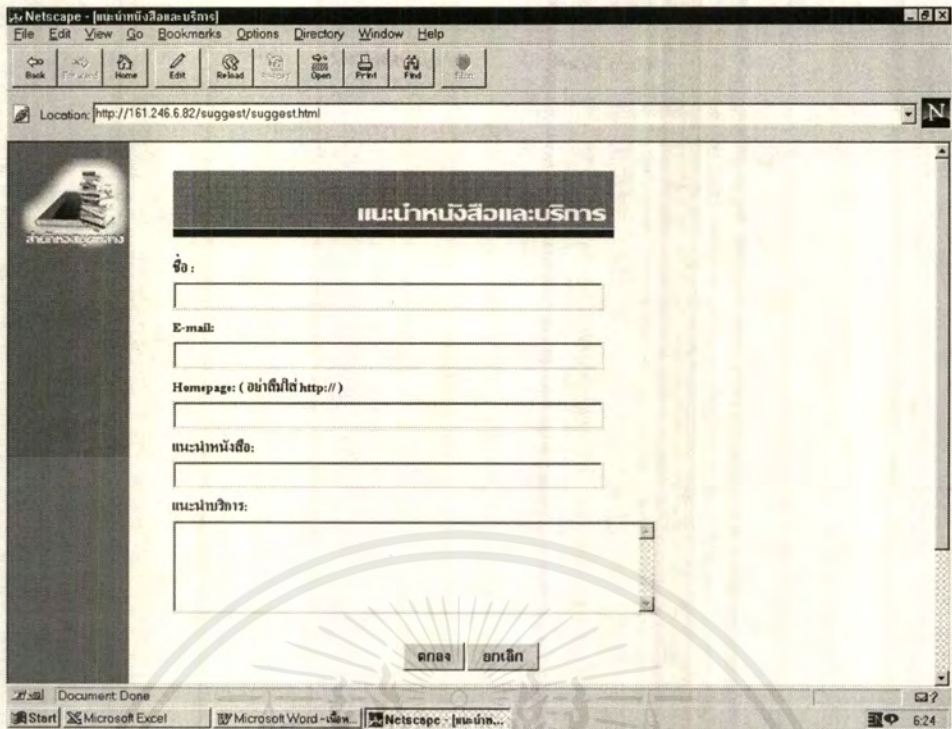
การทดลองในส่วนแนะนำหนังสือและบริการนี้ได้ทำการเขียนโปรแกรมขึ้นมาแล้วทำการทดสอบดูว่า สามารถรับข้อมูลมาเก็บไว้ได้จริงหรือไม่ โดยการทดสอบบนเครื่องเดียวกัน ทำได้โดยใช้ Perl Compile โปรแกรมสามารถอ่านเขียนไฟล์ได้จริงหรือไม่ แล้วไฟล์ที่ถูกเขียนลงไปนั้นมีลักษณะถูกต้องหรือไม่ ตอนแรกนี้ที่ได้ทำการเขียนโปรแกรมจะเกิดปัญหาจากการที่ไม่สามารถแยกค่าแต่ละค่าออกมาแสดงผลได้อย่างถูกต้อง การเก็บข้อมูลในตอนแรกจะเป็นแบบฟิลด์แต่ละเรคคอร์ดนั้นจะใช้ช่องว่างในการแยกค่าแต่ละฟิลด์ออกจากกัน ซึ่งจะแสดงให้เห็นในบรรทัดด้านล่าง

```
4 = maOak s8013273@kmitl.ac.th http://www.kmitl.ac.th Core Java Testpage 6/1/98 18:13
```

ถ้าทำแบบบรรทัดด้านบนแล้วจะเกิดปัญหาที่ว่าเมื่อในฟิลด์แต่ละฟิลด์มีการเว้นวรรคจะทำให้โปรแกรมทำงานไม่ถูกต้อง จึงต้องหาแบบอื่นมาใช้แทน ซึ่งจากการทดลองต่อไปจึงพบว่าถ้าใช้เครื่องหมาย && แทนการเว้นวรรคจะทำให้โปรแกรมทำงานได้ถูกต้อง เพราะคิดว่าไม่น่าจะมีผู้ใดใช้เครื่องหมาย && ถึงสองอัน คิดว่าน่าจะใช้สูงสุดก็แค่อันเดียวในการแทนเครื่องหมายและเท่านั้น จะแสดงให้เห็นในบรรทัดด้านล่าง

```
4 = maOak&&s8013273@kmitl.ac.th&&http://www.kmitl.ac.th&&Java&&Test&&6/1/98&&18:13
```

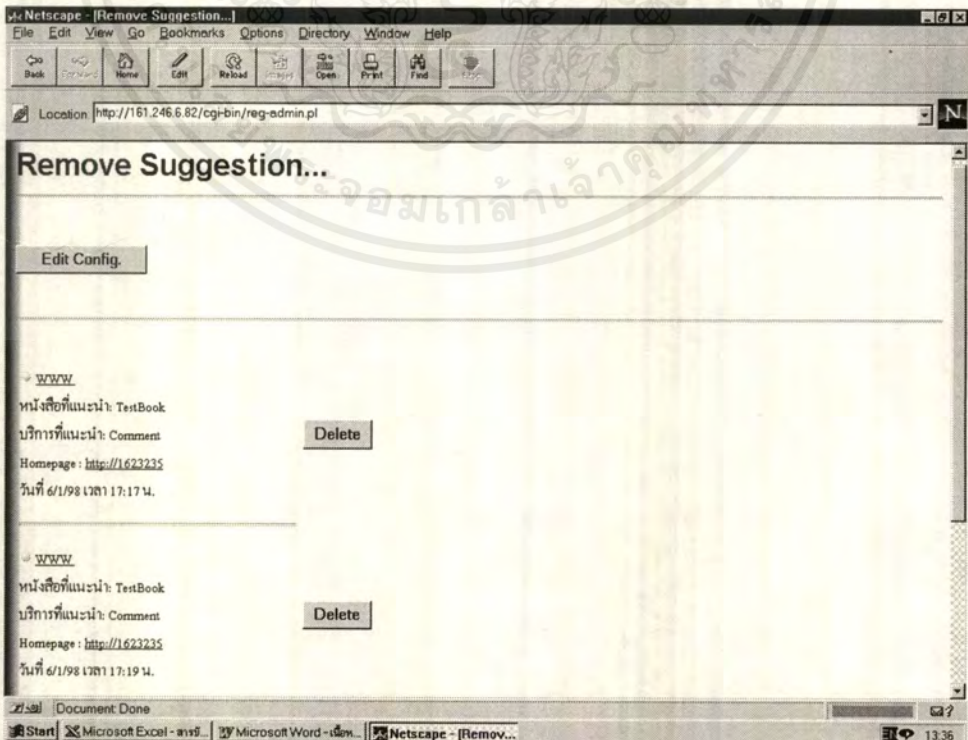
เมื่อปัญหาข้อนี้ผ่านไปจากการทดลองต่อมาจึงได้พบว่าเมื่อทดลองใส่ข้อมูลลงไปแล้วการเรียกข้อมูลออกมาแสดงผลนั้นจะเป็นแบบที่ว่าข้อมูลที่ถูกใส่ไปหลังสุดจะถูกแสดงหลังสุด โดยจะขัดกับหลักความคิดที่จะเรียกค่าแนะนำและบริการ ซึ่งผู้ที่เรียกค่าแนะนำขึ้นมาดูนั้นย่อมอยากจะได้สิ่งที่เพิ่งจะเพิ่มเข้ามาใหม่ๆ มากกว่าที่จะดูของเก่า จึงทำให้ต้องเขียนโปรแกรมขึ้นมาใหม่ โดยให้โปรแกรมนั้นอ่านข้อมูลเก่าเข้ามาเก็บไว้ในตัวแปรก่อน จากนั้นจะเขียนเปิดไฟล์ใหม่ซึ่งไฟล์ใหม่นั้นชื่อเหมือนไฟล์เก่าจะทำให้ข้อมูลนั้นถูกลบหายไปทั้งหมด แต่ได้เก็บข้อมูลเก่าไว้ในตัวแปรก่อนล่วงหน้าแล้ว จากนั้นเมื่อสร้างไฟล์ได้ จะนำข้อมูลใหม่เขียนเข้าไปในไฟล์ก่อน จากนั้นค่อยนำข้อมูลที่อยู่ในตัวแปรเขียนตามลงไป พอเอาไฟล์ข้อมูลนั้นมาแสดงผลจะเห็นได้ว่าข้อมูลใหม่จะอยู่ก่อนข้อมูลเก่า ผลของการทำงานในส่วนรับข้อมูลและแสดงผลนั้นจะแสดงออกมาเป็น HTML ดังรูปที่ 6.1 และ 6.2 ตามลำดับ



รูปที่ 6.1 แสดงโสมเพจส่วนรับข้อมูลจากผู้ใช้

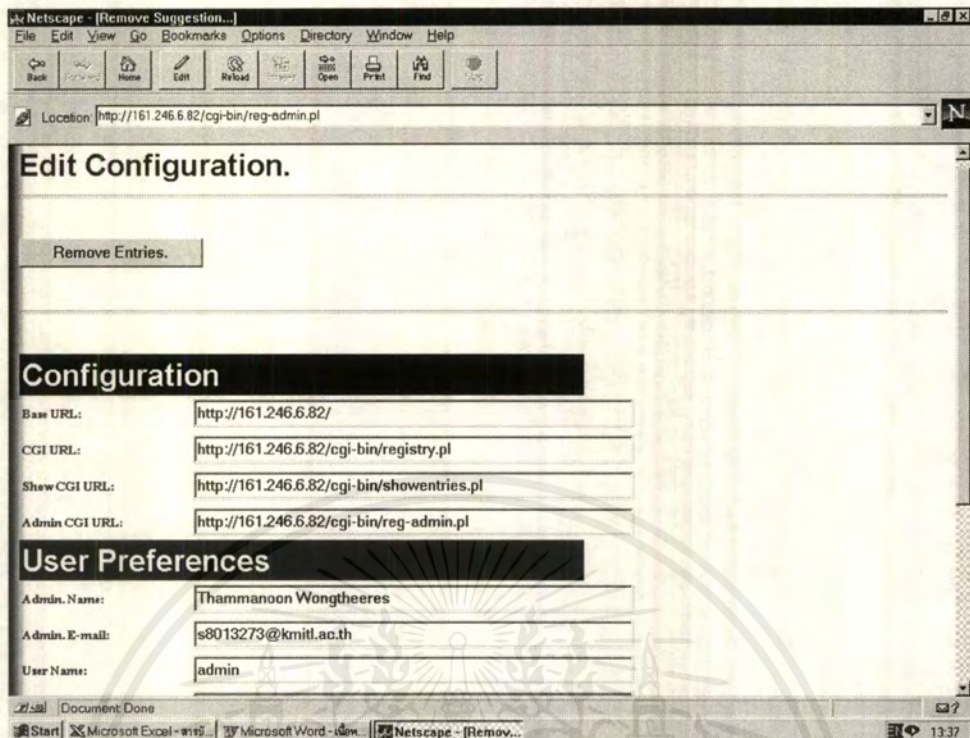
ส่วนผู้ดูแลระบบ

ในการทดลองของส่วนผู้ดูแลระบบนี้จะให้ผู้ดูแลระบบใช้งานโปรแกรมได้ง่าย ซึ่งจะประกอบด้วยกัน 3 ส่วนดังรูปซึ่งจากการทดลองสามารถให้ผู้ใช้ทำการใช้งานโปรแกรมได้ง่ายจริงตามลำดับ



รูปที่ 6.2 แสดงโสมเพจส่วนผู้ดูแลระบบในส่วนลบคำแนะนำหนังสือ

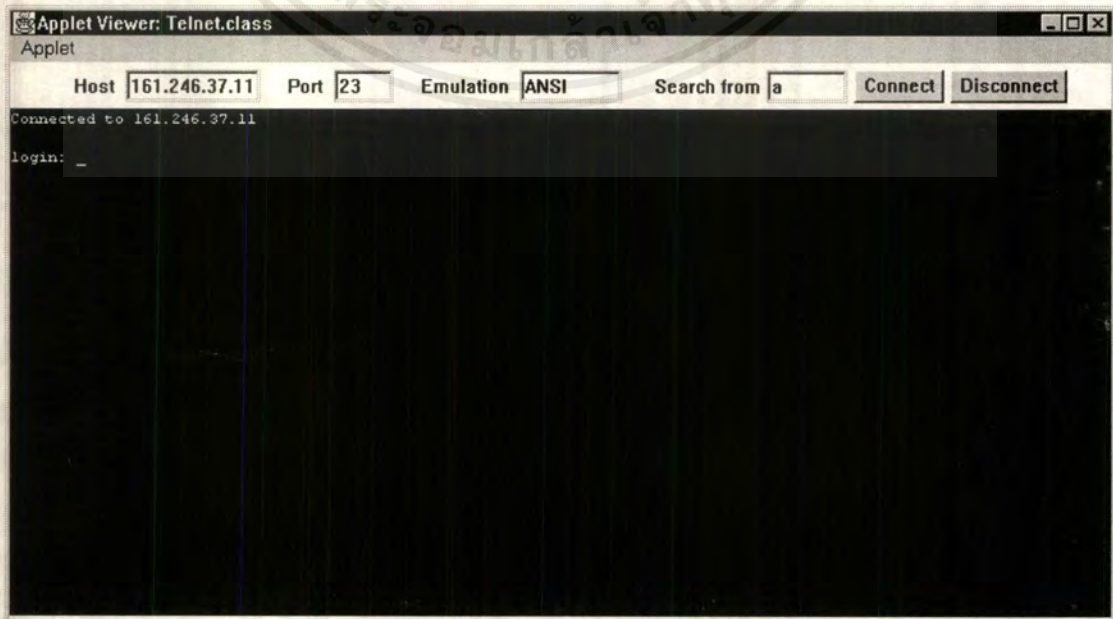
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดงโฮมเพจส่วนผู้ดูแลระบบในส่วนแก้ไขคอนฟิก

6.2 การทดลองส่วนของโปรแกรม Telnet

การทดลองโปรแกรมในส่วนของ Telnet เรานำโปรแกรมที่เราทำการเขียนขึ้นมาทำการ RUN แล้วนำผลที่ได้ว่าสามารถ connect ได้ถูกต้องหรือไม่อย่างไร โดยเรากำหนดเครื่อง client (IP : 161.246.2.154) และ เครื่อง HOST คือเครื่องของหอสมุดกลาง (IP: 161.246.37.11) แล้วสังเกตผลจากการที่เรา ทำการ RUN โปรแกรมได้ดังรูปข้างล่าง



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ โดยที่รูปที่ 6.4 แสดงให้เห็นว่าโปรแกรมเราสามารถ connect ได้แล้ว ให้ประโยชน์ด้านการค้าไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Applet Viewer: Telnet.class

Applet

Host Port Emulation Search from

You searched for the AUTHOR: john FMTTL

169 AUTHORS found, with 237 entries; entries 1-8 are: CALL #

John, Bromham	1 Principles of perspective	NC750 W284p
John, Cathy	2 Painting nature's details in watercolor	NA2237 J625p
John, Fritz,	3 Introduction to calculus and analysis	QA303 C663i
	4 Introduction to calculus and analysis	QA303 C663i
	5 Partial differential equations	QA374 J625p 1991
John, Geraint,	6 Handbook of sports and recreational building d	GV413 H29 1996
	7 Stadia : a design and development guide	GV413 J625s
John, James E.A.,	8 Engineering thermodynamics with heat transfer	TJ265 H23e 1989

Please type the NUMBER of the item you want to see, OR

F > Go FORWARD P > PRINT J > JUMP

M > NEW Search D > DISPLAY AUTHORS only

A > ANOTHER Search by AUTHOR L > LIMIT this Search

Choose one (1-8,F,N,A,P,D,L,J) _

รูปที่ 6.5 หน้าจอหลังจากที่เราทำการ ใส่ข้อมูลรายชื่อผู้แต่งที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปและวิจารณ์

จากการที่ได้พัฒนาโฮมเพจของทางสำนักหอสมุดกลางสถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหารลาดกระบังนี้ ทำให้ได้ข้อสรุปที่ว่าสามารถปรับปรุงงานบริการที่มีอยู่เดิมของสำนักหอสมุดกลางให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งทางเจ้าหน้าที่ของทางสำนักหอสมุดกลางเองก็พึงพอใจเป็นอย่างมากสำหรับงานที่ได้พัฒนาขึ้น และทางคณะผู้จัดทำหวังว่างานที่ได้พัฒนาขึ้นมานี้จะได้นำไปใช้งานจริงตามวัตถุประสงค์ของคณะผู้จัดทำที่ได้ตั้งใจไว้ต่อไป โฮมเพจนี้อาจจะยังไม่ค่อยแพร่หลายนักในขณะที่คณะผู้จัดทำกำลังเขียนปฏิญญาพันธบัตรฉบับนี้ แต่ก็ได้ทำการเผยแพร่ออกไปโดยการขอให้ทางผู้ดูแลโฮมเพจหลักของทางสถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ช่วยในการทำลิงค์จากโฮมเพจหลักของสถาบันมายังโฮมเพจนี้แล้ว และได้ทำการเพิ่มที่อยู่ของโฮมเพจไปยัง Search Engine ต่างๆ เช่น Yahoo, Infoseek, Lycos เป็นต้น จำนวนผู้ใช้โฮมเพจสามารถดูได้จาก Counter ที่ขึ้นอยู่ที่เพจหลักของโฮมเพจ ในส่วนหลักของโฮมเพจนี้ค่อนข้างจะมีความสมบูรณ์แล้ว คงจะเหลือข้อมูลบางส่วนที่ยังไม่มี ในขณะที่ซึ่งคณะผู้จัดทำได้ทำการเตรียมรูปภาพที่จะใช้ให้แล้ว

ปัญหาที่จะเกิดขึ้นในอนาคต คือ เมื่อจะมีการเปลี่ยนแปลงโฮมเพจจำเป็นต้องมีผู้ที่เชี่ยวชาญพอสมควรมาแก้ไข และจะเป็นการลำบากอย่างยิ่งที่จะแก้ไขโปรแกรมที่คณะผู้จัดทำได้เขียนขึ้นมา คณะผู้จัดทำจึงมีความพยายามให้สามารถแก้ไขงานให้เป็นไปได้มากที่สุด เช่น ทำรูปภาพเพื่อรองรับงานต่างๆ ที่คาดว่าจะเกิดขึ้นในอนาคต เพื่อให้รูปเข้ากันกับโฮมเพจหลัก และงานที่เกี่ยวกับ CGI ที่ต้องเซตค่าต่างๆ สามารถแก้ไขผ่านทางเว็บเบราว์เซอร์ได้

แนวทางการพัฒนา

1. พัฒนาเซิร์ฟเวอร์ให้มีความสามารถสูงยิ่งขึ้น แต่เดิมใช้ Pentium 100 MHz Ram 32 MB
2. เพิ่มเต็มข่าวสารที่ยังไม่มีให้ครบถ้วน
3. ปรับปรุงพระราชดำรัสที่มีอยู่ให้มีความทันสมัยอยู่เสมอ
4. พัฒนาโปรแกรมการค้นหาหนังสือ ให้ค้นหาได้รวดเร็วมากยิ่งขึ้น

ภาคผนวก ก

คำสั่งการทำงานต่างๆ ของเทลเนท

NVT Code

NAME	CODE	MEANING
NULL (NUL)	0	No Operation
Line Feed (LF)	10	Moves the printer to the next print line, keeping the same horizontal position.
Carriage Return (CR)	13	Moves the printer to the left margin of the current line.
BELL (BEL)	7	Produces an audible or visible signal (which does NOT move the print head).
Back Space (BS)	8	Moves the print head one character position towards the left margin.
Horizontal Tab (HT)	9	Moves the printer to the next horizontal tab stop . It remains unspecified how either party determines or establishes where such tab stops are located.
Vertical Tab (VT)	11	Moves the printer to the next vertical tab stop. It remains unspecified how either party determines or establishes where such tab stops are located.
Form Feed (FF)	12	Moves the printer to the top of the next page, keeping the same horizontal position.

TELNET commands

NAME	CODE	MEAING
NOP	241	No operation.
Data Mark	242	The data stream portion of a Synch. This should always be accompanied by a TCP Urgent notification.
Break	243	NVT character BRK.
Interrupt Process	244	The function IP.
Abort output	245	The function AO.
Are You There	246	The function AYT.
Erase character	247	The function EC.
Erase Line	248	The function EL.
Go ahead	249	The GA signal.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TELNET CHARSET Option

NAME	CODE	MEAING
SE	240	End of subnegotiation parameters.
SB	250	Indicates that what follows is subnegotiation of the indicated option.
WILL	251	Indicates the desire to begin performing, or confirmation that you are now performing, the indicated option.
WON'T	252	Indicates the refusal to perform, or continue performing, the indicated option.
DO	253	Indicates the request that the other party perform, or confirmation that you are expecting the other party to perform, the Indicated option.
DON'T	254	Indicates the demand that the other party stop performing, or confirmation that you are no longer expecting the other party to perform, the indicated option.
IAC	255	Data Byte 255.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการชิ้นนี้สำเร็จลุล่วงลงได้เพราะความช่วยเหลือจากหลายๆ ฝ่ายที่กรุณาและผู้จัดทำโครงการ ซึ่งโครงการนี้จะสำเร็จไม่ได้ถ้าขาดบุคคลเหล่านี้

อ.ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษาที่ช่วยเหลือทางคณะผู้จัดทำเป็นอย่างมากในทุกๆ เรื่อง
 ดร.เอื้อน ปิ่นเงิน ผู้อำนวยการสำนักหอสมุดกลางและที่ปรึกษาโครงการ
 เจ้าหน้าที่สำนักหอสมุดกลาง ทุกๆ ท่านที่เป็นธุระจัดหาข้อมูลตลอดจนอุปกรรมในการทำงานต่างๆ

และสุดท้ายนี้ ขอขอบคุณเพื่อนๆ ทุกคนที่คอยเป็นกำลังใจให้กับเราคณะผู้จัดทำเสมอมา และต้องขอบคุณผู้ใช้อินเทอร์เน็ตทุกๆ คนที่ได้เข้ามาใช้บริการโฮมเพจของเรา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Bob Breedlove, etal, “Web Programming Unleashed”, Samsnet
2. William E. Weinman, “The CGI Book”, New Reders, 304p., 1996.
3. Larry Wall and Randal L. Schwartz., “Programming Perl”, O'Reilly & Associates, 465p., 1992.
4. Douglas E. Comer., “Internetworking with TCP/IP volume1”, Prentice Hall, 474p., 1991.
5. Ed Anuff., “The Java Sourcebook”, Wiley, 498p., 1996.
6. Peter Van Der Linden, “Just Java”, Prentice Hall, 537p., 1997
7. Gary Cornell, Cay S. Horstmann., “Core Java”, Prentice Hall, 766p., 1997.

