

การพัฒนาโปรแกรมการโอนถ่ายไฟล์ ส่วนเซิร์ฟเวอร์

FTP SERVER PROGRAM DEVELOPMENT



กมลรัตน์ เตชสิริโกศล
ขนิษฐสรณ์ เวียรศิลป์
เยาวลักษณ์ แก้วกิติพงษ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....

เลขทะเบียน 36140

วัน, เดือน, ปี 1.1.ค.ศ. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FTP SERVER PROGRAM DEVELOPMENT



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 1999**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การพัฒนาโปรแกรมการโอนถ่ายไฟล์ ส่วนเซิร์ฟเวอร์
 FTP SERVER PROGRAM DEVELOPMENT



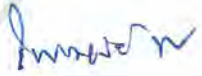
ชื่อนักศึกษา นางสาวกมลรัตน์ เตชศิริโกศล 39054601
 นางสาววนิชฐสรณ์ เวียรศิลป์ 39054608
 นางสาวเยาวลักษณ์ แก้วกิติพงษ์ 39054654

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์ไพบูรณ์ พันธรักษ์พงษ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้รับปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2542

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	อาจารย์สิริลักษณ์ อนันต์สถิตย์สิน	
กรรมการ	อาจารย์นันทิกา เบญจเทพานันท์	
กรรมการและอาจารย์ที่ปรึกษา	อาจารย์ไพบูรณ์ พันธรักษ์พงษ์	



(อาจารย์ไพบูรณ์ พันธรักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การพัฒนาโปรแกรมการโอนถ่ายไฟล์ ส่วนเซิร์ฟเวอร์	
ชื่อนักศึกษา	นางสาวกมลรัตน์ เตชสิริโกศล	39054601
	นางสาวชนิษฐสรณ์ เวียรศิลป์	39054608
	นางสาวเขาวลัษณ์ แก้วกิติพงษ์	39054654
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2542	
อาจารย์ที่ปรึกษา	อาจารย์ไพโรบลย์ พันธรักษ์พงษ์	

บทคัดย่อ

โครงการพิเศษเรื่อง การพัฒนาโปรแกรมการโอนถ่ายไฟล์ ส่วนเซิร์ฟเวอร์ (FTP Server Program Development) นี้ เป็นการศึกษาการเขียนโปรแกรมสำหรับโอนถ่ายไฟล์ระหว่างคอมพิวเตอร์ 2 เครื่อง (Client – Server) ซึ่งจะทำการเขียนโปรแกรมส่วนเซิร์ฟเวอร์ (Server) โดยใช้ภาษาจาวาในการเขียนโปรแกรม

โครงการพิเศษนี้จะเป็นการพัฒนาโปรแกรมการโอนถ่ายไฟล์ส่วนเซิร์ฟเวอร์ ที่สามารถติดต่อกับโปรแกรมโอนถ่ายไฟล์ส่วนไคลเอนต์ได้ โดยสามารถรองรับคำสั่ง FTP (FTP commands) พื้นฐานได้ตามมาตรฐานสากลที่ระบุไว้ใน RFC 959 พร้อมทั้งโครงการพิเศษได้เพิ่มคุณสมบัติเพิ่มเติมดังนี้ รองรับการต่อไฟล์ (Resume) สามารถกำหนดจำนวนผู้ใช้ที่จะเข้ามาใช้บริการ (Maximum number of users) และกำหนดเวลาสิ้นสุดการใช้งานของผู้ใช้เมื่อผู้ใช้ขาดการติดต่อ (Idle timeout)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	FTP Server Program Development	
Students	Miss. Kamonrat Techasirikosol	39054601
	Miss. Kanitsorn Vearasilp	39054608
	Miss. Yaowaluk Kaewkitipong	39054654
Degree	Bachelor's Degree of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Sciences	
Academic Year	1999	
Special Project Advisor	Lecturer Praiboon Pantaragphong	

ABSTRACT

The special project "FTP Server Program Development" is a study of developing program for transferring files between 2 computers (Client & Server). The program developed using jdk 1.2.1 is an application for Server.

This special project will be a FTP Server program which can communicate with Ftp Client program, covering basic FTP commands according to RFC 959. Moreover, we have included special features which are support the resume function, define the maximum number of users and define the idle timeout.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่อง โปรแกรมส่งรับไฟล์ข้อมูลส่วนเซิร์ฟเวอร์ สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำขอขอบพระคุณ อาจารย์ไพโรบลุย์ พันธรักษ์พงษ์ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำ และเป็นที่ยปรึกษาในการแก้ปัญหาต่าง ๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้ และขอขอบพระคุณ อาจารย์วิสันต์ ตั้งวงศ์เจริญ ที่ช่วยให้คำแนะนำ และให้ข้อคิดดี ๆ ในการทำปัญหาพิเศษนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจ และทุนทรัพย์ การทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อน ๆ พี่ ๆ และน้อง ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2543

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ที่มาของปัญหาพิเศษ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของปัญหาพิเศษ.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
1.6 การวางแผนงาน.....	3
บทที่ 2 หลักการที่เกี่ยวข้อง.....	5
2.1 TCP/IP Protocol.....	5
2.1.1 Overview.....	5
2.1.2 Transmission Control Protocol (TCP).....	8
2.1.3 ความสัมพันธ์และการทำงานร่วมกันของโพรโตคอลในแต่ละชั้น.....	9
2.2 FTP (File Transfer Protocol).....	11
2.2.1 FTP Model.....	11
2.2.2 Data Transfer Function.....	12
2.2.3 Data Structure.....	14
2.2.4 Establishing Data Connection.....	14
2.2.5 Data Connection Management.....	15
2.2.6 Transmission Mode.....	15
2.2.7 File Transfer Functions.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3	ซ็อกเก็ต.....	21
2.3.1	ชนิดซ็อกเก็ต.....	21
2.3.2	พอร์ต และแอดเดรสของซ็อกเก็ต.....	22
2.3.3	การทำงานระหว่างซ็อกเก็ต และ Application layer.....	22
2.4	เครื่องมือที่ใช้ในกรโปรแกรม.....	25
2.4.1	ภาษาจาวา.....	25
2.4.2	Class diagram java.....	32
2.4.3	ตารางแสดงรายละเอียด Class diagram java.....	34
บทที่ 3	ตัวอย่าง FTP Server ที่ทำการศึกษา.....	40
3.1	BisonWare FTP Server.....	40
3.2	FTP Dog.....	42
3.3	FTP Serv-U.....	43
3.4	G6 FTP Server.....	45
3.5	ตารางเปรียบเทียบคุณสมบัติของ FTP Server แต่ละค่าย.....	47
บทที่ 4	ขั้นตอนในการพัฒนาโปรแกรม.....	48
4.1	โปรแกรมการทดสอบการติดต่อ.....	48
4.1.1	ขั้นตอนการติดต่อผ่านพอร์ตระหว่างโปรแกรมไคลเอนต์ และ โปรแกรมเซิร์ฟเวอร์.....	48
4.1.2	ขั้นตอน และหน้าต่างจอภาพโปรแกรมที่ทำการรันโปรแกรมติดต่อผ่านพอร์ต.....	49
4.2	การออกแบบโปรแกรม.....	51
4.2.1	ความต้องการขั้นต่ำ.....	51
4.2.2	รายละเอียดโปรแกรม.....	51
4.2.3	ประสิทธิภาพในการใช้งานโปรแกรม.....	51
4.2.4	การออกแบบการกำหนดสิทธิผู้ใช้งานโปรแกรม Together Ftp Server.....	51
4.2.5	โปรแกรมที่ออกแบบสามารถทำงานตามคำสั่งตามตารางที่ 4.1.....	52
4.2.6	อัลกอริทึมการทำงานของโปรแกรม Together Ftp Server.....	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 การติดตั้งและใช้งานโปรแกรม Together Ftp Server	56
5.1 ขั้นตอนการติดตั้งโปรแกรม Together Ftp Server	56
5.2 ภาพแสดงจอภาพโปรแกรม และวิธีการใช้โปรแกรม.....	56
5.2.1 เมนูการทำงานภายใน FTP Server.....	56
5.2.2 เมนู File.....	57
5.2.3 เมนู Edit.....	58
5.2.4 เมนู Setup.....	58
5.3 การทดลองโปรแกรม FTP ไคลเอนต์ ติดต่อกับโปรแกรม Together Ftp Server.....	62
5.3.1 โปรแกรม Cute Ftp.....	62
5.3.2 โปรแกรม Coffee cup Ftp.....	63
5.3.3 โปรแกรม Bullet Ftp.....	64
5.3.4 สรุปผลการทดลองโปรแกรม FTP Client ที่ทำการติดต่อกับโปรแกรม Together Ftp Server	64
บทที่ 6 สรุปผลและข้อเสนอแนะ.....	67
6.1 สรุปผล.....	67
6.2 ข้อเสนอแนะ.....	68
บรรณานุกรม.....	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงคำสั่งประเภท Control connection.....	17
2.2 แสดงคำสั่งประเภท File management command.....	17
2.3 แสดงคำสั่งประเภท Command that set data formats.....	18
2.4 แสดงคำสั่งประเภท Command that support file	18
2.5 แสดงรายละเอียดของออปชั่นทั้งสองออปชั่น.....	27
2.6 แสดงตัวอย่างการติดต่อสื่อสารกันบนเครือข่ายโดยใช้ TCP และ UDP.....	28
2.7 แสดงรายละเอียดของ Class diagram java.....	34
3.1 แสดงตารางเปรียบเทียบคุณสมบัติของ FTP Server แต่ละค่าย.....	47
4.1 แสดงคำสั่งและการทำงานระหว่าง FTP Server และ FTP Client.....	52
5.1 ตารางสรุปผลการทดลอง FTP ส่วนไคลเอนต์ที่ทดลองกับ โปรแกรม Together Ftp Server.....	66

สารบัญรูป

รูปที่	หน้า
2.1 แสดงการแบ่งการทำงานของ Network Protocol.....	6
2.2 แสดงโครงสร้าง TCP/IP ที่สร้างขึ้นใหม่.....	6
2.3 แสดงกฎการสื่อสารในรูปแบบ TCP/IP	7
2.4 แสดง Data Encapsulation.....	8
2.5 แสดง FTP โมเดล.....	11
2.6 แสดงเดต้าบล็อกลูก.....	16
2.7 แสดงโครงสร้างของคลาส.....	32
3.1 แสดงหน้าจอโปรแกรม BisonWare FTP Server.....	40
3.2 แสดงหน้าจอโปรแกรม FTP Dog.....	42
3.3 แสดงหน้าจอโปรแกรม FTP Serv-U.....	43
3.4 แสดงหน้าจอโปรแกรม G6 FTP Server.....	45
4.1 แสดงขั้นตอนการทำงานของโปรแกรมทดลองการติดต่อ.....	48
4.2 แสดงหน้าจอโปรแกรมฝั่งเซิร์ฟเวอร์.....	49
4.3 แสดงจอภาพโปรแกรมฝั่งไคลเอนต์.....	50
4.4 แสดงหน้าจอโปรแกรมฝั่งเซิร์ฟเวอร์หลังจากทำการรันโปรแกรมฝั่งไคลเอนต์.....	50
5.1 แสดงจอภาพโปรแกรม FTP Server.....	57
5.2 แสดงจอภาพเมนูภายในเมนู File.....	57
5.3 แสดงจอภาพเมนูภายในเมนู Edit.....	58
5.4 แสดงจอภาพเมนูภายในเมนู Setup.....	59
5.5 แสดงตัวอย่างการเปิดหน้าต่าง User.....	59
5.6 แสดงจอภาพหน้าต่าง User.....	60
5.7 แสดงตัวอย่างการเปิดหน้าต่าง FTP Server.....	61
5.8 แสดงจอภาพหน้าต่าง FTP Server Setting.....	62
5.9 แสดงจอภาพโปรแกรม Cute Ftp version 2.8 ที่ใช้ทดลอง.....	63
5.10 แสดงจอภาพโปรแกรม Coffeecup Free Ftp ที่ใช้ทดลอง.....	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาของปัญหาพิเศษ

ในปัจจุบันมีการใช้เทคโนโลยีของอินเทอร์เน็ต (Internet) กันอย่างแพร่หลาย ไม่ว่าจะเป็น World Wide Web, IRC, E-mail ฯลฯ ซึ่งมีพื้นฐานอยู่บนความต้องการในการติดต่อสื่อสาร โดยไม่จำกัดระยะทางระหว่างผู้ส่งสื่อ และผู้รับสื่อ บ่อยครั้งที่การติดต่อสื่อสารมีความต้องการที่จะรับส่งข้อมูลที่ถูกเก็บอยู่ในรูปแบบของไฟล์ข้อมูลในคอมพิวเตอร์ ซึ่งโปรแกรมที่จะใช้ในการโอนถ่ายไฟล์นั้นคือ โปรแกรมโอนถ่ายไฟล์ (FTP) โดยจะมีส่วนไคลเอนต์ (Client) ที่ทำหน้าที่ส่งไฟล์ข้อมูลไปยังส่วนเซิร์ฟเวอร์ (Server) หรือรับข้อมูลมาจากส่วนเซิร์ฟเวอร์ ซึ่งส่วนเซิร์ฟเวอร์จะคอยเก็บไฟล์ข้อมูลที่ได้รับมานั้นไว้ในที่เก็บข้อมูลของส่วนเซิร์ฟเวอร์ และนำไฟล์ข้อมูลที่ถูกเรียก ส่งไปยังส่วนไคลเอนต์

จากความสำคัญของการโอนถ่ายไฟล์นี้ เราจึงมีความเห็นว่า การศึกษาในเรื่องของโปรแกรมโอนถ่ายไฟล์มีประโยชน์อย่างมากที่จะทำให้เกิดความเข้าใจพื้นฐาน เพื่อนำมาประยุกต์ และนำไปพัฒนาต่อไปได้ในอนาคต ทางผู้จัดทำจึงได้ทำการศึกษาเกี่ยวกับโปรแกรมโอนถ่ายไฟล์ (File Transfer Protocol) และคิดทำโปรแกรมโอนถ่ายไฟล์ ที่ทำหน้าที่เป็นเซิร์ฟเวอร์

1.2 วัตถุประสงค์

1. ศึกษาพื้นฐานของการติดต่อสื่อสารระหว่างคอมพิวเตอร์ โดยผ่านทางโปรโตคอล (Protocol) TCP / IP
2. ศึกษา และทำความเข้าใจการทำงานของ File Transfer Protocol โดยละเอียด
3. เพื่อศึกษาการทำงานของโปรแกรมโอนถ่ายไฟล์ ที่ทำหน้าที่เป็นเซิร์ฟเวอร์บริการ
4. เพื่อให้สามารถสร้างโปรแกรมโอนถ่ายไฟล์ฝั่งเซิร์ฟเวอร์ ที่มีฟังก์ชันมาตรฐานใช้ในการติดต่อกับโปรแกรมโอนถ่ายไฟล์ฝั่งไคลเอนต์

1.3 ขอบเขตของปัญหาพิเศษ

1. โปรแกรมจะถูกสร้างเพื่อใช้งานบนวินโดวส์ 95, 98, NT ซึ่งเป็นระบบปฏิบัติการที่มีใช้กันอย่างแพร่หลายในปัจจุบันนี้
2. สามารถใช้ในการรับส่งไฟล์จากเครื่องเซิร์ฟเวอร์ ไปยังเครื่องไคลเอนต์ได้
3. โปรแกรมสามารถรองรับการต่อไฟล์จากฝั่งไคลเอนต์ได้
4. โปรแกรมมีฟังก์ชันการกำหนดเวลาสิ้นสุดการใช้งานของผู้ใช้เมื่อผู้ใช้ขาดการติดต่อ (Idle timeout)
5. สามารถกำหนดจำนวนผู้ใช้ที่เข้ามาใช้บริการในขณะหนึ่ง ๆ ได้ (Maximum number of users)
6. สามารถกำหนดสิทธิการใช้งานของผู้ใช้ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำโปรแกรมไปใช้ได้
2. ได้ความรู้เกี่ยวกับโปรแกรมโอนถ่ายไฟล์ และสามารถนำไปประยุกต์ใช้ในอนาคค

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาการทำงาน Client/Server
2. ศึกษาโปรโตคอล TCP/IP
3. ศึกษาโปรโตคอล FTP
4. ศึกษาการใช้งานภาษาจาวา
5. ทดลองเขียนโปรแกรมติดต่อผ่านพอร์ต
6. ทดลองเขียนโปรแกรมส่งไฟล์
7. ค้นหาซอฟต์แวร์ FTP Server มาทดลองใช้
8. กำหนดขอบเขตของงานปัญหาพิเศษ
9. สรุปเนื้อหาทั้งหมดเพื่อทำเอกสาร
10. ทำเอกสาร และเอกสารการนำเสนอ (เทอม 1)
11. ศึกษาขอบเขตการทำงานของปัญหาพิเศษ และวิเคราะห์ข้อมูลเพื่อวางแผน
12. ออกแบบระบบงาน
13. สร้างโปรแกรมตามที่ได้ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. ทดสอบโปรแกรมที่สร้างขึ้น

15. สรุปผลและข้อเสนอแนะ

1.6 การวางแผนงาน

1 กค. - 15 ตค.	ศึกษา และรวบรวมข้อมูล
16 ตค. - 25 กย.	ทดลองเขียนโปรแกรม
26 กย. - 30 กย.	ค้นหาซอฟต์แวร์ FTP Server มาทดลองใช้
1 ตค. - 12 ตค.	กำหนดขอบเขตการทำงานปัญหาพิเศษ และสรุปเนื้อหาทั้งหมดเพื่อทำเอกสาร
13 ตค. - 19 ตค.	จัดทำเอกสารประกอบโครงการงานปัญหาพิเศษ
20 ตค. - 25 ตค.	จัดทำเอกสารประกอบการนำเสนอ
1 พย. - 30 พย.	ศึกษาขอบเขตการทำงานของปัญหาพิเศษ และวิเคราะห์ข้อมูลเพื่อวางแผน
1 ธค. - 20 ธค.	ออกแบบโปรแกรม
21 ธค. - 20 กพ.	สร้างโปรแกรมตามที่ได้ออกแบบ
21 กพ. - 15 มีค.	ทดสอบ และแก้ไขโปรแกรมที่ได้สร้างขึ้น
16 มีค. - 20 มีค.	สรุปโครงการงานปัญหาพิเศษ และข้อเสนอแนะ
21 มีค. - 29 มีค.	จัดทำเอกสารประกอบโครงการงานปัญหาพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Task Name	7/99			8/99			9/99			10/99			11/99			12/99			1/00			2/00			3/00		
	1/7	11/7	21/7	1/8	11/8	21/8	1/9	11/9	21/9	1/10	11/10	21/10	1/11	11/11	21/11	1/12	11/12	21/12	1/1	11/1	21/1	1/2	11/2	21/2	1/3	11/3	21/3
ศึกษาและรวบรวมข้อมูล																											
ทดลองเขียน โปรแกรม																											
ค้นหาซอฟต์แวร์ Ftp Server มาทดลองใช้																											
ศึกษาและรวบรวมข้อมูล																											
จัดทำเอกสารประกอบโครงการนำพิเศษ																											
จัดทำเอกสารประกอบกรรณำเสนอ																											
ศึกษาขอบเขตการทำงานและวิเคราะห์																											
ออกแบบ โปรแกรม																											
สร้าง โปรแกรมตามที่ได้ออกแบบ																											
ทดสอบและแก้ไขโปรแกรมที่สร้างขึ้น																											
สรุปโครงการนำพิเศษและข้อเสนอแนะ																											
จัดทำเอกสารประกอบโครงการนำพิเศษ																											

บทที่ 2

หลักการที่เกี่ยวข้อง

2.1 โพรโทคอล TCP/IP

2.1.1 หลักการเบื้องต้น

TCP/IP เป็นโพรโทคอลที่ใช้สื่อสารกันในระบบอินเทอร์เน็ต ซึ่งโพรโทคอลนี้สามารถแบ่งได้เป็น 2 กลุ่มใหญ่ ได้แก่ Transmission Control Protocol (TCP) และ Internet Protocol (IP) โดย TCP/IP เกิดจากการคิดค้นของกระทรวงกลาโหมของประเทศสหรัฐอเมริกา ที่ต้องการเชื่อมต่อเครื่องคอมพิวเตอร์ที่มีอยู่มากมายหลายยี่ห้อเข้าด้วยกันเป็นระบบเครือข่ายเดียว ปัจจุบันระบบสื่อสารข้อมูลโดยใช้ชุดโพรโทคอล TCP/IP เป็นที่ยอมรับแก่คนทั่วไป มีการใช้กันอย่างแพร่หลายจนกลายเป็นมาตรฐานสำหรับการติดต่อสื่อสารที่มีอยู่ในเครื่องคอมพิวเตอร์ทุกชนิด ทุกระบบ และทุกขนาด

หลักการดำเนินงานทั้งหมดของโพรโทคอล จะประกอบไปด้วยหลาย ๆ ชั้น (Layer) ซึ่งนำมาวางซ้อนทับกันได้ออกมาในรูปที่เรียกว่า Protocol Stack แต่ละชั้นจะมีหน้าที่การทำงานที่ชัดเจน และไม่เกี่ยวข้องกัน โดยแต่ละชั้นจะรู้เพียงวิธีการส่งข้อมูลไปยังชั้นอื่น ๆ แต่ไม่รู้ถึงการทำงานข้างใน มีมาตรฐานที่เป็นที่ยอมรับกันทั่วไป เรียกว่า Open System Interconnection (OSI) Reference Model ซึ่งทำการแบ่งการทำงานของ Network Protocol ออกเป็น 7 ชั้น ดังรูป 2.1

Application Layer
Consists of application programs that use the network
Presentation Layer
Standardizes data presentation to the applications
Session Layer
Manages sessions between application
Transport Layer
Provide end-to-end error detection and correction
Network Layer
Manages connections across the network for the upper layers
Data Link Layer
Provides reliable data delivery across the physical link
Physical Layer
Defines the physical characteristics of the network media

รูปที่ 2.1 แสดงการแบ่งการทำงานของ Network Protocol

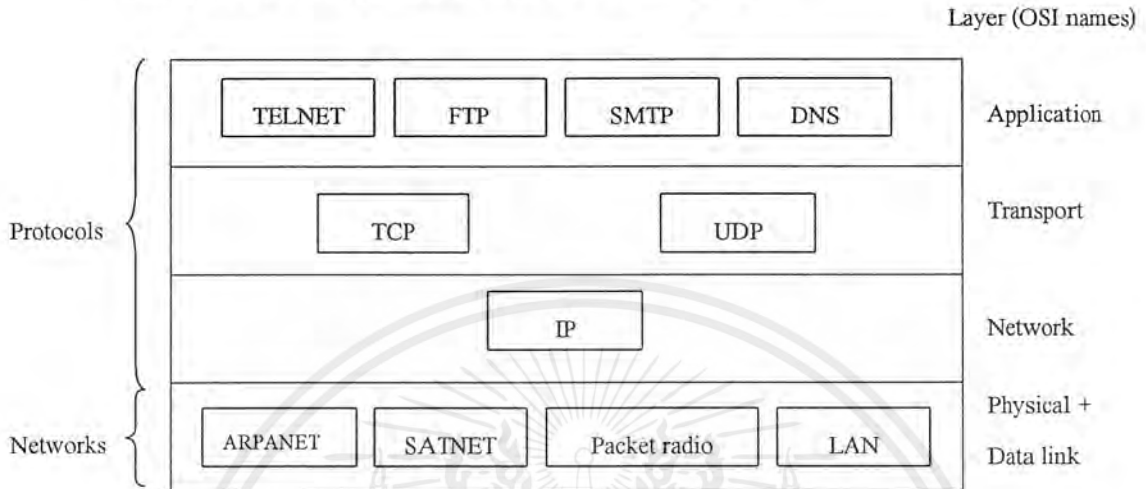
แต่สำหรับการศึกษา TCP/IP จะมีการสร้างโครงสร้างขึ้นมาใหม่โดยแบ่งเป็น 4 ชั้น ดังรูป 2.2

Application Layer
Consists of application and processes that use the network
Transport Layer
Provides end-to-end data delivery services
Internet Layer
Defines the datagram and handles the routing of data
Network Access Layer
Consists of routines for accessing physical network

รูปที่ 2.2 แสดงโครงสร้าง TCP/IP ที่สร้างขึ้นใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

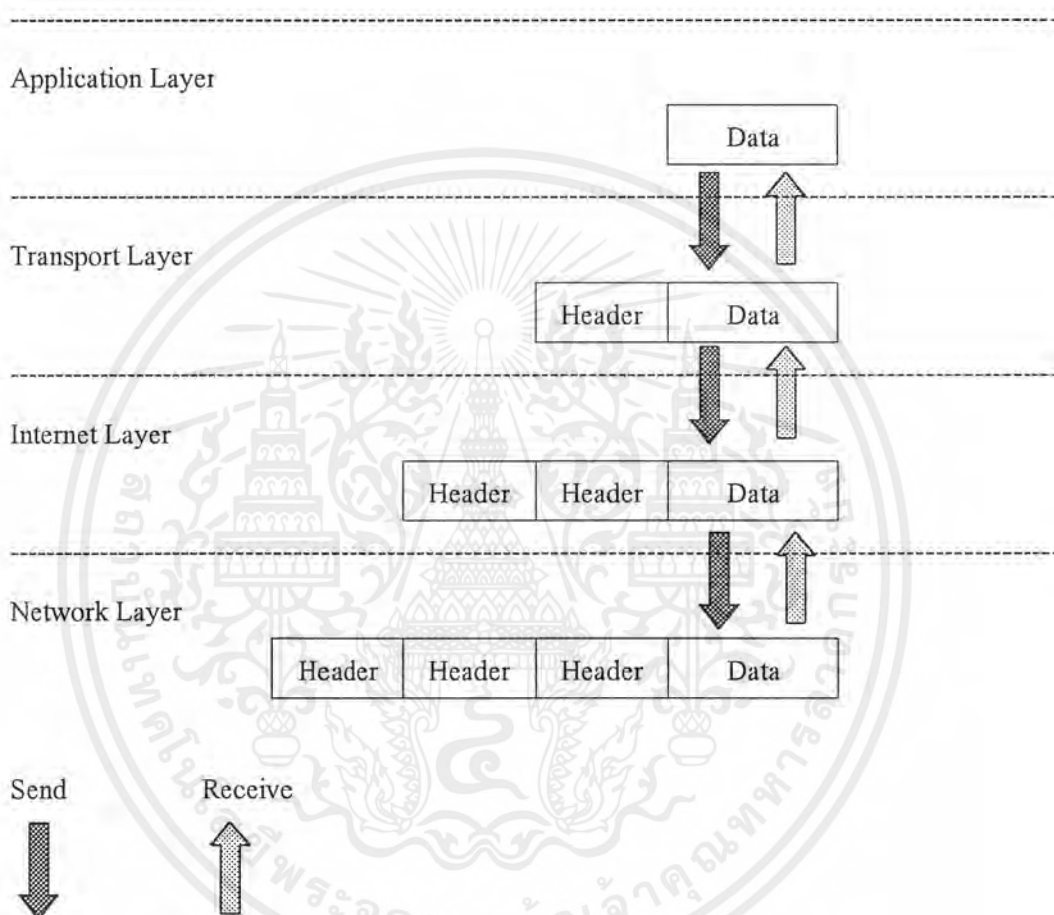
โดยกฎการสื่อสารในรูปแบบของ TCP/IP แสดงดังรูป 2.3



รูปที่ 2.3 แสดงกฎการสื่อสารในรูปแบบ TCP/IP

ลักษณะการทำงานคือ ข้อมูลจะถูกส่งลงมาจากชั้นบนลงข้างล่าง ขณะที่ข้อมูลถูกส่งผ่านในแต่ละชั้นจะทำการเพิ่มข้อมูลควบคุมเข้าไปเพื่อให้การส่งข้อมูลถูกต้อง และเป็นการส่งพารามิเตอร์ที่จำเป็นไปให้กับชั้นในเครื่องปลายทาง ข้อมูลเหล่านี้เรียกว่า เฮดเดอร์ (Header) ซึ่งแต่ละชั้นจะมีเฮดเดอร์ที่เป็นรูปแบบของตัวเอง การเพิ่มเฮดเดอร์เข้าไปพร้อมกับข้อมูลเรียกว่า Data Encapsulation โดยจะกระทำเช่นนี้ไปเรื่อย ๆ จนกระทั่งส่งออกไปยังสายสื่อสาร เมื่อเครื่องปลายทางได้รับเฟรมข้อมูลก็จะถอดรหัสส่วนหัวออก แล้วนำข้อมูลให้กับชั้นบนต่อไป เป็นเช่นนี้ไปเรื่อย ๆ จนถึง Application Layer

Data Encapsulation แสดงดังรูป 2.4



รูปที่ 2.4 แสดง Data Encapsulation

2.1.2 Transmission Control Protocol (TCP)

เป็นโพรโทคอลแบบ Connection-Oriented และมีการตรวจสอบความถูกต้องของข้อมูลที่ได้รับด้วย ทำให้การส่งเป็นไปอย่างน่าเชื่อถือ การส่งข้อมูลของ TCP จะถูกส่งที่เซกเมนต์ (Segment) โดยเครื่องปลายทางจะส่งสัญญาณตอบรับกลับมายังเครื่องต้นทางสำหรับทุก ๆ เซกเมนต์ที่มันได้รับ และตรวจสอบแล้วไม่พบข้อผิดพลาด ถ้าหากว่าเครื่องต้นทางไม่ได้รับสัญญาณตอบรับกลับมา ก็จะ

สันนิษฐานว่าเซกเมนต์เกิดปัญหา และทำการส่งเซกเมนต์นั้นไปอีกครั้ง วิธีการนี้เราเรียกว่า Positive Acknowledgement with Re-transmission (PAR)

การที่บอกว่า TCP เป็นการส่งข้อมูลแบบ Connection-Oriented นั้น แสดงว่ามันจะต้องมีการส่งสัญญาณ Handshake ระหว่างเครื่องต้นทาง และเครื่องปลายทาง เพื่อให้แน่ใจว่าสามารถติดต่อถึงกันได้ แล้วเริ่มส่งข้อมูล วิธีการ Handshake ที่ TCP ใช้ นั้น เราเรียกว่า Three-Way Handshake โดยเมื่อเครื่องต้นทางต้องการส่งข้อมูล มันจะส่งสัญญาณ Synchronize (SYN) ไปยังเครื่องปลายทาง เมื่อเครื่องปลายทางได้รับสัญญาณ SYN แล้ว มันก็จะส่งสัญญาณ Acknowledgement (ACK) กลับมา เมื่อสัญญาณ ACK วิ่งกลับไปยังเครื่องต้นทาง ก็ถือว่าเส้นทางเชื่อมต่อได้เกิดขึ้นแล้ว เครื่องต้นทางจะส่ง ACK กลับไปให้เครื่องปลายทางพร้อมกับเริ่มต้นการส่ง

2.1.3 ความสัมพันธ์ และการทำงานร่วมกันของโพรโทคอลในแต่ละชั้น

จากลำดับชั้นของชุดโพรโทคอล TCP/IP และแอปพลิเคชัน จะเห็นได้ว่าแอปพลิเคชันแต่ละประเภทใน Application Layer เลือกใช้โพรโทคอลในชั้น Transport Layer แตกต่างกันไป เช่น SMTP เลือกใช้ TCP ในขณะที่ BOOTP เลือกใช้ UDP แต่สำหรับ TCP และ IP จะเลือกใช้โพรโทคอล IP ใน Internet Layer และจากนั้นในชั้น Network Layer ก็จะขึ้นกับว่าผู้ใช้งานเลือกใช้เครือข่ายประเภทใดในการส่งผ่านชุดโพรโทคอล TCP/IP

ในการส่งข้อมูลของแอปพลิเคชันต่าง ๆ ที่ทำงานอยู่ไปยังเครื่องปลายทางนั้นอาศัย TCP หรือ UDP ในการส่งข้อมูลไปให้ IP เพื่อทำการส่งไป นั่นคือข้อมูลของทุก ๆ โปรแกรมจะต้องส่งผ่าน IP ทั้งนี้ เราเรียกลักษณะเช่นนี้ว่าการ Multiplexing ส่วนกระบวนการย้อนกลับนั้น เราเรียกว่า Demultiplexing คือ ข้อมูลที่รับเข้ามาจะเข้าไปยัง IP แล้ว IP ก็จะแจกจ่ายข้อมูลเหล่านั้นขึ้นไปยังแอปพลิเคชันที่เป็นเจ้าของให้ถูกต้อง

ข้อมูลที่มาถึงจะต้องผ่านเลเยอร์ชั้นล่างขึ้นไปถึงชั้นบนใน Internet Layer นั้น Internet Protocol จะใช้หมายเลขโพรโทคอล ในการส่งข้อมูลขึ้นไปยัง Transport Layer และในทำนองเดียวกัน Transport Layer ก็จะใช้หมายเลขพอร์ตในการส่งข้อมูลขึ้นไปยัง Application Layer จากหลักการใช้ IP address, โพรโทคอล และหมายเลขพอร์ต ข้างต้นจะเห็นได้ว่าเราต้องการให้ทุก ๆ คาด้านแกรมที่ส่งไปมาในอินเทอร์เน็ตเนตนั้นมีปลายทางที่ไม่ซ้ำกัน (Unique) แต่ละคาด้านแกรมจะมีปลายทางซึ่งไปที่แอปพลิเคชันที่แน่นอนในเครื่องปลายทาง แต่วิธีการข้างต้นนี้ยังไม่เพียงพอเนื่องจากเครื่องคอมพิวเตอร์เครื่องหนึ่งสามารถที่จะรันโปรแกรมเดียวกันพร้อม ๆ กันได้ เช่น ในเครื่อง UNIX เซิร์ฟเวอร์ตัวหนึ่งอาจมีผู้ใช้หลายคนกำลังใช้โปรแกรมคุยอยู่ ในกรณีอย่างนี้เราจะทราบได้อย่างไรว่า คาด้านแกรมที่เข้ามาเป็นของ

โปรแกรมตัวใด เพราะหมายเลข IP Address, โพรโตคอล และพอร์ตเหมือนกันหมด ปัญหานี้สามารถแก้ไขได้โดยใช้วิธีที่เรียกว่า Dynamically Allocated-Port คือ ระบบจะไม่กำหนดเลขพอร์ตไว้ตายตัว แต่จะทำการกำหนดเลขพอร์ตให้กับโพรเซส ในขณะที่มันเริ่มต้นทำงาน ซึ่งเลขนี้จะต้องไม่ซ้ำกับเลขพอร์ตของโพรเซสอื่น ตัวอย่างเช่นถ้าเครื่องปลายทางเครื่องหนึ่งมีผู้ใช้ A และ B กำลังใช้โปรแกรม Telnet อยู่ เครื่องต้นทาง (ซึ่งไม่จำเป็นจะต้องเป็นเครื่องเดียวกัน) จะกำหนดหมายเลขพอร์ตต้นทาง ให้เป็นค่าสุ่มค่าหนึ่ง และเลขพอร์ตปลายทางให้เป็นหมายเลขพอร์ตมาตรฐาน ในตัวอย่างนี้ Telnet จะเท่ากับ 23 ด้วยวิธีการนี้ คาด้าแกรมของ A และ B จะมีเลขคู่ที่แตกต่างกัน เช่น ของ A อาจเป็น 3044,23 ส่วน B อาจจะเป็น 1027,23 ทำให้คาด้าแกรมสามารถเดินทางไปยังแอปพลิเคชันที่ถูกต้องได้



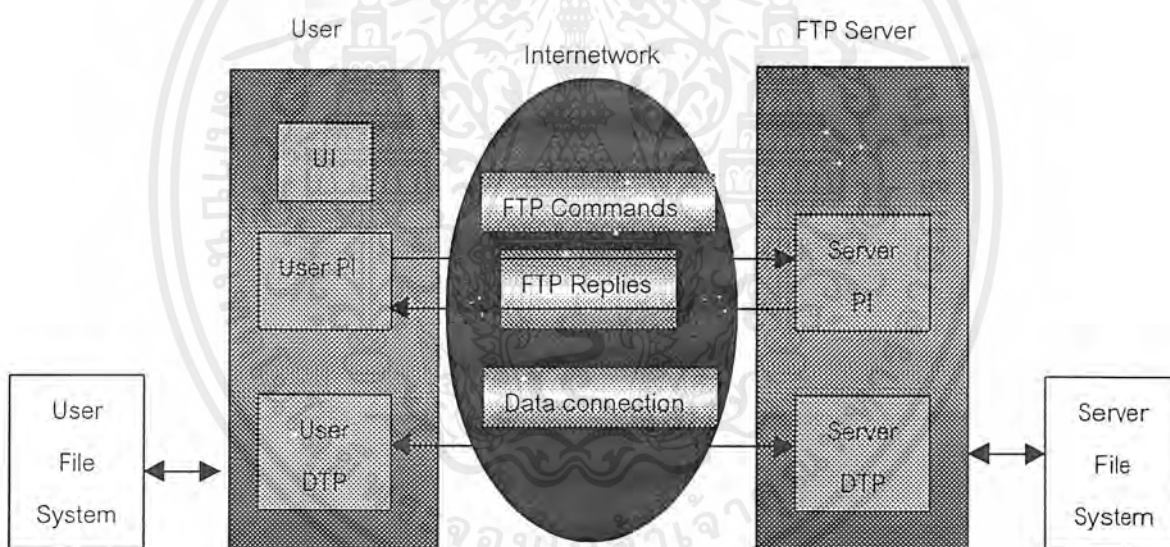
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โพรโทคอล FTP (File Transfer Protocol)

เป็นเทคนิควิธีการที่ใช้ในการรับส่งไฟล์ระหว่างคอมพิวเตอร์สองเครื่อง (Client และ Server) โดยอาศัยโพรโทคอล TCP/IP โดย FTP จะให้ความสามารถแก่ผู้ใช้ในการสำเนาไฟล์ระหว่างระบบ, ดูไดเรกทอรีที่ตั้ง และทำงานทั่วไปที่พบได้บ่อย เช่น การเปลี่ยนชื่อ หรือการลบไฟล์ ซึ่งฟังก์ชันเหล่านี้เป็นส่วนหนึ่งของมาตรฐานในชุดโพรโทคอล TCP/IP

FTP แตกต่างจากโพรโทคอลแอปพลิเคชันส่วนใหญ่ตรงที่ FTP จะใช้โพรโทคอล TCP สองตัวแยกจากกันในการเชื่อมต่อระหว่างโปรแกรมโอนถ่ายข้อมูลส่วนเซิร์ฟเวอร์ และโปรแกรมโอนถ่ายข้อมูลส่วนไคลเอนต์ โดยโพรโทคอล TCP แรกที่ใช้เมื่อเกิดการเชื่อมต่อครั้งแรกเป็น Control connection ใช้สำหรับไคลเอนต์ในการรับส่งคำสั่ง (Command) และสำหรับเซิร์ฟเวอร์ในการส่งการตอบรับ (respond) ส่วนโพรโทคอล TCP ตัวที่สองเป็น Data connection ใช้เพื่อการรับส่งข้อมูล

2.2.1 FTP Model



รูปที่ 2.5 แสดง FTP โมเดล

ส่วนของ FTP Client ประกอบด้วย User Interface (UI) ซึ่งอาจเป็นแบบ Command-line-driven หรือแบบ Window architecture และอีกสองส่วนประกอบคือ User Protocol Interpreter (PI) และ User Data Transfer Process (DTP) ในส่วนของ FTP Server ประกอบด้วย 2 ส่วน คือ Server Protocol Interpreter (PI) และ User Data Transfer Process (DTP)

เมื่อเริ่มทำงาน FTP Client รับชื่อ และ IP address ของ FTP Server จากผู้ใช้ แล้ว Client-PI จะสร้างการเชื่อมต่อโดยใช้โพรโทคอล TCP ไปยังพอร์ตของ FTP Server (หมายเลขพอร์ต 21) ซึ่งเชื่อมต่อ

กับ Server-PI (เป็น Control connection) ในส่วนของ FTP Server เมื่อเริ่มทำงานจะเปิดพอร์ต TCP (หมายเลขพอร์ต 21) ไว้เพื่อรอ Control connection จาก User-PI ซึ่งจะส่งคำสั่งมา และ Server-PI จะส่งคำตอบรับ (Respond) กลับมาให้ผู้ใช้ สถานะต่าง ๆ จะถูกแสดงออกมาทาง UI

ในการส่งผ่านเดต้าจะทำโดย FTP Client จะเปิดพอร์ตชั่วคราว (Local port) และผ่านค่า IP address และหมายเลขพอร์ต TCP ไปให้ FTP Server และใช้พอร์ตนี้ในการส่งผ่านเดต้า ดังนั้นก่อนที่เดต้าจะถูกส่งผ่านระหว่าง FTP Client/Server จำเป็นต้องมีการสร้าง TCP คอนเนกชัน ตัวที่สองระหว่าง Client DTP กับ Server DTP โดยเมื่อ Client-PI ส่งหมายเลขพอร์ตมายัง Server-PI ผ่านทาง Control connection แล้ว Server DTP จะแอกทีฟพอร์ต (port 20) บน เซิร์ฟเวอร์สำหรับ Data connection เมื่อการเชื่อมต่อถูกสร้างขึ้น Client-PI จะส่ง Data transfer command มา แล้วเดต้าจะถูกส่งผ่าน Data connection เมื่อทำการส่งเดต้าเรียบร้อยแล้ว เซิร์ฟเวอร์จะทำให้แอกทีฟของส่วน Data connection ปิดลง เป็นผลให้การเชื่อมต่อในด้าน โคลเอนต์ถูกปิดลงด้วย

2.2.2 DATA TRANSFER FUNCTION

มีหลายคำสั่งที่เกี่ยวข้องกับการส่งผ่านเดต้าระหว่างโฮสต์ คำสั่งของการส่งผ่านเดต้าเหล่านี้จะรวมถึง MODE command ซึ่งใช้ระบุว่ามีขงเดต้าจะถูกส่งผ่านอย่างไร และ STRUcture และ TYPE command ซึ่งถูกใช้เพื่อกำหนดวิธีที่เดต้าจะถูกนำเสนอ

2.2.2.1 DATA TYPE

ผู้ที่เป็นผู้ระบุชนิด (Type) ของ Representation ซึ่ง Type อาจจะเป็นได้ทั้งแบบ Implicit (ASCII, EBCDI) หรือ Local type การกำหนดขนาดไบต์เพื่อแปลงจะอ้างเป็น Logical byte size ซึ่งการส่งทาง Data connection จะเรียกว่า Transfer byte size การนำเสนอรูปแบบที่ใช้มีดังนี้

2.2.2.1.1 ASCII TYPE

เป็น Default type ถูกใช้เป็นตัวเริ่มต้นสำหรับการส่งผ่านเท็กซ์ไฟล์ เดต้าที่ส่งจะถูกส่งในรูปแบบของ Standard 8-bit NVT-ASCII

2.2.2.1.2 EBCDIC TYPE

ใช้สำหรับการส่งระหว่างโฮสต์ ซึ่งใช้ EBCDIC สำหรับ Internal character representation สำหรับส่งข้อมูลที่ถูกส่งจะมีขนาด 8 บิต EBCDIC Character ซึ่ง ASCII และ EBCDIC จะต่างกันที่การระบุหน้าที่

2.2.2.1.3 IMAGE TYPE

การส่งจะส่งเป็นแพค แพคละ 8 บิต การรับจะรับข้อมูลเป็นแพคเช่นกัน Image type จะใช้สำหรับการเก็บ หรือการกู้ข้อมูลกลับมา และใช้สำหรับการส่งไฟล์ข้อมูลประเภทไบนารี

2.1.2.1.4 LOCAL TYPE

สำหรับคอมพิวเตอร์ที่มีขนาดของไบต์แตกต่างกับเครื่องอื่น เช่น 11-bit bytes ข้อมูลที่ส่งจะมีขนาดเป็น Logical byte ระบุโดยพารามิเตอร์ตัวที่ 2 คือ Byte size ค่าของ Byte size ต้องเป็นเลขฐาน 10 จะไม่มีค่าปกติ (Default) ขนาดของ Logical byte ไม่จำเป็นต้องเท่ากับขนาดของ Transfer byte (8-bit byte) ถ้ามีความแตกต่างในขนาดของ Byte logical ควรจะเผด็จัดกัน และการแปลงนี้ต้องสามารถแปลงกลับได้

2.2.2.1.5 FORMAT CONTROL

Character file อาจถูกส่งไปยังโฮสต์ เนื่องด้วย 1 ใน 3 จุดมุ่งหมายดังนี้

1. เพื่อการพิมพ์
2. เพื่อการเก็บ
3. เพื่อการ Processing

ถ้าไฟล์ถูกส่งไปเพื่อการพิมพ์ โฮสต์ที่รับต้องรู้ด้วยว่าจะ Represent vertical format อย่างไร ในกรณีที่สองจะต้องสามารถเก็บไฟล์ที่โฮสต์ และรับ (Retrieve) ภายหลังได้ในฟอร์มเดิม ส่วนในกรณีสุดท้ายควรจะสามารถย้ายไฟล์จากโฮสต์หนึ่งไปยังโฮสต์หนึ่ง และโปรแกรมไฟล์ที่โฮสต์ที่สองได้โดยปราศจากปัญหาที่จะเกิดขึ้น

ซึ่ง Format ASCII หรือ EBCDIC เพียงอย่างเดียวไม่เพียงพอกับความต้องการของเงื่อนไขเหล่านี้ทั้งหมด ดังนั้น Type เหล่านี้จึงมี Second parameter บ่งบอกถึง 1 ใน 3 รูปแบบ ต่อไปนี้

1. NON PRINT เป็น Default format ไฟล์ไม่มีการกำหนด Vertical format ถ้ามีการส่งไปพิมพ์ จะใช้ค่ามาตรฐานที่กำหนดอยู่ในเครื่องพิมพ์ โดยปกติแล้วฟอร์แมตรูปแบบนี้จะใช้สำหรับไฟล์เพื่อการประมวลผล และเก็บลงที่เก็บข้อมูล
2. TELNET FORMAT CONTROL ไฟล์จะเก็บรูปแบบของ Vertical format control (เช่น <CR>, <LF>, <NL>, <VT>, <FF>)
3. CARRIAGE CONTROL (ASA) ไฟล์จะเก็บรูปแบบของ ASA (FORTRAN) Vertical format control

2.2.3 DATA STRUCTURES

FTP ยอมให้โครงสร้างของไฟล์สามารถถูกกำหนดได้ ซึ่งมีอยู่ 3 ประเภท คือ

1. File structure เป็น Default เมื่อ STRUcture command ไม่ได้ถูกเรียกใช้ โครงสร้างของไฟล์จะมีผลกระทบกับทั้ง Transfer mode และ Interpretation and storage ของไฟล์ ในโครงสร้างไฟล์แบบนี้จะไม่มี Internal structure และไฟล์จะถูกพิจารณาให้เป็น Continuous sequence ของเดต้าไบต์
2. Record structure ไฟล์จะถูกสร้างขึ้นในลักษณะของ Sequential records
3. Page structure ใช้สำหรับส่งไฟล์ที่ไม่มีความต่อเนื่อง ซึ่งในบางครั้งจะเรียกว่า "Random access file" ไฟล์เหล่านี้จะอยู่ในรูปของ pages

2.2.4 ESTABLISHING DATA CONNECTION

กลไกของการส่งเดต้าจะประกอบด้วย การ Set up data connection ให้มีพอร์ตที่เหมาะสม และเลือกพารามิเตอร์ในการส่ง ทั้งผู้ใช้ และ Server DTPs จะต้องมีการ Default data port ใน User process จะต้องระบุ Data port ให้ตรงกับ Control connection port ส่วน Server process จะระบุ Data port เป็นพอร์ตที่ติดต่อกับ Control connection port

Byte size ของการส่งข้อมูลจะเท่ากับ 8 bit-byte ขนาดไบนารีมีความสัมพันธ์กับการส่งข้อมูลสัมพันธ์กับการ Representation ของเดต้าภายในระบบไฟล์ของโฮสต์หนึ่ง ๆ

กระบวนการ Data transfer จะคอยฟังที่ Data port ก่อนที่จะส่ง Request command โดย FTP request command จะเป็นตัวกำหนดทิศทางการส่งข้อมูลที่เซิร์ฟเวอร์จะรับ Request จะเริ่มต้นส่ง Data connection ไปที่พอร์ต เมื่อทำการคอนเน็กแล้วการโอนถ่ายข้อมูลจะเริ่มระหว่าง DTP และจากนั้น Server-PI จะตอบไปยัง User-PI

ทุก ๆ FTP จะต้องมีการใช้ Default data port ซึ่ง User-PI เท่านั้นที่สามารถเปลี่ยนเป็น Non-default สามารถเปลี่ยนเป็น Data port ได้โดยใช้ Port command

โดยทั่วไปเซิร์ฟเวอร์จะมีหน้าที่รับผิดชอบในส่วนของการเก็บรักษา Data connection ทั้งตอนเริ่มต้น และการปิด (Close) ยกเว้นเมื่อ User-PI ส่ง Data transfer ใน Transfer mode ที่ Request connection ให้ปิดที่แสดงด้วย EOF Server จะต้องปิด Data connection ภายใต้งื่อนไขดังนี้:

1. เซิร์ฟเวอร์มีการส่งไฟล์ที่สมบูรณ์ใน Transfer mode ที่เรียกให้ปิด Connection โดย EOF
2. เซิร์ฟเวอร์รับคำสั่ง ABORT จากผู้ใช้
3. พอร์ตถูกเปลี่ยนโดยคำสั่งจากผู้ใช้
4. Control connection ปิดเองโดยบังเอิญ หรือกรณีอื่น ๆ

5. เกิดข้อผิดพลาดขึ้น

2.2.5 DATA CONNECTION MANAGEMENT

Default Data Connection Ports : การใช้งานของ FTP implementation ทั้งหมดจะใช้พอร์ต Default มีเพียง User-PI ที่สามารถกำหนดค่าเริ่มต้นการใช้ Non-default port

Negotiating Non-Default Data Ports : User-PI อาจจะระบุ Non-default port ในด้านไคลเอนต์ หรือร้องขอไปยังด้านเซิร์ฟเวอร์ เพื่อให้ระบุ Non-default port

Reuse of the Data Connection : ในการใช้โหมดของการส่งผ่านเดต้าแบบ Stream mode เมื่อถึงจุดสิ้นสุดของไฟล์แล้ว การเชื่อมต่อจะถูกปิดลง เป็นเหตุให้เกิดปัญหาเมื่อมีหลาย ๆ ไฟล์ต้องการการส่งผ่าน มี 2 วิธีที่ใช้ในการแก้ปัญหา คือ หนึ่งทำการใช้ Non-default port และ สองใช้โหมดของการส่งผ่าน โหมดอื่น

1. ใช้ Negotiating non-default data port
2. ใช้ Transfer mode อื่น

2.2.6 TRANSMISSION MODE

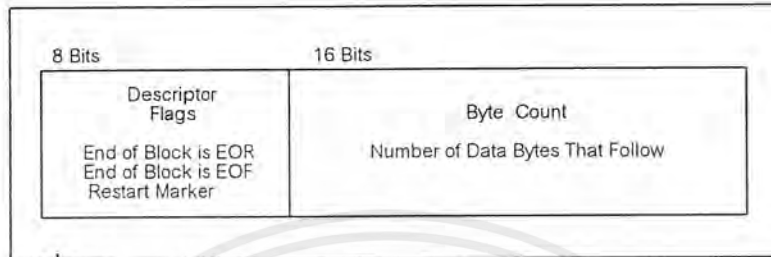
Transmission mode จะถูกใช้ร่วมกับโครงสร้างไฟล์เพื่อกำหนดว่าเดต้าจะถูกเข้ารูป (Format) สำหรับการส่งผ่านอย่างไร ซึ่งมีอยู่ 3 ประเภท คือ

2.2.6.1 STREAM MODE

สำหรับ File-structure ไฟล์จะถูกส่งในลักษณะชุดการไหล (Stream) ของไบนารี FTP จะอาศัย โพรโทคอล TCP เพื่อตรวจสอบความถูกต้องของเดต้า และไม่ต้องมีการเพิ่มแฮชเตอร์ หรือตัวกำหนดขอบเขต (Delimiter) เข้ามาในเดต้า วิธีการเดียวที่จะส่งสัญญาณเมื่อพบว่าถึงจุดสิ้นสุดของไฟล์แล้ว โดยการปิด Data connection สำหรับ Record-structure แต่ละเรคคอร์ดจะถูกกำหนดขอบเขตโดย 2 ไบนารี สำหรับ End Of Record (EOR) control code และอีก 2 byte code ถูกใช้สำหรับ End Of File (EOF)

2.2.6.2 BLOCK MODE

ไฟล์จะถูกส่งผ่านเป็นลำดับชุดของเดต้าบล็อก แต่ละบล็อกเริ่มต้นด้วย 3-byte header ซึ่งเซคเตอร์จะมีรูปแบบดังรูป 2.6



รูปที่ 2.6 แสดงเดต้าบล็อก

สังเกตได้ว่า Descriptor เก็บค่า EOR flag ซึ่งใช้สำหรับกำหนดขอบเขตของเรคคอร์ด และ EOF flag ใช้สำหรับกำหนดความเป็นบล็อกสุดท้ายของไฟล์

2.2.6.3 COMPRESS MODE

นำเสนอรูปแบบที่เป็นพื้นฐานของการบีบอัดเดต้า เมื่อไฟล์ถูกส่งในโหมดนี้ Repeated character จะถูกส่งอย่างมีประสิทธิภาพ การเข้ารหัสพิเศษถูกใช้เพื่อส่งสัญญาณเป็นการบอกว่า Particular character ควรถูกขยาย

2.2.6.4 ERROR RECOVERY AND RESTART

ในปัจจุบันหลาย ๆ องค์กรมีความจำเป็นต้องการจะส่งไฟล์ที่มีขนาดใหญ่มาก สมมติว่าในระหว่างการส่งผ่านไฟล์ที่ใหญ่ขึ้นเกิดการล้มเหลวขึ้น ถ้าเป็น FTP transfer ทั่วไป ก็อาจจะต้องเริ่มต้นทั้งหมดใหม่อีกครั้งหรือไม่ผู้ใช้ก็จะต้องทำการ Reconstruct ไฟล์ขึ้นมาใหม่ด้วยตัวเอง การให้บริการ Restart ถูกออกแบบขึ้นมาเพื่อใช้แก้ปัญหาหนี้ ถ้าบล็อก หรือ Compress mode transfer และการบริการ Restart ถูกนำมาใช้แล้ว การส่ง FTP จะมีความสามารถในการแทรก Restart marker ลงไปในชุดการไหลของข้อมูล เมื่อใดก็ตามที่ผู้รับได้รับตำแหน่ง (Marker) จะเขียนเดต้าไปบน Non-volatile storage และเก็บแท็คของตำแหน่งในเดต้าถ้าระบบของผู้รับเป็นโหนด, ผู้ใช้จะถูกแจ้งถึงตำแหน่งแต่ละตัวทันทีที่เดต้าถูกเก็บ (Store) ถ้าระบบผู้รับเป็นแบบ Remote ข้อความ (Message) จะถูกส่งกลับไปให้ผู้ใช้งาน Control connection เพื่อบ่งชี้ว่าเดต้าถูกทำเป็น Safe-stored หลังจากระบบล้มเหลวผู้ใช้สามารถใช้ Restart command โดยใช้ค่าของตำแหน่งเป็นตัวอาร์กิวเมนต์ การให้บริการ Restart นี้เป็นเพียงทางเลือก มีเพียงไม่กี่ TCP/IP ที่มีการบริการนี้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 FILE TRANSFER FUNCTIONS

2.2.7.1 FTP COMMAND

โปรแกรมโอนถ่ายไฟล์ฝั่งเซิร์ฟเวอร์ ที่สามารถรองรับ FTP command ที่สามารถส่งผ่าน Control connection ได้ดังนี้

2.2.7.1.1 ACCESS CONTROL COMMAND

คำสั่งสำหรับ โคลงอนต์ที่ใช้ในการเข้าถึงเซิร์ฟเวอร์ มีดังตารางที่ 2.1

ตารางที่ 2.1 แสดงคำสั่งประเภท Control command

Command	Definition	Parameter
USER	ใช้บ่งบอกว่าผู้ใช้คือใคร	ชื่อผู้ใช้
PASS	บอกรหัสผ่านของผู้ใช้คนนั้น	รหัสผ่าน
QUIT	ผู้ใช้ต้องการตัดการติดต่อ	ไม่มี
ABOR	ยกเลิกคำสั่งล่าสุด และการ โอนถ่ายไฟล์ที่เกี่ยวข้อง	ไม่มี

2.2.7.1.2 FILE MANAGEMENT COMMAND

คำสั่งที่อนุญาตให้ผู้ใช้จัดการเกี่ยวกับไฟล์ และ โดเรกทอรีที่เซิร์ฟเวอร์ได้ มีดังตารางที่

2.2

ตารางที่ 2.2 แสดงคำสั่งประเภท File management command

Command	Definition	Parameter(s)
CWD	เปลี่ยน โดเรกทอรี	ชื่อ โดเรกทอรีที่ต้องการ ไป
CDUP	เปลี่ยนจาก โดเรกทอรีปัจจุบัน ขึ้นไปยัง โดเรกทอรีแม่	ไม่มี
DELE	ลบไฟล์หรือ โดเรกทอรี	ชื่อไฟล์หรือชื่อ โดเรกทอรี
LIST	แสดงรายการข้อมูลเกี่ยวกับไฟล์	ชื่อ โดเรกทอรี, รายการของไฟล์, หรือ ไม่มี เพื่อเอาข้อมูลเกี่ยวกับ โดเรกทอรีที่ทำงานอยู่
MKD	สร้าง โดเรกทอรีใหม่	ชื่อ โดเรกทอรีใหม่
NLST	แสดงรายการไฟล์ใน โดเรกทอรี	ชื่อ โดเรกทอรี หรือ ไม่ต้องใส่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		พารามิเตอร์ถ้าหมายถึงไครกทอริที่ทำงานอยู่
PWD	แสดงชื่อของไครกทอริที่ทำงานอยู่	ไม่มี
RNFR	ตรวจสอบว่าไฟล์ที่ต้องการเปลี่ยนชื่อมีอยู่จริงหรือไม่	ชื่อไฟล์ที่ต้องการเปลี่ยน
RNTO	เปลี่ยนชื่อไฟล์จากเดิมไปเป็นชื่อใหม่	ชื่อไฟล์ใหม่
SIZE	ให้ค่าขนาดของไฟล์	ชื่อไฟล์

2.2.7.1.3 COMMAND THAT SET DATA FORMATS

คำสั่งที่ใช้ตั้งค่าของเคต้าฟอร์แมตที่จะใช้เมื่อทำการสำเนาไฟล์มีดังตารางที่ 2.3

ตารางที่ 2.3 แสดงคำสั่งประเภท Command that set data formats

Command	Definition	Parameter
TYPE	บอกชนิดของเคต้าและพรีนทฟอร์แมต (ถ้ามี) สำหรับการโอนถ่ายไฟล์	ASCII, EBCDIC Image/Binary, Non-print, Telnet, ASA

2.2.7.1.4 COMMAND THAT SUPPORT FILE

คำสั่งที่ใช้สำหรับเซต Data Connection, สำเนาไฟล์, และรองรับการต่อไฟล์มีดังตารางที่

2.4

ตารางที่ 2.4 แสดงคำสั่งประเภท Command that support file

Command	Definition	Parameter(s)
APPE	นำไฟล์จากไคลเอนต์ต่อท้ายเข้ากับไฟล์ของเซิร์ฟเวอร์	ชื่อไฟล์
PASV	บอกเน็ตเวิร์คแอดเดรส และ	IP แอดเดรส และหมายเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	พอร์ตที่ใช้สำหรับ Data Connection ที่เริ่มต้นโดยไคลเอนต์	พอร์ต
PORT	ไคลเอนต์ส่งเน็ตเวิร์คแอดเดรส และพอร์ตที่ใช้สำหรับ Data connection ที่เริ่มต้นโดยเซิร์ฟเวอร์ (เซิร์ฟเวอร์เป็นผู้เปิดพอร์ต)	IP แอดเดรส และหมายเลขพอร์ต
REST	หาคำแหน่งที่จะเริ่มต้นใหม่ (restart marker) (ตามด้วยคำสั่งโอนถ่ายไฟล์ที่จะถูกเริ่มต้นใหม่)	ค่าของตำแหน่งที่จะเริ่มต้นใหม่
RETR	ฝั่งไคลเอนต์รับไฟล์	ชื่อไฟล์ที่ไคลเอนต์ต้องการดาวน์โหลด
STOR	ส่งไฟล์ไปเก็บไว้ที่เซิร์ฟเวอร์	ชื่อไฟล์ที่ไคลเอนต์ต้องการอัปโหลด
NOOP	บอกให้เซิร์ฟเวอร์ส่งข้อความกลับมาว่า "O.K." (ป้องกันไม่ให้ถูกตัด ในขณะที่ไคลเอนต์ขาดการติดต่อหรือ idle)	ไม่มี
SYST	ถามเซิร์ฟเวอร์ว่าระบบปฏิบัติการของเซิร์ฟเวอร์คืออะไร	ไม่มี

2.2.7.2 FTP REPLIES OR FTP RESPONSE MESSAGE

เมื่อคำสั่ง (Command) ถูกส่งมายังเซิร์ฟเวอร์ FTP Server PI จะแปล และนำการกระทำ (Action) ที่เกี่ยวข้องกับคำสั่งนั้นออกมา แล้วส่งข้อความแสดงสถานะ (Status message) กลับไปให้ไคลเอนต์ผ่านทาง Control connection เพื่อบ่งชี้ถึงความสำเร็จหรือล้มเหลวของการกระทำที่ร้องขอมา The response message Ids ทั้งหมดจะอยู่ในรูป XYZ คำ โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X : determines the general type of the reply

Y : indicates what type of reply

Z : gives more detail

ซึ่งโค้ด XYZ นี้จะถูกเข้าใจโดย PI เพื่อเป็นการง่ายในการใช้ จึงมีการเพิ่มข้อความที่อ่านได้ลงไปด้วย ตัวอย่างเช่น

Event : At login.

Reply : 220 Lagguy Microsoft FTP Service (Version 2.0)

Event : After The output is received

Reply : 226 Transfer complete



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ซ็อกเก็ต

ซ็อกเก็ต คือวิธีการ (Method) สำหรับการสื่อสารกันระหว่างโปรแกรมไคลเอนต์กับโปรแกรมเซิร์ฟเวอร์ในเครือข่าย ซ็อกเก็ตถูกกำหนดให้อยู่ในฐานะ “The endpoint in a connection” ซ็อกเก็ตถูกสร้างและใช้กับชุดของโปรแกรมร้องขอ (Programming request) หรือ Socket application programming interface (API) Socket API ส่วนใหญ่คือ Berkeley UNIX C language interface for sockets ซ็อกเก็ตสามารถถูกใช้เพื่อสื่อสารกันระหว่างโพรเซส (Processes) ที่อยู่ภายในเครื่องคอมพิวเตอร์เดียวกัน ในระบบแบบ “Connectionless” ของอินเทอร์เน็ต (ซึ่งเซิร์ฟเวอร์ต้องจัดการกับการร้องขอจากไคลเอนต์หลาย ๆ ที่ และไม่มีการรักษาการเชื่อมต่อไว้นาน)

โดยทั่วไปซ็อกเก็ตจะสื่อสารข้อมูลกับซ็อกเก็ตตัวอื่น ๆ (ที่กำลังทำงานอยู่) ใน Communication Domain เดียวกันโดยใช้ IP ช่วย

2.3.1 ชนิด ซ็อกเก็ต

ซ็อกเก็ต สามารถแบ่งได้เป็น 2 ชนิด คือ

2.3.1.1 ซ็อกเก็ต แบบ Stream

ทำงานกับข้อมูลที่จะส่งไปโดยไม่กำหนดขอบเขตของเรคคอร์ด หรือไม่เป็นเรคคอร์ด (ข้อมูลที่ส่งเรียกว่า a stream of bytes) ซึ่งสามารถส่งได้ทั้ง 2 ทาง (ส่ง และรับในซ็อกเก็ตเดียวกัน) และสตรีมจะรับประกันว่าข้อมูลถูกส่งถึงปลายทางอย่างถูกต้อง โดยข้อมูลไม่สลับกัน และไม่มีการส่งซ้ำซ้อน โดยสตรีมยังสามารถจัดการส่ง และรับข้อมูลขนาดใหญ่ได้ดีอีกด้วยเพราะโพรโตคอลใน Network Layer อาจแตกข้อมูลให้อยู่ในขนาดที่สมควร

การสื่อสารข้อมูลแบบสตรีมนั้นเป็นการสื่อสารข้อมูลแบบ Connection-Oriented สตรีมนั้นจะต้องมีการเชื่อมต่อกันอย่างชัดเจน ถ้าซ็อกเก็ต A ต้องการเชื่อมต่อกับ ซ็อกเก็ต B แล้ว ซ็อกเก็ต A จะต้องมีกรร้องขอเชื่อมต่อไปยัง ซ็อกเก็ต B ซ็อกเก็ต B จะต้องยอมรับหรือปฏิเสธการร้องขอนั้น ๆ เหมือนกับโทรศัพท์ ซึ่งจะต้องมีผู้เรียกและผู้รับสายจึงจะคุยกันได้ และการสนทนาจะไม่มีเสียงซ้ำกัน (ถ้าผู้พูดไม่ได้พูดซ้ำ) และลำดับของเสียงที่พูดจะตรงกับลำดับของเสียงที่ผู้ฟังได้ยิน และที่สำคัญที่สุดคือเสียงที่ผู้พูดพูดไปนั้นจะถูกส่งไปถึงผู้รับแน่นอน ด้วยเหตุนี้จึงทำให้ Stream socket เป็นที่นิยมมากกว่า

Datagram socket

2.3.1.2 ซ็อกเก็ต แบบดาต้าแกรม

Datagram sockets สนับสนุนการส่งข้อมูลสองทาง โดยไม่รับประกันว่าข้อมูลจะถูกส่งไปถึงอย่างถูกต้องหรือไม่ และข้อมูลอาจมีการสลับกัน หรือ ซ้ำซ้อนแต่ขนาดของเรคคอร์ดจะได้รับการปกป้องรักษา แม้ว่าขนาดของเรคคอร์ดจะใหญ่กว่าขนาดที่ผู้รับกำหนด

การสื่อสารข้อมูลแบบดาต้าแกรม เป็นการสื่อสารข้อมูลแบบ Connectionless ไม่จำเป็นต้องสร้างการเชื่อมต่อโดยชัดเจน เพียงส่ง Datagram message ไปยังซ็อกเก็ตที่กำหนดก็จะสามารถสื่อสารข้อมูลกันได้แล้ว และดาต้าแกรมซ็อกเก็ตยังมีความสามารถในการส่งข้อมูลแบบกระจาย (Broadcasting Message)

2.3.2 พอร์ต และแอดเดรสของซ็อกเก็ต

2.3.2.1 พอร์ต

พอร์ตเป็นเลขที่ใช้กำหนดค่าให้กับซ็อกเก็ตแต่ละตัว โดยจะไม่มีเลขที่ซ้ำกันเกิดขึ้นพร้อมกัน ดังนั้นซ็อกเก็ตของแอปพลิเคชันจะมีค่าพอร์ตที่ไม่ซ้ำกัน ทำให้สามารถมีแอปพลิเคชันที่ทำงานกับซ็อกเก็ตได้หลายตัวพร้อมกัน

2.3.2.2 ซ็อกเก็ต แอดเดรส

ซ็อกเก็ตแต่ละตัวจะติดต่อกับ Internet Protocol (IP) Address ในเครือข่ายโดยทั่วไปแอดเดรสจะเปรียบเสมือนชื่อเครื่องเช่น www.kmitl.ac.th หรือ “161.246.10.21”

2.3.3 การทำงานระหว่าง ซ็อกเก็ต และ Application layer

สตรีมของเดต้าถูกแบ่งเป็นดาต้าแกรม ส่งไปยังคอมพิวเตอร์อีกเครื่อง และดาต้าแกรมถูกนำมารวมกันในภายหลัง อย่างไรก็ตามวิธีที่จะเปิดคอนเนกชันไปยังคอมพิวเตอร์ที่ระบุ และบอกคอมพิวเตอร์ว่า คุณต้องการไฟล์ใด และควบคุมการถ่ายโอนของไฟล์ งานเหล่านี้จะถูกทำโดยแอปพลิเคชันโพรโตคอล ซึ่งรันอยู่บน TCP/IP นั่นก็คือ เมื่อแอปพลิเคชันโพรโตคอลต้องการส่งข้อความ มันก็จะให้ข้อความนั้นแก่ TCP จากนั้น TCP ตรวจสอบให้แน่ใจว่าข้อความนั้นถึงจุดหมายปลายทาง เนื่องจาก TCP และ IP ดูแลในเรื่องรายละเอียดทางด้านเครือข่ายทั้งหมด แอปพลิเคชันโพรโตคอลสามารถปฏิบัติต่อ Network connection เสมือนว่ามันเป็นเพียงไบนารีสตรีมธรรมดา ๆ ก่อนที่จะลงดีกรายละเอียดเกี่ยวกับแอปพลิเคชันโปรแกรมจะต้องอธิบายก่อนว่าจะหาแอปพลิเคชันได้อย่างไร สมมติว่าต้องการจะส่งไฟล์ไปที่คอมพิวเตอร์ที่มี Internet Address 128.6.4.7 คุณต้องการมากกว่าแค่ Internet Address อย่างเดียวเท่านั้น คุณต้องติดต่อไปยัง FTP Server ที่ปลายอีกข้างหนึ่ง ตามปกติแล้ว Network program จะถูกทำขึ้นมาพิเศษเพื่องานนั้น ๆ โดยเฉพาะ ระบบส่วนใหญ่มีโปรแกรมแยกต่างหากไว้สำหรับจัดการกับการโอนถ่ายไฟล์, ไว้สำหรับ Remote terminal logins, สำหรับเมล์ หรือ อื่น ๆ เมื่อคุณคอนเนกไปที่ 129.6.4.7 คุณต้องระบุว่าคุณต้องการจะคุยกับ FTP Server โดยแต่ละ Server มี “Well-known sockets” ทวนอีกครั้งว่า TCP ใช้หมายเลขพอร์ตในการเก็บรายละเอียดต่าง ๆ ของแต่ละ Conversation อย่างไรก็ตามหมายเลขพอร์ตที่เฉพาะเจาะจงจะถูกกำหนดให้กับโปรแกรมที่ร้องขอ Request (การร้องขอ) อยู่ ตัวอย่างเช่น ถ้าคุณต้องการส่งไฟล์ คุณจะ Start โปรแกรม FTP ซึ่งโปรแกรม FTP นี้จะเปิดคอนเนกชัน

โดยใช้เลขคู่ (เช่น 1234) สำหรับหมายเลขพอร์ตที่ปลายทางฝั่งของมันเอง อย่างไรก็ตามพอร์ตหมายเลข 21 ให้กับปลายทางอีกฝั่งหนึ่ง นี่คือพอร์ตอย่างเป็นทางการสำหรับ FTP Server ในที่นี้จะมีสองโปรแกรมที่ต่างกันมาเกี่ยวข้อง โดยคุณรัน FTP บนฝั่งของคุณ

โปรแกรมที่ออกแบบมาเพื่อตอบรับคำสั่งจากเทอร์มินอลของคุณและส่งคำสั่งต่อไปยังปลายทางอื่น โปรแกรมที่คุณคุยด้วยบนอีกเครื่องหนึ่งก็คือ FTP Server ซึ่งถูกออกแบบมาเพื่อตอบรับคำสั่งจาก Network connection แทนที่จะตอบรับจาก Interactive terminal ไม้มีความจำเป็นใดสำหรับโปรแกรมของคุณที่จะใช้เลขซ็อกเก็ตที่ Well-known อย่างไรก็ตามเซิร์ฟเวอร์ต้องมีเลขซ็อกเก็ตที่ Well-known เพื่อคนจะได้สามารถเปิดคอนเนกชัน และส่งคำสั่งไปยังเซิร์ฟเวอร์ได้ หมายเลขพอร์ตอย่างเป็นทางการสำหรับแต่ละโปรแกรมอยู่ใน “Assigned Number” คอนเนกชันหนึ่ง ๆ จะถูกอธิบายด้วยเซตของเลข 4 จำนวน คือ Internet Address ของแต่ละปลายทาง และหมายเลข TCP พอร์ตของแต่ละปลายทาง แต่ละดาต้าแกรมจะมีเลขทั้ง 4 นี้ (Internet Address อยู่ใน IP header และหมายเลข TCP พอร์ตอยู่ใน TCP เฮดเดอร์) ไม่มี 2 คอนเนกชันใด ๆ ที่สามารถมีเซตของเลขเหมือนกันได้ อย่างไรก็ตาม มีแค่เพียงตัวเลขตัวหนึ่งที่แตกต่างกันก็พอแล้ว ตัวอย่างเช่น เป็นไปได้ที่จะมีผู้ใช้ 2 คนที่แตกต่างกันบนคอมพิวเตอร์เครื่องเดียวกันที่กำลังส่งไฟล์ไปยังเครื่องเดียวกัน คอนเนกชันจะมีพารามิเตอร์ดังนี้ คอนเนกชัน ที่ 1 : 128.6.4.194, 128.6.4.7 1234 , 21 คอนเนกชัน ที่ 2 : 128.6.4.194, 128.6.4.7 1235 ,21 เนื่องจากใช้เครื่องคอมพิวเตอร์เครื่องเดียวกัน Internet Address จึงเหมือนกันและเนื่องจากผู้ใช้ทั้งสองกำลังถ่ายโอนไฟล์ปลายทางข้างหนึ่งของคอนเนกชันจึงเกี่ยวข้องกับหมายเลขพอร์ตที่ Well-known สำหรับ FTP สิ่งเดียวที่แตกต่างก็คือ หมายเลขพอร์ตสำหรับโปรแกรมที่ผู้ใช้รันอยู่ ซึ่งนั่นก็เป็นสิ่งแตกต่างที่เพียงพอแล้ว โดยปกติแล้วอย่างน้อยฝั่งหนึ่งฝั่งใดของคอนเนกชันจะขอให้ Network software กำหนดหมายเลขพอร์ตที่รับประกันได้ว่าจะไม่เหมือนใคร ซึ่งตามธรรมชาติจะเป็นฝั่งผู้ใช้เนื่องจากเซิร์ฟเวอร์ต้องใช้เลขที่ Well-known เราได้ทราบถึงวิธีการเปิดคอนเนกชันแล้ว ต่อไปนี้จะกล่าวถึงแอปพลิเคชันดังที่ได้กล่าวไปข้างต้น เมื่อ TCP ได้เปิดคอนเนกชันแล้ว เราก็มักย้าย ๆ สายไว้ติดต่อก่อนที่ยังยากก็เป็นหน้าที่ของ TCP และ IP อย่างไรก็ตามเรายังจำเป็นต้องมีการตกลงกันเรื่องสิ่งที่จะส่งผ่านคอนเนกชันซึ่งก็คือ ข้อตกลงในเรื่องเซตของคำสั่งที่แอปพลิเคชันจะเข้าใจ และรูปแบบว่าจะส่งสิ่งเหล่านั้นอย่างไร โดยทั่วไปสิ่งที่จะส่งก็คือ คำสั่ง และเคต้ารวมกัน ใช้ Context (อรรถอธิบาย) เพื่อให้แตกต่าง ตัวอย่างเช่น เมล์โพร โทคอลทำงานดังนี้ : โปรแกรมเมล์ของคุณเปิดคอนเนกชันไปยังเมลเซิร์ฟเวอร์ที่ปลายทางอีกฝั่งหนึ่ง โปรแกรมคุณให้ ชื่อเครื่อง, ผู้ส่งข้อความ และผู้รับ จากนั้นโปรแกรมจึงส่งคำสั่งที่บอกว่ากำลังจะเริ่มส่วนที่เป็นข้อความ ณ จุดนั้น ปลายทางอีกฝั่งหนึ่งจะหยุดปฏิบัติกับ (คิดว่า) สิ่งที่ได้รับว่าเป็นคำสั่ง เมื่อถึงจุดสิ้นสุดของข้อความเครื่องหมายพิเศษ (เครื่องหมาย จุด “.” ในแถวแรก) จะ

ถูกส่งไปยัง Mail Server หลังจากนั้นปลายทางทั้งสองจะเข้าใจว่าโปรแกรมของคุณกำลังจะส่งคำสั่งต่อ นี่คือวิธีที่ง่ายที่สุด และเป็นวิธีที่แอปพลิเคชันส่วนใหญ่ใช้กัน การถ่ายโอนโปรแกรมก่อนข้างจะยุ่งยากกว่านั้น File Transfer Protocol เกี่ยวข้องกับ 2 คอนเน็กซ์ที่แตกต่างกันโดยเริ่มต้นเหมือนกับเมล์โปรแกรมของผู้ใช้ส่งคำสั่งอย่างเช่น “log me in as this user”, นี่คือ Password ของฉัน, ส่งไฟล์ชื่อนี้ให้ฉัน อย่างไรก็ตามเมื่อคำสั่งที่จะส่งเด้าถูกส่งไป คอนเน็กซ์ที่สองจะถูกเปิดเพื่อส่งเด้าโดยเฉพาะ เป็นไปได้แน่นอนที่จะส่งเด้าบนคอนเน็กซ์เดียวกับคำสั่งอย่างที่มีเมลล์ทำ อย่างไรก็ตามการถ่ายโอนไฟล์บ่อยครั้งที่ใช้เวลานาน ผู้ออกแบบ File Transfer Protocol ต้องการปล่อยให้ผู้ใช้สามารถส่งคำสั่งต่อได้เลย ในขณะที่กำลังมีการถ่ายโอนไฟล์อยู่ ตัวอย่างเช่น ผู้ใช้อาจจะมีข้อซักถาม หรือ อาจจะยกเลิกการถ่ายโอนด้วยเหตุนี้ผู้ออกแบบรู้สึกว่าเป็นการดีที่สุดที่จะใช้อีกคอนเน็กซ์สำหรับเด้าต่างหาก และปล่อยคอนเน็กซ์เดิมไว้สำหรับคำสั่ง (และยังเป็นไปได้ที่จะเปิดคอนเน็กซ์สำหรับคำสั่งไปยังคอมพิวเตอร์ 2 เครื่องที่ต่างกัน และบอกให้ทั้งสองว่าให้ส่งไฟล์จากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง) Remote terminal connections ใช้อีกกลไกหนึ่ง สำหรับ Remote logins, มีเพียงแค่หนึ่งคอนเน็กซ์เท่านั้น โดยปกติจะส่งเด้าเมื่อจำเป็นที่จะต้องส่งคำสั่ง (เช่น เพื่อที่จะ Terminal type หรือเปลี่ยน Mode) จะใช้อักษรพิเศษเพื่อบ่งชี้ว่าตัวอักษรต่อไปคือคำสั่ง ถ้าผู้ใช้ต้องการจะพิมพ์อักษรพิเศษนั้นเป็นเด้า ต้องพิมพ์สองตัวส่งไป

อย่างไรก็ตามมีระเบียบแบบแผนที่แอปพลิเคชันใช้เหมือนกันอยู่สองอย่างที่จะกล่าวถึงคือ ข้อแรก The common network representation : TCP/IP ตั้งใจให้สามารถใช้ได้บนคอมพิวเตอร์ใด ๆ แต่โชคไม่ดีที่ทุกคอมพิวเตอร์จะตกลงในเรื่องว่าจะแสดงเด้าอย่างไร มีความแตกต่างในเรื่องรหัสตัวอักษร (ASCII vs. EBCDIC), เรื่องจุดสิ้นสุดของบรรทัด (carriage return, line feed หรือ a representation using counts) และแตกต่างในเรื่องว่า Terminal คาดหวังว่าตัวอักษรจะถูกส่งมาแยกเป็นตัวหรือเป็นบรรทัดต่อครั้ง (แต่ TCP/IP ไม่สนใจเกี่ยวกับ Representation TCP ส่งในรูปของ Octets ห้าแปดลงตัว) อย่างไรก็ตามโปรแกรมที่ทั้งสองฝั่งจะต้องตกลงว่าจะแปล Octets อย่างไร RFC สำหรับแต่ละแอปพลิเคชัน ระบุ Representation มาตรฐานสำหรับแอปพลิเคชันนั้น ๆ ปกติจะเป็น “net ASCII” ซึ่งใช้ตัวอักษร ASCII จุดสิ้นสุดของบรรทัดใช้ Carriage return ตามด้วย Line feed แอปพลิเคชันส่วนใหญ่ก็เช่นกัน ทำบทบัญญัติสำหรับคอมพิวเตอร์ 2 เครื่องในข้อตกลงเรื่อง Representation แบบอื่น ๆ ที่อาจพบว่าจะสับสน ตัวอย่างเช่น PDP-10 เป็นแบบ 36-bit words ก็มีทางที่ PDP-10's สองเครื่องจะตกลงที่จะส่งแบบ 36-bit binary file ในทำนองเดียวกัน สองระบบที่ขอบแบบ Full-duplex ก็สามารถตกลงกันเองได้ อย่างไรก็ตามแต่ละแอปพลิเคชันมี Representation มาตรฐานที่ทุกเครื่องต้องรองรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เครื่องมือที่ใช้ในการโปรแกรม

2.4.1 ภาษาจาวา

ในปี 1995 Sun Microsystems ได้อธิบายลักษณะของจาวาว่า “เป็นภาษาที่ง่ายต่อการเขียนโปรแกรม และการตีความ มีลักษณะอ้างอิงเชิงวัตถุ (Object Oriented) สะดวกต่อการใช้งาน มีสถาปัตยกรรมที่เป็นกลางมีประสิทธิภาพ มีการทำงานแบบมัลติเทร็ด (Multithread) และเป็นภาษาแบบไดนามิก” จาวาอนุญาตให้ผู้เขียนโปรแกรมสามารถเพิ่มเติมฟังก์ชันต่าง ๆ ในแพ็คเกจ (Package) เล็ก ๆ ซึ่งสามารถทำงานได้ทุกที่ แม้แต่บนระบบที่มีความแตกต่างกันอย่างมากบนเน็ตเวิร์ค (Network) เช่น อินเทอร์เน็ต ซึ่งเป็นลักษณะพิเศษของจาวาที่ออกแบบมารองรับการทำงานบนเน็ตเวิร์ค ซึ่งภาษาคอมพิวเตอร์อื่น ๆ อีกหลายตัวไม่มีความสามารถทางด้านนี้

จาวามีลักษณะคล้ายกันกับโปรแกรมภาษา C และ C++ ซึ่งมีลักษณะเป็นภาษาอ้างอิงเชิงวัตถุ แต่แตกต่างกันตรงที่ภาษาจาวา ได้ถูกออกแบบมาเพื่อช่วยให้ใช้งานบนระบบปฏิบัติการใด ๆ ก็ได้ ไม่ว่าจะเป็นคอสม, วินโดวส์, ยูนิกซ์ หรือแม้กระทั่งเครื่องระบบเมนเฟรม โดยการเขียนโปรแกรมเพียงครั้งเดียวช่วยลดเวลาการทำงานของผู้เขียนโปรแกรมที่ต้องการเขียนโปรแกรมสำหรับใช้งานบนระบบปฏิบัติการหลาย ๆ ระบบ ทำให้จาวาเป็นภาษาที่มีประสิทธิภาพสูง

2.4.1.1 ประเภทของการโปรแกรมในจาวาโปรแกรมมิ่ง

1. จาวาแอปพลิเคชัน (JAVA Application) เป็นการนำจาวามาเขียนโปรแกรมที่สามารถนำมาใช้งานได้โดยอิสระ (Stand Alone Program) เหมือนกับการเขียนโปรแกรมภาษาระดับสูงอื่น ๆ เช่น C++, ปาสคาล ทั้งนี้สามารถนำแอปพลิเคชันไปใช้งานที่คอมพิวเตอร์ต่างแพลตฟอร์ม (Platform)
2. จาวาแอปเพลต (JAVA Applet)

2.4.1.2 ขั้นตอนของ JAVA

1. สร้างซอร์สโค้ด (Source Code) ขึ้นมาก่อน ชื่อไฟล์จะต้องเป็นชื่อเดียวกับคลาสหลัก เมื่อสร้างเสร็จแล้วให้กำหนดนามสกุลเป็น .java
2. นำไฟล์ที่ได้จากข้อ 1 มาแปลงเป็นไบต์โค้ด (Bytecode) ด้วยวิธีการที่เรียกว่าคอมไพล์ (Compile) เมื่อผ่านการคอมไพล์แล้วจะได้ไฟล์ใหม่มา 1 ไฟล์ โดยมีชื่อเดิมแต่นามสกุลเป็น .class
3. นำไฟล์ที่ได้จากข้อ 2 มาเรียกใช้ ขั้นตอนนี้จะเรียกว่าอินเตอร์พรีต (Interpret) โดยจะเรียกใช้ที่คอมพิวเตอร์ต่างแพลตฟอร์มได้

2.4.1.3 ลักษณะเด่นของ JAVA

1. จาวามีขนาดเล็ก และเป็นภาษาที่ง่ายไวยกรณ์ (Syntax) ของจาวาถูกพัฒนามาจาก C++ ทำให้โปรแกรมเมอร์ที่คุ้นเคยกับภาษานี้สามารถเรียนรู้จาวาได้อย่างง่ายดาย ภาษาจาวาเป็นโปรแกรมภาษาที่ออกแบบมาเพื่อให้เขียนง่าย กระทัดรัด แต่มากด้วยประสิทธิภาพ
2. อัจฉริยะเชิงวัตถุ (Object Oriented Programming) ภาษาจาวาเป็นภาษาที่มีลักษณะอัจฉริยะเชิงวัตถุ (Object Oriented Programming :OOP) ซึ่งการเขียนโปรแกรมลักษณะนี้มีข้อดีคือ สามารถนำโปรแกรมที่เขียนไว้แล้วกลับมาใช้ใหม่โดยโปรแกรมอื่น ๆ ได้อีกโดยง่าย แนวคิดและวิธีการส่วนใหญ่ในการเขียนโปรแกรมอัจฉริยะเชิงวัตถุของภาษาจาวานั้น สืบทอดมาจากภาษา C++
3. การทำงานของจาวาที่ไม่ขึ้นกับแพลตฟอร์ม ซอฟต์แวร์ของคอมพิวเตอร์ส่วนมากจะถูกพัฒนาเฉพาะสำหรับระบบปฏิบัติการบางประเภทเท่านั้น แต่เมื่อพัฒนาโปรแกรมด้วยภาษาจาวา จะสามารถนำโปรแกรมที่พัฒนาขึ้นนี้ไปใช้งานในระบบปฏิบัติการของคอมพิวเตอร์แบบใด ๆ ก็ได้ โดยไม่ต้องทำการเปลี่ยนแปลงใด ๆ ทั้งสิ้น เพราะสถาปัตยกรรมของจาวาเป็นแบบไบนารี ซึ่งเครื่องคอมพิวเตอร์ทุกประเภทสามารถทำการประมวลผลโค้ดแบบไบนารีได้ ข้อดีนี้จะช่วยลดเวลาการพัฒนาโปรแกรมได้มาก

2.4.1.4 JDK (Java Development Kit)

เป็นชุดโปรแกรมที่ใช้สำหรับสร้างและพัฒนาโปรแกรมภาษาจาวา JDK ที่ใช้ในโครงการพิเศษนี้เป็นเวอร์ชัน 1.2.1 เนื่องจากทางคณะผู้จัดทำโครงการพิเศษใช้เครื่องคอมพิวเตอร์ส่วนบุคคลกับระบบปฏิบัติการวินโดวส์ 95 อยู่ จึงดาวน์โหลดไฟล์ข้อมูลที่ชื่อ “jdk121-win32.exe” สำหรับวินโดวส์ 95 มาจาก <http://ftp.javasoft.com> หลังจากดาวน์โหลดโปรแกรมมาเรียบร้อยแล้วให้ทำการติดตั้งโปรแกรมจากนั้นให้แก้ไขไฟล์ “AUTOEXEC.BAT” โดยทำตามขั้นตอนดังนี้

1. คลิกเมาส์ที่ปุ่ม Start
2. เลือก Run บนเมนูที่ปรากฏขึ้นมา
3. พิมพ์คำว่า “sysedit” ลงในบ็อกซ์ของ Open
4. คลิกเมาส์ที่ปุ่ม OK ภายในบ็อกซ์ จะปรากฏรายชื่อ ไฟล์บนระบบแสดงขึ้นมา
5. เลือกไฟล์ “AUTOEXEC.BAT” แล้วนำไปเพิ่มเติมพาทชื่อ c:\jdk1.2.1\bin ไว้ในคำสั่ง Path ที่มีอยู่เดิมใน “AUTOEXEC.BAT” โดยใช้เครื่องหมาย “ ; ” ต่อท้ายตัวอักษรสุดท้ายในคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.5 เครื่องมือที่ใช้ในการทำงานกับจาวา

จะอยู่ภายในชุดโปรแกรมของ JDK ได้แก่

1. javac.exe จาวาคอมไพเลอร์ (JAVA Compiler) ใช้ในการคอมไพล์ (compile) ไฟล์ซอร์สโค้ดโดย มีส่วนขยายเป็น .java ให้เป็นไบนารีโค้ดโดยมีส่วนขยายเป็น .class รูปแบบของคำสั่งการใช้ javac คือ

```
javac [option] filename.java
```

2. java.exe ตัวแปลภาษาจาวา (JAVA Interpreter) ใช้สำหรับทำการรันโปรแกรมแอปพลิเคชันที่คอมไพล์แล้ว รูปแบบคำสั่งการใช้ java คือ

```
java [option] classname
```

หมายเหตุ : *Classname* ที่ใช้ในคำสั่งนี้หมายถึงเฉพาะชื่อของคลาสเท่านั้น *ไม่ต้องใส่ .class* ลงไปด้วย

3. jdb.exe จาวาดีบักเกอร์ (JAVA Debugger) เป็นโปรแกรมที่ใช้สำหรับตรวจสอบหาข้อบกพร่องบนสภาพแวดล้อมของจาวา การใช้งาน jdb จะเป็นลักษณะคอมมานด์ไลน์ไดรเวน (Command-Line driven) การตรวจสอบสามารถทำได้กับไฟล์ทั้งบนเครื่องคอมพิวเตอร์ท้องถิ่น (Local) และบนเครื่องคอมพิวเตอร์ควบคุม (Remote) แต่เมื่อกำหนดการตรวจสอบไฟล์ของเครื่องควบคุมต้องใช้ออฟชั่น (Option) การตรวจสอบที่ระบุโฮสต์ (Host) และต้องรู้รหัสผ่านด้วย รายละเอียดของออฟชั่นทั้งสองออฟชั่นมีดังนี้

ตารางที่ 2.5 แสดงรายละเอียดของออฟชั่นทั้งสองออฟชั่น

ออฟชั่น	คำอธิบายหน้าที่ของออฟชั่น
Host hostname	ออฟชั่นนี้บอก jdb ให้ทราบว่า โปรแกรมหรือไฟล์ภาษาจาวาเก็บไว้ที่ใด
Password password	ออฟชั่นนี้เป็นการส่งรหัสผ่านของไฟล์นั้นเพื่อเข้าตรวจสอบไฟล์บนเครื่องคอมพิวเตอร์ควบคุม รหัสผ่านนี้ต้องเป็นรหัสเดียวกันกับที่สร้างขึ้นด้วย Java Debug (ออฟชั่นหนึ่งของตัวแปลภาษาจาวา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.6 เครื่องข่ายเบื้องต้น

คอมพิวเตอร์แต่ละเครื่องสามารถติดต่อสื่อสารกันบนเครือข่ายได้ โดยใช้ Transport Control Protocol (TCP) หรือ User Datagram Protocol (UDP) เป็นตัวกลางในการสื่อสาร ดังตัวอย่างตามตาราง

ตารางที่ 2.6 แสดงตัวอย่างการติดต่อสื่อสารกันบนเครือข่ายโดยใช้ TCP และ UDP

เลเยอร์	ตัวอย่าง
แอปพลิเคชัน (Application)	HTTP, FTP, SMTP, Telnet
ทรานสปอร์ต (Transport)	TCP, UDP
เน็ตเวิร์ค (Network)	IP
ฟิสิคัล	Ethernet, Fiber Optic, Token Ring

หมายเหตุ : จาวาทำงานอยู่ในชั้นทรานสปอร์ต

การเขียนโปรแกรมจาวาที่ใช้ติดต่อสื่อสารกันบนเครือข่ายจะต้องทำการเขียนโปรแกรมที่ชั้นแอปพลิเคชัน โดยไม่ต้องกังวลกับ TCP และ UDP เลยเพราะภาษาจาวามีการรองรับการรับส่งข้อมูลผ่านเครือข่ายอยู่ในตัวเองในแพคเกจ java.net โดยจะต้องประกาศในรูปแบบดังนี้

```
import java.net.*;
```

แพคเกจ java.net ประกอบด้วยคลาสที่ใช้สร้างการติดต่อกับเครือข่าย มักใช้ควบคู่กับแพคเกจ java.io สำหรับเขียนและอ่านข้อมูลจากเครือข่าย

หมายเหตุ : แพคเกจ java.io ประกอบด้วยคลาสสำหรับปฏิบัติงานกับอินพุต (Input) และเอาต์พุต (Output)

2.4.1.7 ข้อแตกต่างระหว่าง TCP และ UDP

2.4.1.7.1 TCP

เมื่อ 2 แอปพลิเคชันต้องการติดต่อกัน แอปพลิเคชันทั้งสองจะต้องสร้างการเชื่อมต่อ (connection) และส่งข้อมูลไปมาในระหว่างการเชื่อมต่อ ซึ่งก็เหมือนกับการโทรศัพท์นั่นเอง TCP จะมีการรับประกันด้วยว่าข้อมูลที่ถูกส่งออกไปถูกต้องครบถ้วน และสมบูรณ์

TCP จัดการการติดต่อแบบจุดต่อจุด (Point-to-Point Channel) สำหรับแอปพลิเคชันที่ต้องการความน่าเชื่อถือระหว่างการติดต่อ ตัวอย่างแอปพลิเคชันที่ต้องการความน่าเชื่อถือระหว่างการติดต่อ เช่น Hypertext Transfer Protocol (HTTP), โปรแกรมโอนถ่ายข้อมูล (File Transfer Protocol : FTP และ เทลเน็ต (Telnet) เป็นต้น

2.4.1.7.2 UDP

โปรโตคอล UDP จัดการการสื่อสารที่ไม่ต้องการการการันตีระหว่าง 2 แอปพลิเคชันบนเครือข่าย UDP จะส่งข้อมูลออกไปเป็นแพคเกจเรียกว่า ดาต้าแกรม (Datagram) การส่งดาต้าแกรมนี้จะเหมือนกับการส่งจดหมายผ่านไปรษณีย์ ซึ่งในลำดับขั้นตอนการส่งนั้นไม่สำคัญ และไม่รับประกันว่าข้อมูลจะถึงผู้รับหรือไม่

2.4.1.8 พอร์ต (Port)

การติดต่อสื่อสารระหว่างคอมพิวเตอร์จะทำงานติดต่อผ่านพอร์ตเพื่อทำการรับส่งข้อมูล และพอร์ตยังใช้ในการแยกข้อมูลที่ส่งมาให้แอปพลิเคชันของอีกเครื่องด้วยเพราะ 1 แอปพลิเคชัน จะมี 1 พอร์ตเท่านั้นข้อมูลจะถูกส่งออกมาพร้อมกับ IP Address (32 bit) ซึ่งเป็น IP Address ของเครื่องปลายทาง หมายเลขพอร์ต (16 bit) จะถูกใช้โดย TCP และ UDP เพื่อส่งข้อมูลไปยังแอปพลิเคชันที่ต้องการ หมายเลขพอร์ตจะมีขอบเขตตั้งแต่ 0-65535 โดยที่หมายเลขพอร์ตในช่วง 0-1023 จะถูกสงวนเอาไว้ให้กับบริการหลัก ๆ เช่น HTTP และ FTP

2.4.1.9 คลาสเน็ตเวิร์คใน JDK (Network Class in JDK)

คลาสที่อยู่ในแพคเกจ java.net ใช้สำหรับการติดต่อสื่อสารผ่านเครือข่ายด้วย TCP มีดังนี้ URL, URLConnection, Socket และ ServerSocket ซึ่งในการเขียนโปรแกรมโอนถ่ายข้อมูลฝั่งเซิร์ฟเวอร์ (FTP Server) จะใช้คลาส Socket และคลาส ServerSocket

2.4.1.10 การติดต่อกับซ็อกเก็ต (Socket)

ในการติดต่อผ่านเครือข่ายจะต้องอาศัยซ็อกเก็ต เพื่อให้การติดตั้งสามารถเกิดขึ้น และดำเนินต่อไปได้ เมื่อส่งข้อมูลไปยังซ็อกเก็ต ข้อมูลจะถูกส่งออกไปยังปลายทางโดยอัตโนมัติ โปรแกรมที่ติดต่อกันระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง สามารถอยู่บนระบบปฏิบัติการที่แตกต่างกัน แต่สามารถจะติดต่อกันได้ รวมถึงการติดต่อรูปแบบอื่น ๆ ระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง เพราะซ็อกเก็ตเป็นเครื่องมือที่ใช้ในการติดต่อระหว่างจุดสองจุดแบบสองทิศทาง

ภาษาจาวามีฟังก์ชันที่สามารถเข้าถึงข้อมูลด้วยซ็อกเก็ตที่มีอยู่ในจาวาแล้ว และในภาษาจาวายังมีคลาสที่สามารถติดต่อ และใช้งานข้อมูลจากเครื่องคอมพิวเตอร์เครื่องอื่นได้อย่างง่ายดายด้วย

ซ็อกเก็ตในการติดต่อส่งข้อมูลระหว่าง 2 ฝ่าย คือฝั่งเซิร์ฟเวอร์ (Server) และฝั่งไคลเอนต์ (Client) โดยที่ฝั่งเซิร์ฟเวอร์จะสร้างซ็อกเก็ตเพื่อรอคอยการติดต่อที่จะเกิดขึ้น ซ็อกเก็ตจะใช้ชื่อโฮสต์และหมายเลขพอร์ตเพื่อสร้างการติดต่อ ฝั่งเซิร์ฟเวอร์จะมีการใช้พอร์ตที่เป็นหมายเลขมาตรฐานในการติดต่อแต่ละแบบ โดยที่ฝั่งไคลเอนต์จะติดต่อกับฝั่งเซิร์ฟเวอร์ โดยใช้ชื่อโฮสต์ และหมายเลขพอร์ตตามที่ฝั่งเซิร์ฟเวอร์กำหนด และเมื่อฝั่งไคลเอนต์สามารถติดต่อกับฝั่งเซิร์ฟเวอร์ได้แล้ว ระบบจะสร้างซ็อกเก็ตเฉพาะสำหรับการติดต่อระหว่างฝั่งเซิร์ฟเวอร์ และฝั่งไคลเอนต์ จากนั้นฝั่งเซิร์ฟเวอร์ก็จะสามารถรอฟังการติดต่อจากไคลเอนต์ตัวอื่นต่อไป

2.4.1.11 ไคลเอนต์กับซ็อกเก็ต

ในการสร้างการติดต่อขึ้นใหม่จะต้องใช้คลาสซ็อกเก็ตเพื่อสร้างซ็อกเก็ต และระบุข้อมูลในการติดต่อได้แก่ ชื่อโฮสต์ และหมายเลขพอร์ต ซึ่งการสร้างซ็อกเก็ตเพื่อใช้ในการติดต่อจะต้องสร้างโดยใช้รูปแบบดังต่อไปนี้

```
Socket s = new Socket("Chaokhun",7);
```

และเมื่อการทำงานระหว่างแอปพลิเคชันกับซ็อกเก็ตสิ้นสุดลง จะต้องทำการยกเลิกการใช้งานซ็อกเก็ตนั้นด้วยฟังก์ชัน `close()` ดังนี้

```
s.close();
```

2.4.1.12 เซิร์ฟเวอร์กับซ็อกเก็ต

การติดต่อระหว่างเซิร์ฟเวอร์กับซ็อกเก็ตมีความคล้ายคลึงกับการติดต่อระหว่างไคลเอนต์กับซ็อกเก็ตแต่ซ็อกเก็ตของเซิร์ฟเวอร์จะเป็นผู้ฟัง และรอคอยการติดต่อจากพอร์ต เมื่อซ็อกเก็ตฝั่งไคลเอนต์พยายามจะสร้างการติดต่อที่พอร์ตฟังก์ชัน `accept()` จะถูกเรียกใช้เพื่อสร้างการติดต่อให้เกิดขึ้น

การสร้างซ็อกเก็ตฝั่งเซิร์ฟเวอร์ต้องใช้คลาส ServerSocket เพื่อสร้างซ็อกเก็ตของฝั่งเซิร์ฟเวอร์ และรอการติดต่อจากไคลเอนต์ โดยสามารถทำได้ดังนี้

```
ServerSocket ss = new ServerSocket (21);
```

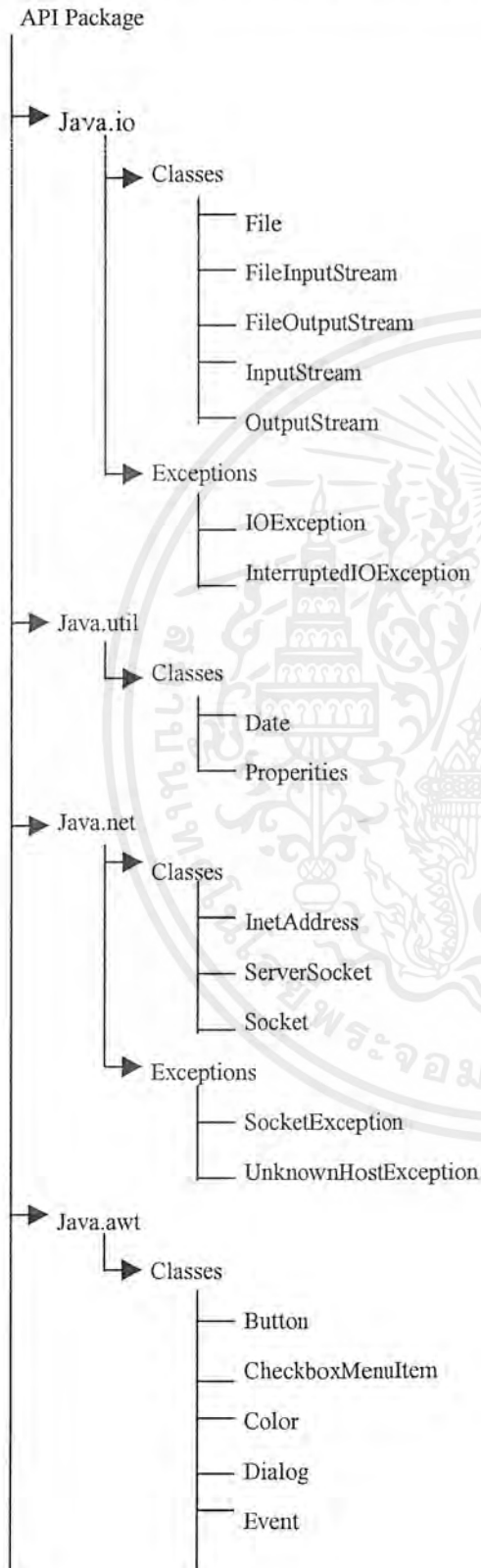
การรอคอยการติดต่อเพื่อร้องขอข้อมูลจากพอร์ตที่กำหนด เมื่อโปรแกรมพร้อมที่จะสร้างการติดต่อกับฝั่งไคลเอนต์ ต้องใช้ฟังก์ชัน accept() และมีลักษณะเช่นเดียวกับการเชื่อมต่อของไคลเอนต์กับซ็อกเก็ตตรงที่เมื่อการติดต่อสิ้นสุดลงจะต้องใช้ฟังก์ชัน close() เพื่อบอกให้รู้ว่าการติดต่อได้สิ้นสุดลง และยกเลิกการใช้ซ็อกเก็ตนั้น

การใช้งานซ็อกเก็ตของทั้งไคลเอนต์ และเซิร์ฟเวอร์ประกอบกันทำให้สามารถพัฒนาโปรแกรมที่มีการติดต่อระหว่างไคลเอนต์ และเซิร์ฟเวอร์ได้ ซ็อกเก็ตเป็นเครื่องมือขั้นสำคัญที่จะต้องรู้จัก และใช้งานมัน เพราะการทำงานของโปรโตคอล TCP/IP จะใช้ซ็อกเก็ตเป็นพื้นฐานในการติดต่อ

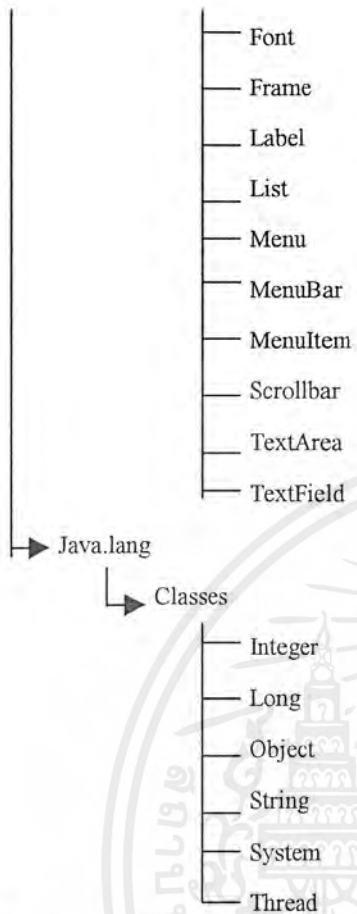


2.4.3 คลาสไลบรารีของ Java

เขียนเป็นโครงสร้าง แสดงได้ดังรูป 2.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 รูปแสดงโครงสร้างของคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 รายละเอียดของแต่ละ Class แสดงได้ดังตารางที่ 2.7

ตารางที่ 2.7 แสดงรายละเอียดขอ Class diagram java

Package	Class	Method	Paramiter	คำอธิบาย	
java.net	InetAddress	getAddress	None	ใช้คืนค่า IP Address ของ InetAddress อี้อปเจกต์	
		getHostAddress	None	ใช้คืนค่า IP Address ในรูปแบบ "%d.%d.%d.%d"	
		getLocalHost	None	คืนค่า IP Address ของโลคอลโฮสต์ (Local Host)	
		toString	None	ใช้เปลี่ยน IP Address ให้อยู่ในรูปแบบของสตริง	
	ServerSocket	ServerSocket(int)	-	สร้างซ็อกเก็ตฝั่งเซิร์ฟเวอร์โดยกำหนดหมายเลขพอร์ต	
		ServerSocket(int,int)	-	สร้างซ็อกเก็ตฝั่งเซิร์ฟเวอร์และเชื่อมต่อกับหมายเลขพอร์ตฝั่งโลคอล	
		ServerSocket (int,int,InetAddress)	-	สร้างซ็อกเก็ตฝั่งเซิร์ฟเวอร์โดยกำหนดหมายเลขพอร์ต รอฟังแบคค็อก (Backlog) และเชื่อมต่อกับหมายเลขพอร์ตฝั่งโลคอล	
		accept	None	รอฟังการติดต่อเพื่อสร้างซ็อกเก็ต	
		close	None	ปิดซ็อกเก็ต	
		getInetAddress	None	ใช้คืนค่าโลคอลแอดเดรสของซ็อกเก็ตฝั่งเซิร์ฟเวอร์	
java.io	BufferedReader	BufferedReader(Reader)	-	สร้าง InputStream อีกครั้งที่ใช้ขนาดของที่พักข้อมูลตามค่า default	
		BufferedReader(Reader, int)	-	สร้าง InputStream อีกครั้งที่ใช้ขนาดของที่พักข้อมูลตามค่าที่ระบุในพารามิเตอร์	
		ReadLine	None	ใช้อ่านสตริงเป็นบรรทัด	
		InputStream Reader	InputStreamReader (InputStream)	-	สร้าง InputStream ที่ใช้การเข้ารหัสอักขระที่เป็นค่า Default
		InputStream	read	byte[]	อ่านข้อมูลเป็นจำนวน ไบต์เท่ากับความยาวของอาร์เรย์จาก InputStream นี้ถึง อาร์เรย์ของไบต์
		InputStream	read	byte [], int,	อ่านข้อมูลเป็นจำนวน ไบต์เท่ากับค่าพารามิเตอร์ที่เป็นจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			int	เต็มตัวสุดท้าย ลงในอาร์เรย์ของไบต์ เมดอดนี้จะบล็อกจนกระทั่งมี Input เข้ามา ถ้าอาร์กิวเมนต์ตัวแรก ไม่มีค่า (null) จะเกิดการ throw NullPointerException ขึ้น
		close	None	ปิด InputStream และปล่อยทรัพยากรของระบบที่เกี่ยวข้อง
	File	File(File, String)	-	สร้างตัวอย่างของไฟล์ที่เป็นตัวแทนของไฟล์ตามชื่อที่ระบุในไคเรกทอรีที่ระบุ
		File(String)	-	สร้างตัวอย่างของไฟล์ที่เป็นตัวแทนของไฟล์ซึ่ง Pathname ก็คือ อาร์กิวเมนต์
		CanRead	None	ทดสอบว่าแอปพลิเคชันสามารถอ่านจากไฟล์ที่ระบุได้หรือไม่
		CanWrite	None	ทดสอบว่าแอปพลิเคชันสามารถบันทึกลงไฟล์ที่ระบุได้หรือไม่
		exists	None	ทดสอบว่าไฟล์มีอยู่จริงหรือไม่
		getName	None	ใช้เมื่อต้องการชื่อไฟล์
		getParent	None	ใช้เมื่อต้องการทราบ Parent path ของไฟล์
		getPath	None	ใช้เมื่อต้องการทราบ Pathname ของไฟล์
		IsDirectory	None	ทดสอบว่าไฟล์ที่แทนโดยไฟล์ออปเจกต์นี้เป็นไคเรกทอรีหรือไม่
		length	None	เมดอดนี้จะคืนค่าความยาวของไฟล์เป็นไบต์ หรือ คืนค่า 0L ถ้าไฟล์ไม่มีอยู่จริง
		List	None	เมดอดนี้จะคืนค่าอาร์เรย์ของชื่อไฟล์ในไคเรกทอรีที่ระบุ
	RandomAccessFile	RandomAccessFile (File, String)	-	สร้าง FileStream ที่มีการเข้าถึงแบบสุ่มเพื่ออ่านจากและ/หรือบันทึกลงไฟล์ที่ระบุโดยอาร์กิวเมนต์ File
		RandomAccessFile (String, String)	-	สร้าง FileStream ที่มีการเข้าถึงแบบสุ่มเพื่ออ่านจากและ/หรือบันทึกลงไฟล์ที่ระบุชื่อ
		close	None	ปิด Filestream ที่มีการเข้าถึงแบบสุ่มนี้และปล่อยทรัพยากรของระบบที่เกี่ยวข้อง
		read	byte[]	อ่านข้อมูลเป็นจำนวนไบต์เท่ากับความยาวของอาร์เรย์ byte [] ลงในอาร์เรย์ของไบต์
		seek	Long (long integer)	เซตพอยน์เตอร์ไปยังตำแหน่งออฟเซตที่ต้องการ
		write	Byte []	ทำการบันทึกข้อมูลจากไบต์อาร์เรย์ที่ระบุ โดยเริ่มต้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				ออฟเซต ลงไฟล์นี้
		write	byte [], int, int	ทำการบันทึกข้อมูลจากไบต์อาร์เรย์ที่ระบุลงไฟล์นี้ เป็นจำนวนไบต์เท่ากับอาร์กิวเมนต์ตัวสุดท้าย โดยเริ่มต้นที่ออฟเซต (อาร์กิวเมนต์ตัวที่สอง)
	PrintWriter	PrintWriter (OutputStream)	-	สร้าง PrintWriter ใหม่ จาก OutputStream ที่มีอยู่โดยไม่มีการทำให้บรรทัดเต็มโดยอัตโนมัติ (Line flushing)
		PrintWriter (OutputStream, boolean)	-	สร้าง PrintWriter ใหม่ จาก OutputStream ที่มีอยู่
		PrintWriter (Writer)	-	สร้าง PrintWriter ใหม่ โดยไม่มีการทำให้บรรทัดเต็มโดยอัตโนมัติ (Line flushing)
		PrintWriter (Writer , boolean)	-	สร้าง PrintWriter ใหม่
		Println	None	ทำการจบ(สิ้นสุด)บรรทัดนั้น ๆ
	OutputStream	OutputStream()	-	-
		close	None	ปิด OutputStream นี้ และปล่อยทรัพยากรที่เกี่ยวข้องทั้งหมด
		write	byte [], int , int	ทำการบันทึกข้อมูลลง OutputStream นี้ เป็นจำนวนไบต์เท่ากับอาร์กิวเมนต์ที่ระบุไว้ตัวสุดท้าย(ค่า len ในตัวอย่างจากไบต์อาร์เรย์) เริ่มต้นที่ออฟเซต (อาร์กิวเมนต์ตัวที่สอง)
	FileInputStream	FileInputStream (File)	-	สร้าง FileInputStream เพื่ออ่านจากไฟล์ที่ระบุในอาร์กิวเมนต์
		FileInputStream(FileDescriptor)	-	สร้าง FileInputStream เพื่ออ่านจาก File Descriptor ที่ระบุ
		FileInputStream (String)	-	สร้าง FileInputStream เพื่ออ่านจากไฟล์ที่มีชื่อของไฟล์ระบุไว้
		close	None	ทำการปิด FileInputStream นี้ และปล่อยทรัพยากรของระบบที่เกี่ยวข้องทั้งหมด
		read	byte []	อ่านข้อมูลจาก InputStream เป็นจำนวนไบต์ที่เท่ากับความยาวของ อาร์เรย์ byte []
		skip	Long (long integer)	ข้ามและทิ้งข้อมูลจาก InputStream เป็นจำนวนไบต์ตามที่ระบุ
	FileOutputStream	FileOutputStream (File)	-	สร้างไฟล์ OuputStream เพื่อบันทึกลงไฟล์ ออปเจ็กต์ ที่ระบุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		FileOutputStream (FileDescriptor)	-	สร้างไฟล์ OutputStream เพื่อบันทึกลง File Descriptor ที่ระบุ
		FileOutputStream (String)	-	สร้างไฟล์ OutputStream เพื่อบันทึกลงไฟล์ที่มีชื่อไฟล์ระบุไว้
		FileOutputStream (String , boolean)	-	สร้างไฟล์ OutputStream เพื่อบันทึกลงไฟล์ที่มีชื่อขึ้นอยู่กับระบบที่ระบุไว้
		Write	byte [] , int , int	บันทึกข้อมูลลงไฟล์ OutputStream เป็นจำนวนไบต์ตามที่ระบุในอาร์กิวเมนต์ตัวสุดท้าย จากไบต์อาเรย์ที่ระบุ เริ่มต้นที่ออฟเซต ที่ระบุในอาร์กิวเมนต์ตัวที่สอง
java.lang	Integer	parseInt	String	แปลงจากสตริงเป็นเลขจำนวนเต็ม
	Long	parseLong	String	แปลงจากสตริงเป็นเลขจำนวนเต็มแบบ Long
	Object	Object	None	สร้างตัวอย่างของคลาส Object.
	String	CharAt	Int	คืนค่าเป็นตัวเลขบอกว่าอักขรตัวนั้นอยู่ตำแหน่งใด (ตัวแรกคือ 0)
		getBytes	None	สำเนาตัวอักษรจากสตริงนี้ ลงในไบต์อาเรย์เป้าหมาย แต่ละไบต์รับ 8 บิตค่า ของตัวอักษรที่สัมพันธ์กัน
		getBytes	int,int,byte [],int	ตัวแรกสุดที่จะสำเนา, ตัวสุดท้ายที่จะสำเนา,ไบต์อาเรย์ที่จะสำเนาลงไป, ตำแหน่งเริ่มต้นของอาเรย์
		IndexOf	String	คืนค่าตำแหน่ง (เลขจำนวนเต็ม) ของตัวอักษรในสตริง ค้นจากซ้ายไปขวา
		IndexOf	String,int	คืนค่าตำแหน่งของสตริงที่ระบุที่พบตัวแรกจากซ้ายไปขวา เริ่มที่ตำแหน่งที่ระบุ
		lastIndexOf	String	คืนค่าตำแหน่ง (เลขจำนวนเต็ม) ของตัวอักษรในสตริง ค้นจากขวาไปซ้าย
		length	None	ให้ค่าความยาวของสตริง
		startsWith	String	ให้ค่าเป็นจริง ถ้าสตริงนั้นเริ่มต้นด้วยพารามิเตอร์ที่ระบุ ให้ค่าเป็นเท็จ ถ้าไม่ใช่
		substring	int	สร้างสตริงใหม่จากสตริงเดิม โดยเริ่มที่ตำแหน่งที่ระบุ
		substring	int,int	สร้างสตริงใหม่จากสตริงเดิม โดยเริ่มและสิ้นสุดตามตำแหน่งที่ระบุ (พารามิเตอร์: ตำแหน่งเริ่มต้น, ตำแหน่งสิ้นสุด)
		toString	None	แปลง Object เป็นสตริง
		ValueOf	int	แปลง integer เป็นสตริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ValueOf	Long	แปลง long เป็นสตริง
		trim	None	ให้สตริงใหม่จากสตริงเดิมที่ตัดช่องว่าง (White spaces) ออกหมด
		toUpperCase	None	แปลงสตริงจากเดิมให้เป็นอักษรใหญ่ทั้งหมด
	System	getProperty	String	แสดงคุณสมบัติเกี่ยวกับระบบ เช่นต้องการทราบเวอร์ชัน คย.การใช้ System.getProperty("java.version");
		err	-	ใช้แสดงข้อความที่เกี่ยวกับ error คย.การใช้ System.err.println("Error exception for client.");
		out	-	ใช้แสดงข้อความหนึ่งบรรทัด คย.การใช้ System.out.println("data");
		exit	int	ออกจากโปรแกรม พารามิเตอร์บ่งบอกสถานะ (เลข 0 สถานะการออกจากโปรแกรมปกติ)
	Thread	start	None	ทำให้ thread เริ่มต้นการ Execute
		run	None	ทำให้ thread ทำงาน โดยจะเขียน Code ใด ๆ ลงไปในเมธอด run ก็ได้ที่ต้องการ เนื่องจากควร Override เมธอดนี้ อยู่แล้ว
java.util	Date	Date	int , int , int , int , int	กำหนดคอปเจ็กต์ Date และเซตค่าเริ่มต้นตามพารามิเตอร์ (ปี ,เดือน, วันที่, ชั่วโมง, นาที) ตามลำดับ เดือน - ระหว่าง 0-11 , วันที่ - ระหว่าง 1-31 ชั่วโมง - ระหว่าง 0-23 , นาที - ระหว่าง 0-59
		Date	int,int,int	กำหนดคอปเจ็กต์ Date และเซตค่าเริ่มต้นตามพารามิเตอร์ (ปี, เดือน, วันที่) เวลาจะเริ่มต้นที่เที่ยงคืนตามเวลาที่ท้องถิ่น ของวันที่ระบุตามพารามิเตอร์
java.awt	Button	Button	None	สร้าง Button โดยที่ไม่มี Label
		Button	String	(พารามิเตอร์เป็น Label) สร้าง Button ที่มี Label ตามพารามิเตอร์ที่ระบุ
	CheckboxMenuItem	CheckboxMenuItem	String	(พารามิเตอร์เป็น Label) สร้าง checkbox ที่มี Label ตามพารามิเตอร์ที่ระบุ เริ่มแรกจะถูกกำหนดให้เป็น "off"
	Dialog	setTitle	String	เซต title ให้กับหน้าต่าง Dialog
		SetResizable	Boolean	เซตเป็น true ถ้าต้องการให้สามารถเปลี่ยนขนาดได้ เซตเป็น false ถ้าไม่
	Font	Font	String, int ,int	สร้างฟอนต์ใหม่ตามชื่อที่ระบุ, รูปแบบ และขนาดของฟอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

				คย. New Font("Helvetica",Font.PLAIN,10);
	Frame	Frame	String	สร้างเฟรมเริ่มแรกจะมองไม่เห็น และมี title ตามพารามิเตอร์
		SetResizable	Boolean	เซต title ให้กับเฟรม
		setTitle	String	เซตเป็น true ถ้าต้องการให้สามารถเปลี่ยนขนาดเฟรมได้ เซตเป็น false ถ้าไม่
	Label	Label	String,int	สร้าง Label ใหม่ โดยกำหนดชื่อ Label และลักษณะการวาง คย. New Label("List user",Label.LEFT); (Label .LEFT : วางติดด้านซ้าย)
		getText	None	ให้ชื่อ Label
		setText	String	กำหนดชื่อ Label
	Menu	Add	MenuItem	เพิ่ม MenuItem ที่ระบุลงไป ใน Menu นี้ ผลลัพธ์ของเมธอดนี้จะได้ MenuItem ที่ถูกเพิ่มเข้ามา
		Add	String	เพิ่ม Item ที่มี Label ที่ระบุ เข้าไปใน Menu นี้ เมธอดนี้สร้าง MenuItem เพื่อเก็บสตริง (พารามิเตอร์ : label)
	MenuBar	MenuBar	None	สร้าง MenuBar ขึ้นมาใหม่
	MenuItem	MenuItem	String	สร้าง MenuItem โดยแสดงข้อความตามพารามิเตอร์ซึ่งเป็น Label Label "-" ถูกสงวนไว้ใช้หมายถึงตัวแยกระหว่าง Menu items.
	TextArea	TextArea	String	สร้างพื้นที่สำหรับข้อความ โดยให้แสดงข้อความที่ระบุในพารามิเตอร์
		TextArea	String,int,int	สร้างพื้นที่สำหรับข้อความ โดยให้แสดงข้อความที่ระบุ และให้มีจำนวนแถว และสดมภ์ตามพารามิเตอร์
		appendText	String	ทำการต่อข้อความที่ระบุไปที่ด้านหลังของข้อความที่มีอยู่แล้ว
	TextField	TextField	String	สร้าง Text Field โดยแสดงข้อความตามพารามิเตอร์
		TextField	Int	สร้าง Text Field โดยกำหนดจำนวนตัวอักษรที่มีได้ใน 1 บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ตัวอย่างโปรแกรม FTP Server

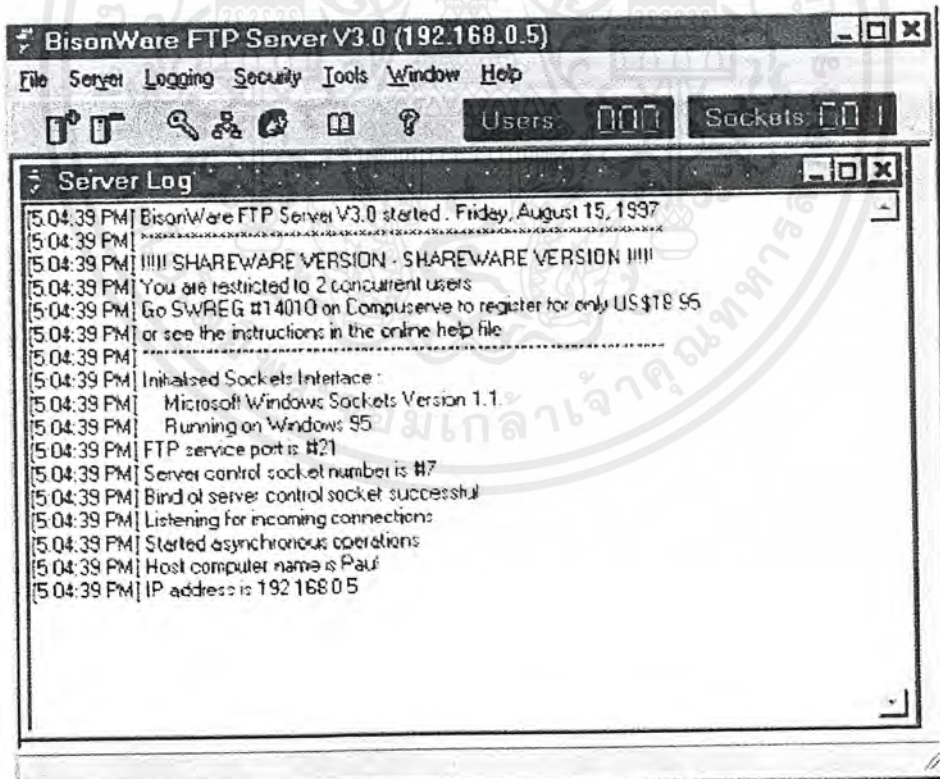
3.1 โปรแกรม BisonWare FTP Server

บริษัทผู้ผลิต: Bisonware

ความต้องการขั้นต่ำ: Windows 95/98/NT

รายละเอียด : bisonware เป็นโปรแกรม FTP Server ที่มีประสิทธิภาพในการใช้งานสูง สามารถให้มีจำนวนผู้ใช้บริการได้อย่างไม่จำกัดจะมีระบบความปลอดภัยอย่างดี มีการแสดงข้อความของผู้ใช้ มีรูปแบบการเก็บรักษาข้อความที่สมบูรณ์ และมีการดำเนินการเหมือนทาร์กไอคอน (task icon)

ลักษณะจอภาพโปรแกรม BisonWare FTP Server แสดงดังรูป



รูปที่ 3.1 แสดงหน้าจอโปรแกรม BisonWare FTP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ฟังก์ชันที่มีในโปรแกรมนี้ :

3.1.1.1 logging

ฟังก์ชันนี้ใช้กำหนดประเภทของ Logfile สามารถเลือกเก็บข้อความการเข้าออกเซิร์ฟเวอร์การส่งไฟล์เข้าออก การปิดเปิดช็อกเก็ตแล้วแต่จะเลือก

3.1.1.2 security

ฟังก์ชันนี้แบ่งเป็นสามส่วนคือ

1. General กำหนดการเข้าถึงไดเรกทอรี จำนวนของผู้ใช้บริการที่มากที่สุดที่สามารถใช้บริการเซิร์ฟเวอร์ได้
2. Usergroup กำหนดผู้ใช้แบบเป็นกลุ่ม สามารถกำหนดรายละเอียดเกี่ยวกับ Logon เช่น ให้มีการใช้ Password กำหนดเกี่ยวกับเวลาที่ให้บริการได้ กำหนดเกี่ยวกับการผ่านข้อมูล เช่น กำหนดจำนวนมากที่สุดที่สามารถอัปโหลด, ดาวน์โหลดได้ กำหนดอัตราการส่งผ่านข้อมูล สามารถกำหนดไดเรกทอรีที่ให้เข้าถึงไฟล์ กำหนดสิทธิการเข้าถึงไฟล์ได้
3. User ใช้กำหนด Password และ Username ของผู้ใช้สามารถกำหนดประเภท Logfile กำหนดจำนวนไบต์ที่ข้อมูลเข้าออก

3.1.1.3 Tool option

ฟังก์ชันนี้ใช้กำหนดตอนที่เปิดโปรแกรม จะทำการซ่อนโปรแกรมหรือไม่ กำหนดข้อความที่แสดงบนหน้าจอผู้ใช้เมื่อเปิดติดต่อกับเซิร์ฟเวอร์ได้แล้วหรือออกจากเซิร์ฟเวอร์ไปแล้ว สามารถกำหนดเสียงที่แสดงเมื่อมีการอัปโหลด, ดาวน์โหลด การติดต่อกับเซิร์ฟเวอร์เมื่อจะทำการเลิกติดต่อกับเซิร์ฟเวอร์

3.2 โปรแกรม FTPDog

บริษัทผู้ผลิต : Wicware

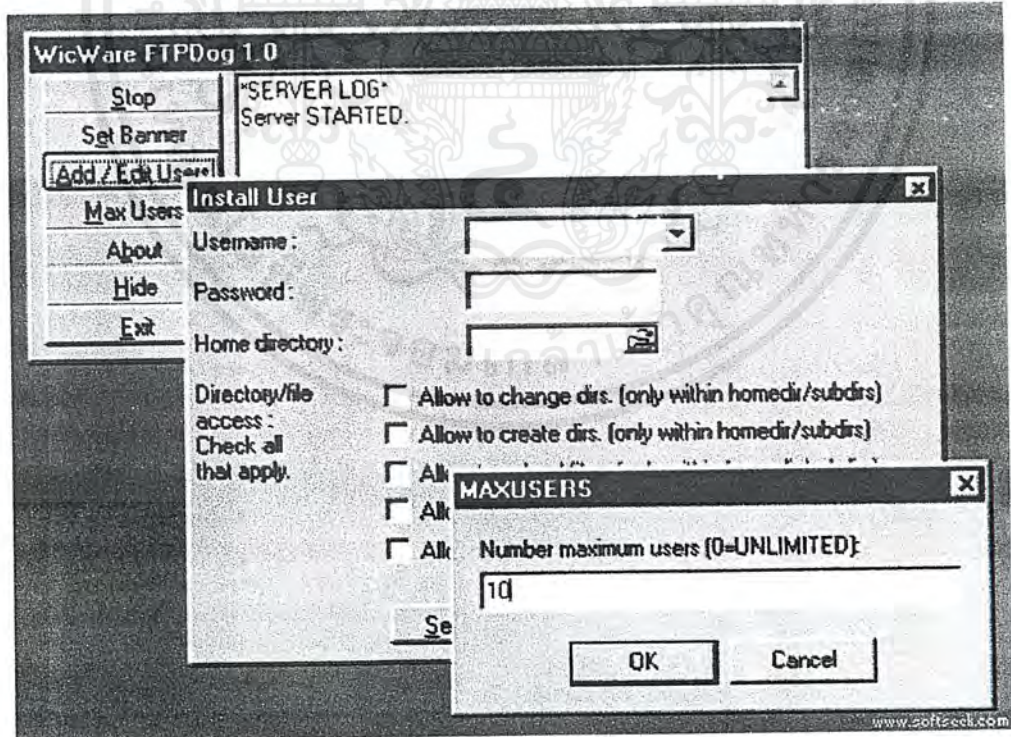
ความต้องการขั้นต่ำ : Windows 95/98/NT, 486 DX 33none.

รายละเอียด : โปรแกรมนี้จะมีฟังก์ชัน FTP ธรรมดา ซึ่งจะมี การควบคุมการเพิ่มการแก้ไข ผู้ใช้ การกำหนดจำนวนผู้ใช้ และการซ่อนโปรแกรม

3.2.1 ฟังก์ชันที่มีในโปรแกรม :

- 3.2.1.1 banner ฟังก์ชันนี้เป็นฟังก์ชันที่ใช้กำหนดข้อความที่แสดงบน FTP client เมื่อสามารถติดต่อมายังเซิร์ฟเวอร์ได้
- 3.2.1.2 add/user เป็นฟังก์ชันที่ใช้ในการกำหนด User และ Password ที่จะเข้ามาใช้เซิร์ฟเวอร์นี้ กำหนดได้เรกทอรีที่มองเห็นได้ กำหนดสิทธิผู้ใช้ได้
- 3.2.1.3 max user เป็นฟังก์ชันกำหนดจำนวนผู้ใช้ทั้งหมดที่สามารถมาใช้เซิร์ฟเวอร์ได้ในเวลาเดียวกัน
- 3.2.1.4 hide เป็นฟังก์ชันของการซ่อนไฟล์ "FTPDog"

ลักษณะจอภาพโปรแกรม WicWare FTPDog 1.0 แสดงดังรูป



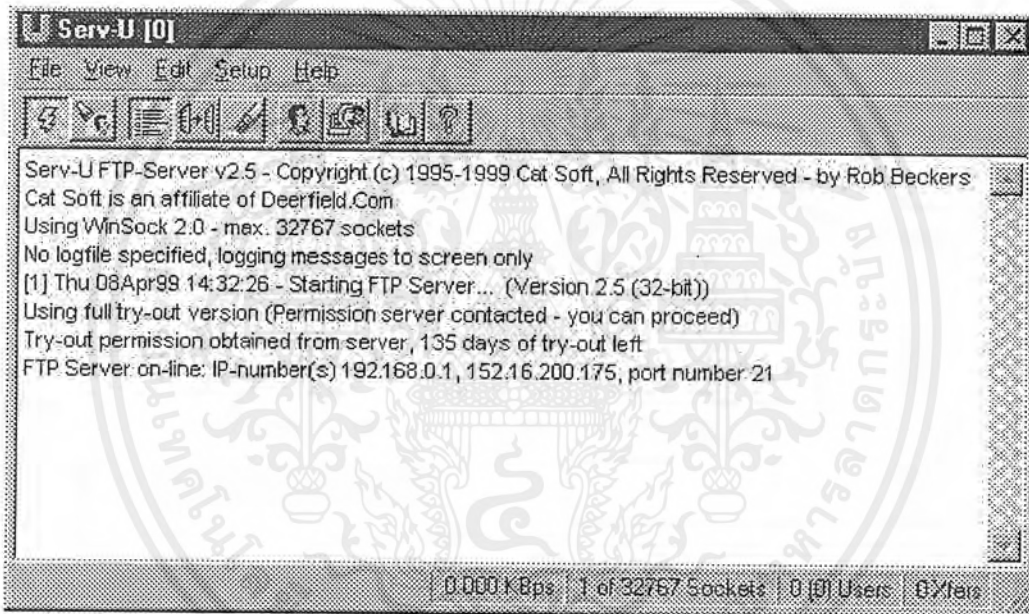
รูปที่ 3.2 แสดงหน้าจอโปรแกรม FTPDog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โปรแกรม FTP Serv-U

- บริษัทผู้ผลิต : Deerfield
- ความต้องการขั้นต่ำ : Windows 95/98/NT.
- รายละเอียด : Serv-U เป็นโปรแกรมที่มีฟังก์ชันมากมายที่แสดง การติดตั้งทำได้ง่าย มีระบบป้องกันที่แข็งแรง สามารถกำหนดอัตราการดาวน์โหลด และอัปโหลดได้ มีการเชื่อมโยงกับยูนิคซ์ มีการแสดงข้อความ และการกำหนดระยะเวลา มีรูปแบบการใช้งานสามารถซ่อนได้

ลักษณะจอภาพโปรแกรม FTP Server-U แสดงดังรูป



รูปที่ 3.3 แสดงหน้าจอโปรแกรม FTPServ-U

3.3.1 ฟังก์ชันที่มีในโปรแกรมนี้ :

- 3.3.1.1 File ในฟังก์ชันนี้จะมีฟังก์ชัน Userinfo เป็นฟังก์ชันที่ใช้ในการแสดงรายชื่อผู้ใช้เซิร์ฟเวอร์อยู่ เราสามารถให้ผู้ใช้ไม่สามารถใช้บริการได้ต่อไป (Ban) ได้โดยเลือกฟังก์ชัน Killuser
- 3.3.1.2 View ในฟังก์ชันนี้จะเป็นการเลือกให้แสดง Toolbar และ Menu image ซึ่งสองอย่างนี้มีไว้เพื่อให้สะดวกในการเรียกใช้ฟังก์ชันต่าง ๆ อีกที
- 3.3.1.3 Edit จะมีฟังก์ชันที่บันทึกข้อมูลที่ขึ้นบนหน้าจอเซิร์ฟเวอร์เป็นไฟล์ไว้เปิดดูที่หลังได้ และมีฟังก์ชัน Copy clear ให้ลบและบันทึกข้อความนั้นๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.4 Setup ภายในเมนูนี้จะประกอบด้วยฟังก์ชันย่อยดังนี้

1. FTP Server สามารถกำหนดหมายเลขพอร์ต จำนวนผู้ใช้มากที่สุด ความเร็วสูงสุดใน การส่งผ่านข้อมูลเซตการติดตั้งโปรแกรมโดยอัตโนมัติ เซตให้ทำการใช้เข้า รหัส Password ได้ กำหนดอัตราการส่งผ่านข้อมูล สามารถกำหนดให้มี IP ได้ หลาย IP address สามารถกำหนดขนาดของแคช (Cache) ที่เก็บรายการไฟล์ ข้อมูลได้
2. Logging สามารถเลือกเก็บข้อมูลได้ เช่นเลือกเก็บไฟล์ที่อัปเดต ไฟล์ที่ดาวน์โหลด เก็บ IP address, ข้อความเกี่ยวกับระบบความปลอดภัย FTP command, FTP reply และสามารถทำการเปลี่ยนฟอนต์บนหน้าจอได้
3. Setup message สามารถกำหนดข้อความที่จะขึ้นบนหน้าจอผู้ใช้ขณะที่ผู้ใช้ติดต่อมายัง เซิร์ฟเวอร์ และเมื่อผู้ใช้เลิกติดต่อกับเซิร์ฟเวอร์
4. User ใช้กำหนด User และ Password ให้ผู้ใช้กำหนดสิทธิการเข้าถึงข้อมูลไฟล์ ส่ง ไฟล์อ่านไฟล์ การอนุญาตให้ผู้ใช้เปลี่ยน Password ได้เอง กำหนดจำนวนผู้ใช้ ณ เวลานั้น กำหนด IP address ที่สามารถเข้าถึงข้อมูลได้
5. Group ทำหน้าที่คล้ายฟังก์ชัน User แต่จะทำในรูปของกลุ่ม

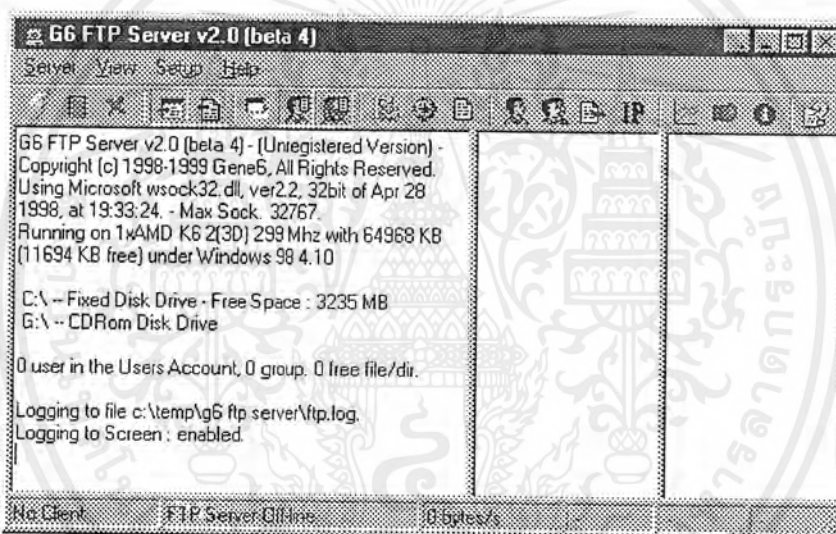
3.4 โปรแกรม G6 FTP Server

บริษัทผู้ผลิต : GeNe6

ความต้องการขั้นต่ำ : Windows 95/98/NT

รายละเอียด : G6 FTP Server เป็นโปรแกรม FTP Server ซึ่ง เป็นไปตาม RFC 959 และอื่น ๆ สามารถที่จะอัปโหลด และดาวน์โหลดสามารถเรียกดูกิจกรรมที่เกิดขึ้นใน เวลานั้นของผู้ใช้มีการกำหนดอัตรา และปรับปรุงการจัดการกลุ่มของผู้ใช้ได้ สามารถกำหนดแสดงเหตุการณ์ที่เกิดขึ้น สามารถแสดงชื่อผู้ใช้ที่เข้ามาใช้บริการบ่อยที่สุด และมีการกำหนดไคเรททอรี มีไอคอนให้ใช้ได้สะดวกขึ้น

ลักษณะจอภาพโปรแกรม G6 FTP Server V2.0 แสดงได้ดังรูป



รูปที่ 3.4 แสดงหน้าจอโปรแกรม G6 FTP Server

3.4.1 ฟังก์ชันที่มี : โปรแกรมนี้จะมีทูลบาร์เป็นไอคอนให้เรียกใช้ฟังก์ชันต่าง ๆ ได้สะดวกขึ้น

3.4.1.1 View เป็นฟังก์ชันใช้ในการเรียกข้อมูลขึ้นมาดูจะประกอบด้วยฟังก์ชันย่อย ๆ ดังนี้

1. Copy สามารถ Copy ข้อความบนหน้าจอออกมาได้
2. Clear ทำการลบข้อความบนหน้าจอทั้งหมด
3. Showuserlist ให้นำหน้าจอแสดงรายชื่อผู้ใช้
4. Showbanlist ให้นำหน้าจอแสดงรายชื่อผู้ใช้ที่ไม่ได้รับอนุญาตให้เข้ามาใช้บริการเมื่อผู้ใช้นั้นติดต่อยังเซิร์ฟเวอร์
5. Hit-o-meter แสดงรายละเอียดเกี่ยวกับไฟล์ที่ทำการอัปโหลด และดาวน์โหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Messagelist แสดงข้อความที่ส่งถึงผู้ใช้
7. Statistic จะเก็บ และแสดงข้อมูลทางสถิติโดยทั่วไปเกี่ยวกับการอัปโหลด ความถี่โหลดการติดต่อของเซิร์ฟเวอร์กับไคลเอนต์
8. Server Activity ดูกิจกรรมทั้งหมดที่เกิดขึ้นในเซิร์ฟเวอร์

3.4.1.2 Setup มีฟังก์ชันย่อยภายในให้เลือกอีกดังนี้

1. Main กำหนดรายละเอียดต่างๆในของ FTP Server เช่น ชื่อของ FTP Server จำนวนผู้ใช้ที่สามารถมีได้มากสุดในขณะเวลาหนึ่ง กำหนดหมายเลขพอร์ต การกำหนดว่าให้โปรแกรมนี้เปิดขึ้นมาเองเมื่อเปิดเครื่องคอมพิวเตอร์ใช้เซิร์ฟเวอร์ กำหนด Priority กำหนดขนาดของแคช กำหนดจำนวนไคเรกทอรีในแคช กำหนดเวลาในการเรียกฐานข้อมูลขึ้นมาใหม่อีกครั้ง
2. Event เป็นการกำหนดสิ่งที่เราต้องการแสดงให้รู้ว่ามีกิจกรรมเกิดขึ้นกับไฟล์ในเซิร์ฟเวอร์
3. Log เป็นฟังก์ชันเกี่ยวกับการเก็บข้อมูลข้อความที่แสดงกิจกรรมทั้งหมดที่เกิดขึ้นโดยสามารถเลือกที่จะเก็บได้ โดยจะเก็บไว้ในรูปของไฟล์
4. User ใช้กำหนด User และ Password ให้ผู้ใช้ กำหนดสิทธิการเข้าถึงข้อมูลไฟล์ ส่งไฟล์ อ่านไฟล์ การอนุญาตให้เปลี่ยน Password เองได้ กำหนดจำนวนผู้ใช้ในเวลานั้น กำหนด IP address ที่สามารถเข้าถึงข้อมูลได้ กำหนดให้ผู้ใช้สามารถคุยกันได้ระหว่างผู้ใช้ด้วยกันด้วย กำหนดให้ผู้ใช้สามารถใช้คำสั่ง ‘who’ ได้ กำหนดให้สามารถแสดงข้อความที่เครื่องผู้ใช้ถ้ามีการติดต่อเข้ามาหรือมีการเปลี่ยนไคเรกทอรี มีการกำหนดอัตราการส่งผ่านข้อมูล และจำนวนไคต์ที่สามารถส่งผ่านข้อมูลได้ในแต่ละครั้ง
5. Group ทำหน้าที่คล้ายฟังก์ชัน User แต่จะทำในรูปของกลุ่ม
6. File/Dir free to download เป็นฟังก์ชันที่ใช้ในกรณีที่ User ใช้ต้องการโหลดไฟล์ โดยไม่ระบุชื่อไฟล์มาโดยตรงเราสามารถทำการหาไฟล์นั้นได้โดยฟังก์ชันนี้
7. Muthome IP ฟังก์ชันนี้เป็นการกำหนดให้ใช้ Address หลาย IP ในเวลาเดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การเปรียบเทียบคุณสมบัติของโปรแกรม FTP Server

ได้ทำการศึกษาคุณสมบัติของโปรแกรม FTP Server จำนวน 4 โปรแกรม สรุปได้ดังตาราง

ตารางที่ 3.1 แสดงตารางเปรียบเทียบคุณสมบัติของโปรแกรม FTP Server

ชนิด	Down load / upload	Resume	Loging	Max User	Status/Tics/list Topten user	Hideing	Menu Image	User Pass- word	Group
BiSon Ware FTP Sever	ได้	ไม่ได้	ได้	ได้	ไม่ได้	ได้	ได้	ได้	ได้
Dog Serv-U	ได้	ไม่ได้	ไม่ได้	ได้	ไม่ได้	ได้	ไม่ได้	ได้	ไม่ได้
G 6 FTP Server	ได้	ได้	ได้	ได้	ได้	ได้	ได้	ได้	ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ขั้นตอนในการพัฒนาโปรแกรม

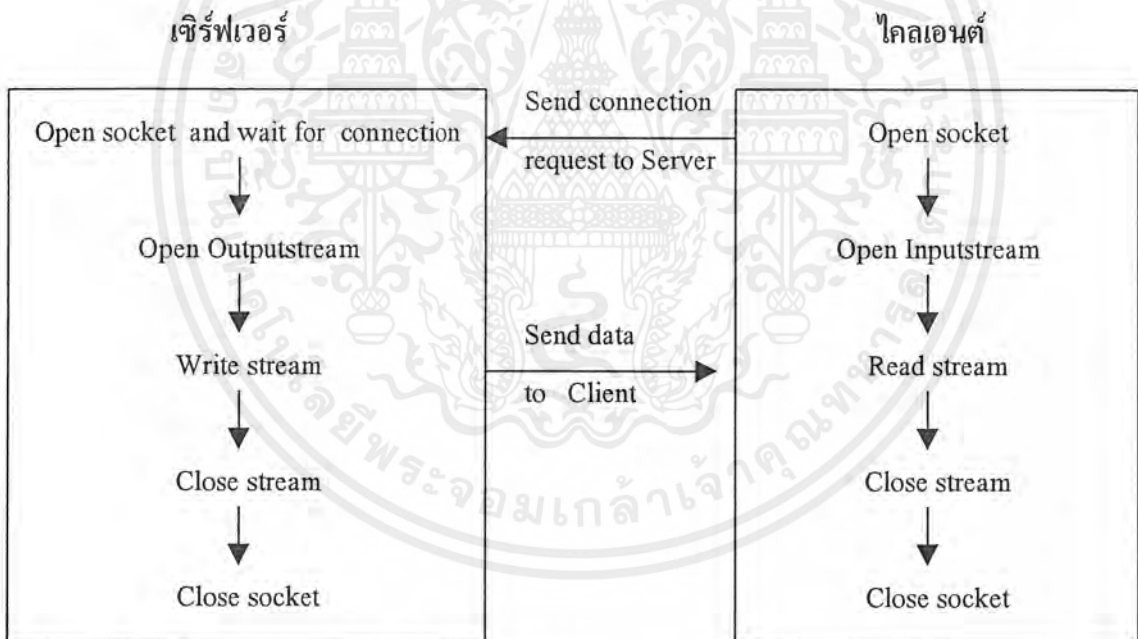
4.1 โปรแกรมทดลองการติดต่อ

การเขียนโปรแกรมเพื่อทดลองการติดต่อผ่านพอร์ตระหว่างไคลเอนต์ และเซิร์ฟเวอร์โดยการทดสอบเริ่มจาก การรันโปรแกรมสองโปรแกรมคือ

1. โปรแกรมฝั่งเซิร์ฟเวอร์
2. โปรแกรมฝั่งไคลเอนต์

4.1.1 ขั้นตอนการติดต่อผ่านพอร์ตระหว่างโปรแกรมไคลเอนต์ และโปรแกรมเซิร์ฟเวอร์

ขั้นตอนการทำงานของโปรแกรมทดลองการติดต่อแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงขั้นตอนการทำงานของโปรแกรมทดลองการติดต่อ

4.1.2 ขั้นตอน และหน้าต่างจอภาพโปรแกรมที่ทำการรันโปรแกรมติดต่อผ่านพอร์ต ชั้นที่ 1 เมื่อทำการรันโปรแกรมฝั่งเซิร์ฟเวอร์จะปรากฏหน้าต่างดังรูป

```

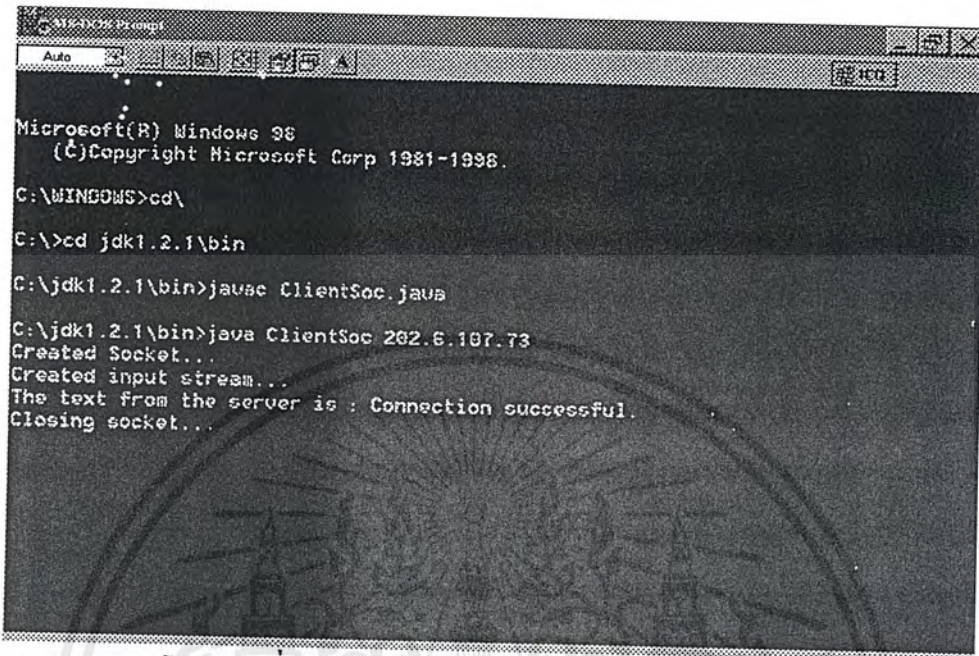
Microsoft(R) Windows 98
(C) Copyright Microsoft Corp 1981-1998.
C:\WINDOWS>cd\
C:\>cd jdk1.2.1\bin
C:\jdk1.2.1\bin>java ServerSec
Listen...

```

รูปที่ 4.2 แสดงหน้าจอโปรแกรมฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 ทำการรันโปรแกรมฝั่งไคลเอนต์จะปรากฏหน้าต่างดังรูป



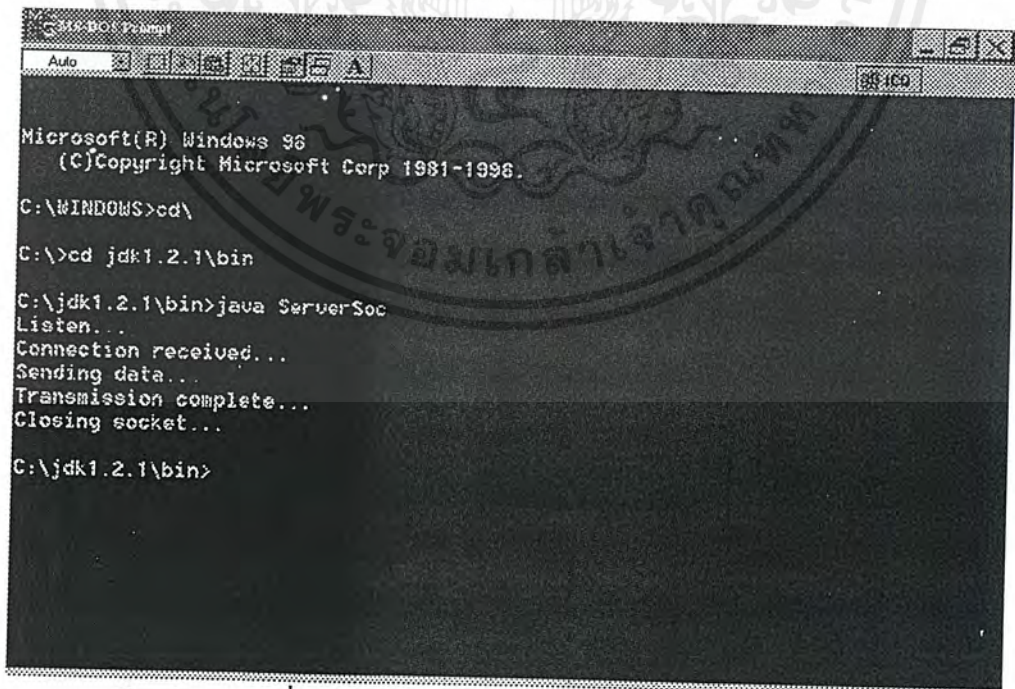
```

Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1998.

C:\WINDOWS>cd\
C:\>cd jdk1.2.1\bin
C:\jdk1.2.1\bin>javac ClientSoc.java
C:\jdk1.2.1\bin>java ClientSoc 202.6.107.73
Created Socket...
Created input stream...
The text from the server is : Connection successful.
Closing socket...
  
```

รูปที่ 4.3 แสดงจอภาพโปรแกรมฝั่งไคลเอนต์

ขั้นที่ 3 หลังจากนั้นหน้าต่างที่รันโปรแกรมฝั่งเซิร์ฟเวอร์จะปรากฏดังรูป



```

Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1998.

C:\WINDOWS>cd\
C:\>cd jdk1.2.1\bin
C:\jdk1.2.1\bin>java ServerSoc
Listen...
Connection received...
Sending data...
Transmission complete...
Closing socket...

C:\jdk1.2.1\bin>
  
```

รูปที่ 4.4 แสดงหน้าจอโปรแกรมฝั่งเซิร์ฟเวอร์หลังจากทำการรันโปรแกรมฝั่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบโปรแกรม

4.2.1 ความต้องการขั้นต่ำ :

โปรแกรมโอนถ่ายไฟล์นี้ใช้ภาษาจาวาในการเขียนจึงสามารถรันได้บนทุกแพลตฟอร์ม (platform)

4.2.2 รายละเอียดโปรแกรม :

โปรแกรมรับส่งข้อมูลฝั่งเซิร์ฟเวอร์นี้มีประสิทธิภาพในการใช้งานสามารถให้ผู้ใช้งานเข้ามาใช้บริการได้หลาย ๆ ผู้ใช้พร้อมกันมีการรองรับการต่อไฟล์เนื่องจากคอนเนกชัน Fail สามารถกำหนดจำนวนผู้ใช้ได้ และกำหนดเวลาการใช้งานของผู้ใช้บริการได้

4.2.3 ประสิทธิภาพในการใช้งานของโปรแกรม:

- สามารถรองรับคำสั่งที่เป็นมาตรฐานของ FTP โพรโตคอล บางคำสั่งใน RFC 959 ได้ (เช่น การรับไฟล์, ส่งไฟล์, การลบไฟล์, การเปลี่ยนชื่อไฟล์ เป็นต้น)
- สามารถรองรับการการต่อไฟล์ (Resume downloading/uploading) ในกรณีที่ทางส่วนไคลเอนต์เกิด Connection fail ก่อนที่จะทำการส่งไฟล์เสร็จ
- สามารถกำหนดเวลาสิ้นสุดการใช้งานของผู้ใช้เมื่อผู้ใช้ขาดการติดต่อ
- มีส่วนติดต่อกับผู้ใช้ (Interface) ทำให้สะดวกต่อการใช้งาน
- สามารถมองเห็น โฮสต์อื่นเสมือนเป็น ไคลเอนท์หนึ่ง (Multiple Host)
- สามารถกำหนดจำนวนผู้ใช้ได้
- สามารถกำหนดสิทธิผู้ใช้งานโปรแกรม Together FTP Server ได้

4.2.4 การออกแบบการกำหนดสิทธิผู้ใช้งานโปรแกรม Together FTP Server

การกำหนดสิทธิของผู้ใช้แบ่งออกเป็น 3 ประเภท ได้แก่

4.2.4.1 Admin สามารถเข้าใช้บริการทุกไคลเอนท์ได้

4.2.4.2 User สามารถเข้าใช้บริการได้ 3 ไคลเอนท์ คือ

- ไคลเอนท์ของผู้ใช้เอง
- ไคลเอนท์ \Public
- ไคลเอนท์ \Chaokhun (โดยไคลเอนท์นี้สามารถเพียงมองเห็นไฟล์ในไคลเอนท์ได้)

4.2.4.3 Anonymous สามารถเข้าใช้บริการได้เพียงไคลเอนท์ \Public

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 โปรแกรมที่ออกแบบสามารถทำงานตามคำสั่งตามตารางที่ 4.1

ตารางที่ 4.1 แสดงคำสั่งและการทำงานระหว่าง FTP Server และ FTP Clients

คำสั่งจาก user	Input ที่ส่งจาก Ftp Client	การทำงาน FTP Server	Output ที่ได้จาก FTP Server
1. connection command	-Connection command & FTP command -Acknowledge -receive Connection request	เช็คที่สามารถเปิดคอนเนกชันให้กับไคลเอนต์ได้หรือไม่ (เช่น อาจเช็คว่ามีจำนวนผู้ใช้ในระบบครบตามที่กำหนดหรือไม่) ถ้าได้จึงเปิด	Acknowledge : connection failed or established. -Hostname -Connection status
2. user name	- FTP command (USER) - Acknowledge	เก็บ Username ไว้ตรวจสอบ	- Acknowledge ว่าได้รับ
3. password	- FTP command (PASS) - Acknowledge	ส่ง Password เพื่อตรวจสอบ	- Acknowledge : connection fail - Connection status :User logged in
4. Change working directory เปลี่ยน directory (Up, down)	-Ftp command (CWD,CDUP) -Directory name	- เก็บค่า Password ไว้ตรวจสอบ จากนั้นตรวจสอบทั้ง Username และ Password	-ไดเรกทอรีที่ต้องการไฟล์ และ ไดเรกทอรีย่อยในไดเรกทอรีที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Rename	- (RNFR), Old filename/Directory - (RNTO), New filename/Directory	- เปลี่ยนชื่อไฟล์ - User กำหนดชื่อใหม่ ให้กับไฟล์ เซิร์ฟเวอร์ตรวจสอบ ว่าไฟล์ที่ต้องการ เปลี่ยนชื่อนั้นมีอยู่จริง หรือไม่ - ถ้ามีจริงจึงรับชื่อใหม่ แล้วนำไปเปลี่ยนซึ่ง ต้องไม่ซ้ำกับของเดิม ที่มีอยู่แล้วด้วย	- Acknowledge ว่า เปลี่ยนชื่อไฟล์หรือได เรกทอรีได้หรือไม่ - List ของไฟล์ และได เรกทอรีย่อยในไดเรก ทอรีปัจจุบันที่มีชื่อไฟล์ ที่ถูกเปลี่ยนไปแล้ว
6. Delete	- (DELE), Directory name, File name	- ลบไฟล์ - ทำการอัปเดต List หลังจากลบไฟล์ทิ้ง แล้ว	- List ของไฟล์และได เรกทอรีย่อยในไดเรก ทอรีปัจจุบันที่ไม่มีไฟล์ นั้นอยู่แล้ว
7. Download	- Download command, File name	- เซิร์ฟเวอร์เปิด FileInputStream แล้ว อ่านจากไฟล์ - เปิด OutputStream เพื่อส่งสิ่งที่อ่านมาให้ ไคลเอนต์	ไคลเอนต์ได้ไฟล์ที่ ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Resume	-Upload, Download command -Location file resume from remote file system -Location file resume from local file system	ไปที่ตำแหน่งของไฟล์ ที่ไคลเอนต์ส่งมาแล้ว ทำการอ่าน	ไฟล์ส่วนที่เหลือ
9. Upload	- Upload command, Filename	- เซิร์ฟเวอร์เปิด InputStream เพื่อรับ ไฟล์จากไคลเอนต์ - เปิด FileOutputStream เพื่อ บันทึกไฟล์ลงยังเครื่อง เซิร์ฟเวอร์	- เซิร์ฟเวอร์ได้ไฟล์ที่ ไคลเอนต์ Upload มา

4.2.6 อัลกอริทึมการทำงานของโปรแกรม Together Ftp Server

4.2.6.1 อัลกอริทึมการทำงานของโปรแกรม Together FTP Server

```

create ServerSocket
for loop //waiting for connection from client
    read maximum value from file TogetherIni
    if counter of client < maximum value
        accept client
        create Socket
        start thread FtpServer
    else
        send good bye message to client
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6.2 อัลกอริทึมการทำงานของโปรแกรม Together FTP Server เมื่อไคลเอนต์ทำการดาวน์โหลดไฟล์

```

Create BufferedReader          //Stream ที่มี buffer สำหรับรับข้อมูลเข้าโปรแกรม
Create PrintWriter             //Stream ของ characters สำหรับส่งข้อมูลออกไปให้
                                ไคลเอนต์

While client still connect
    read message from client
    separate message, that comes from client, to command and parameter
    If command from client is "PORT"
        translate port number to decimal
        send successful message to client
    end if "PORT"
    if command from client is "RETR" //Retrieve or get a file
        create FileInputStream //สำหรับอ่านอินพุตจากไฟล์
        create Socket //เปิดซอกเก็ตจากหมายเลขพอร์ตที่
                        ไคลเอนต์ส่งมา (เป็นการเปิดใช้ชั่วคราว)
        create OutputStream //สำหรับส่งข้อมูลออกจากโปรแกรมในรูปแบบของไบต์
    while not eof
        read data from file
        send data to client
        close OutputStream
        close FileInputStream
        close Socket
        send successful message to client
    end if "RETR"
end while

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การติดตั้งและใช้งานโปรแกรม Together Ftp Server

5.1 ขั้นตอนในการติดตั้งโปรแกรม Together Ftp Server

การติดตั้งโปรแกรม Together Ftp Server มีขั้นตอนดังนี้

1. ติดตั้งโปรแกรมจาวาเวอร์ชัน 1.2.1 (jdk 1.2.1)
2. ไปที่ Start ->run พิมพ์คำว่า Edit autoexec.dat แล้วกด Enter
3. ทำการเซต path = c:\jdk1.2.1\bin
4. สร้างไดเรกทอรีชื่อ Together Ftp Server ไว้ที่ c:\ และนำไฟล์โปรแกรม .java, .class, .ini, .log เก็บไว้ในไดเรกทอรีนี้
5. เข้าไปที่ MS-DOS Prompt ทำการรันโปรแกรม Together Ftp Server ดังตัวอย่าง
c:\Together Ftp Server\java Together
6. หลังจากนั้นจะปรากฏจอภาพโปรแกรม Together Ftp Server

5.2 การทำงานของโปรแกรม Together Ftp Server

5.2.1 เมนูการทำงานภายในโปรแกรม

ประกอบด้วย

1. File
2. Edit
3. Setup

ด้านล่างโปรแกรมจะมีช่องสามช่องคือ

1. ช่องแสดง Idle timeout ที่กำหนดไว้
2. ช่องจำนวน Max user ที่กำหนดไว้
3. ช่องแสดงจำนวน User

เรียงจากซ้ายไปขวา

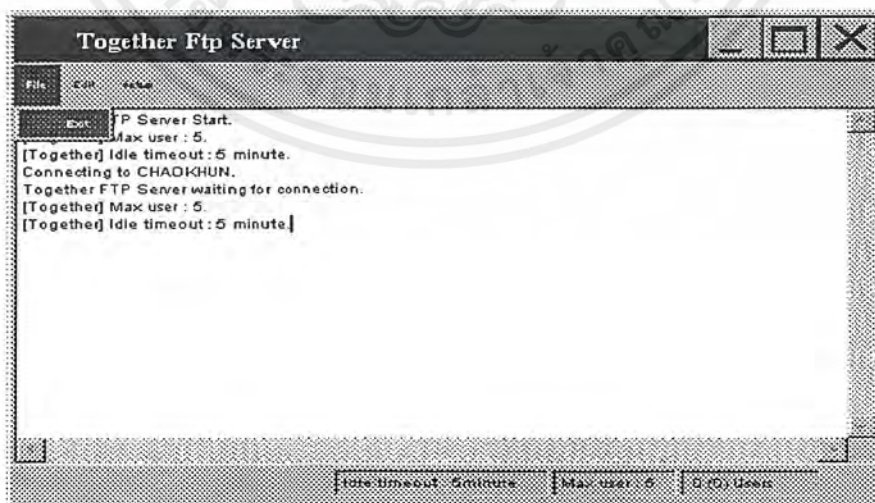
ตัวอย่างจอภาพ เมื่อเริ่มโปรแกรมแสดงได้ดังรูปที่ 5.1



รูปที่ 5.1 แสดงจอภาพโปรแกรม FTP Server

5.2.2 เมนู File

เมนู File ประกอบด้วยคำสั่ง 1. Exit เป็นคำสั่งปิดโปรแกรม FTP Server ภายในเมนู File ประกอบด้วยคำสั่ง ดังรูป 5.2



รูปที่ 5.2 แสดงจอภาพเมนูภายในเมนู File

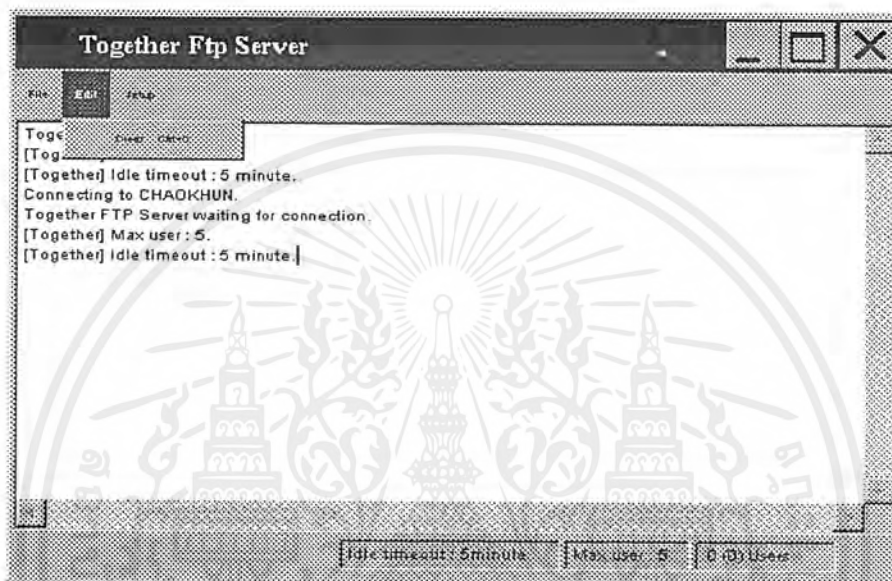
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 เมนู Edit

เมนู Edit ประกอบด้วยคำสั่ง

1. Clear ใช้สำหรับเคลียร์ข้อความที่แสดงบนจอภาพ

ภายในเมนู Edit ประกอบด้วยคำสั่งดังรูป 5.3



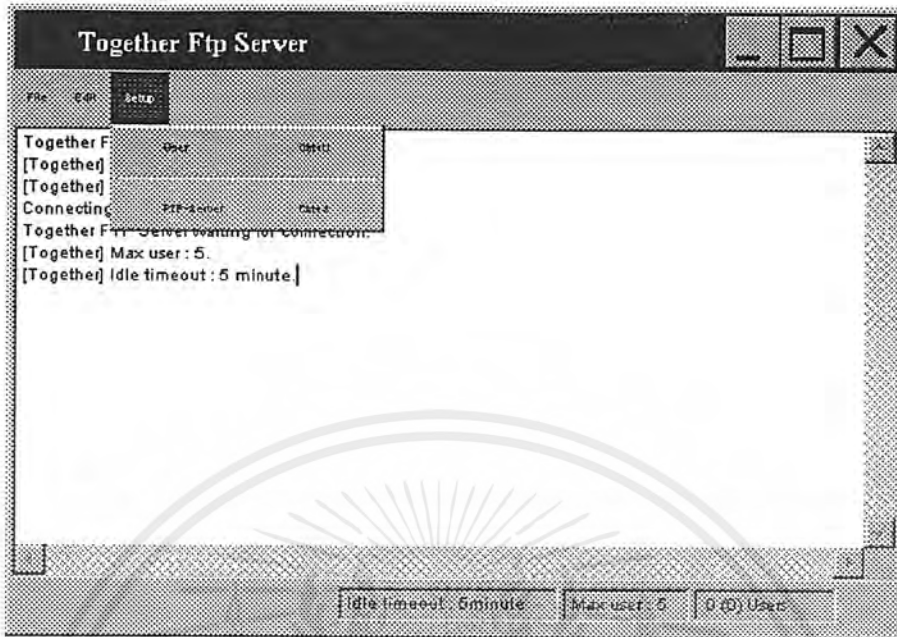
รูปที่ 5.3 แสดงจอภาพเมนูภายในเมนู Edit

5.2.4 เมนู Setup

เมนู Setup ประกอบด้วยคำสั่ง

1. User ใช้สำหรับกำหนด Username และ Password
2. FTP Server ใช้สำหรับกำหนดจำนวนผู้ใช้บริการเซิร์ฟเวอร์ (Maximum number of users) และ Idle timeout

ภายในเมนู Setup จะประกอบด้วยคำสั่งดังรูป 5.4



รูปที่ 5.4 แสดงจอภาพเมนูภายในเมนู Setup

5.2.4.1 ขั้นตอนการกำหนด Username และ Password

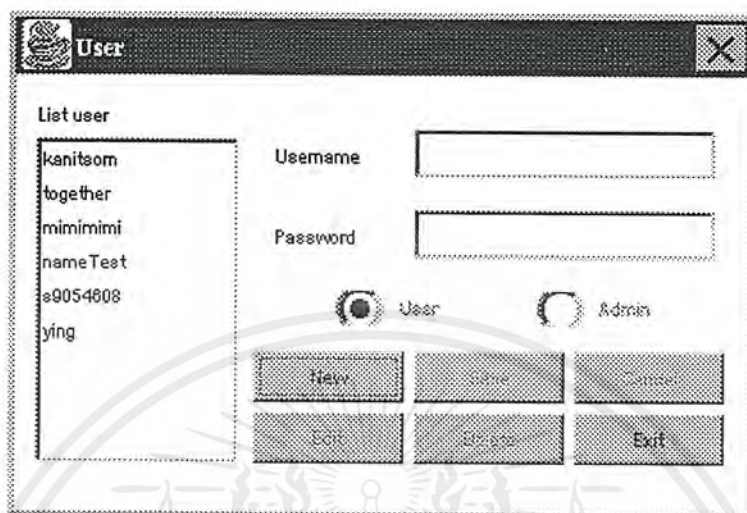
ทำการคลิก Setup -> User ดังรูปที่ 5.5



รูปที่ 5.5 แสดงตัวอย่างการเปิดหน้าต่าง User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะปรากฏหน้าต่างจอภาพดังรูป 5.6



รูปที่ 5.6 แสดงจอภาพหน้าต่าง User

5.2.4.1.1 การกำหนด Username และ Password ชุดใหม่

1. กดปุ่ม New
2. เลือกคลิก User หรือ Admin
3. ใส่ Username และ Password ที่เราต้องการลงไปในห้อง Username และ Password
4. กดปุ่ม Save (Username และ Password ที่ตั้งไว้จะปรากฏที่ช่อง List User)
5. กดปุ่ม Exit เพื่อปิดหน้าต่าง User

5.2.4.1.2 การแก้ไข Username และ Password ที่มีอยู่แล้ว

1. เลือก Username และ Password ที่เราต้องการจะแก้ไขในห้อง List User
2. กดปุ่ม Edit
3. ทำการแก้ไข Username และ Password ที่ต้องการ
4. กดปุ่ม Save
5. กดปุ่ม Exit เพื่อปิดหน้าต่าง User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4.1.3 การลบ Username และ Password

1. เลือก Username และ Password ที่เราต้องการจะลบในช่อง List User
2. กดปุ่ม Delete
3. กดปุ่ม Save
4. กดปุ่ม Exit เพื่อปิดหน้าต่าง User

หมายเหตุ กดปุ่ม Cancel เมื่อยกเลิกข้อความที่ใส่ลงไปหรือไม่ต้องการจะบันทึกข้อมูล

5.2.4.2 ขั้นตอนการกำหนด Max user และ Idle timeout

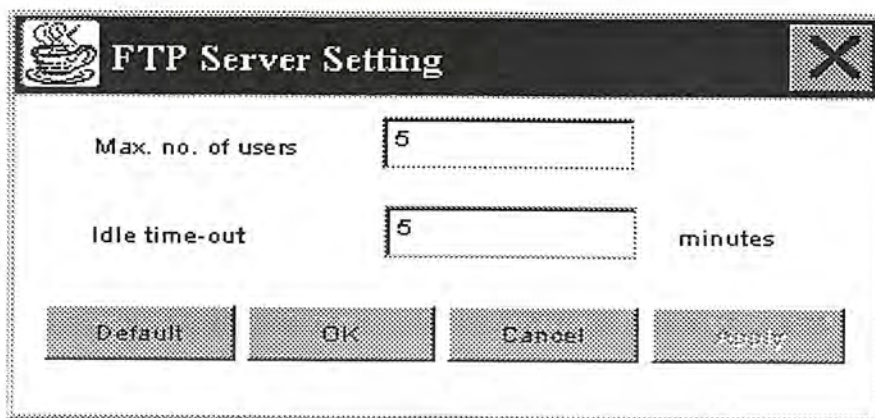
ทำการคลิก User -> FTP-Server ดังรูปที่ 5.7



รูปที่ 5.7 แสดงตัวอย่างการเปิดหน้าต่าง FTP-Server

จะปรากฏหน้าต่างจอภาพดังรูปที่ 5.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดงจอภาพหน้าต่าง FTP-Server Setting

5.2.4.2.1 ขั้นตอนการกำหนดจำนวนผู้ใช้งานสูงสุด (Max user) และ Idle timeout

1. ใส่จำนวนผู้ใช้งานสูงสุด ลงในช่อง Max. no. of users และเวลา Idle timeout ในช่อง Idle timeout
2. กด Ok

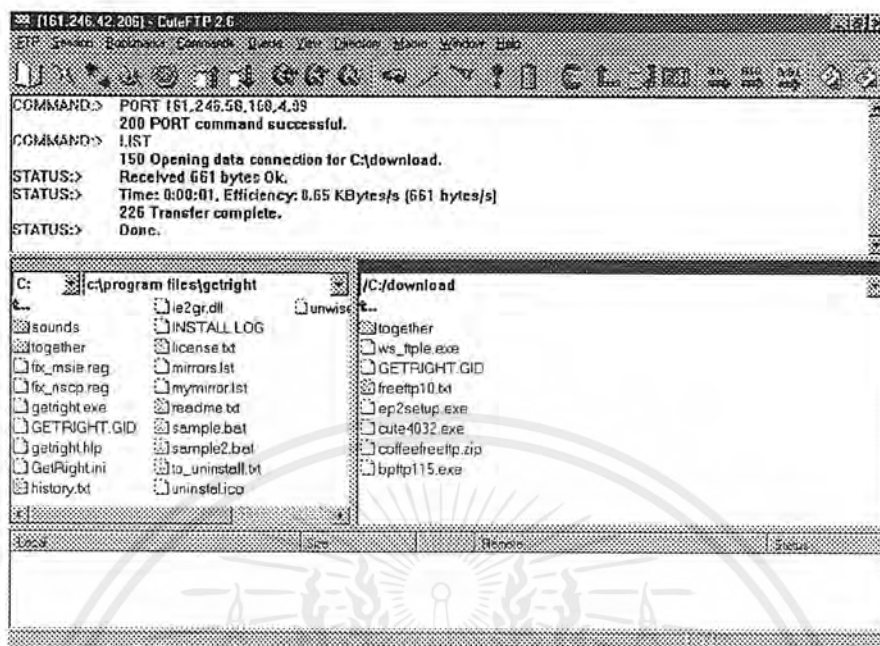
หมายเหตุ กดปุ่ม Default เป็นการกำหนดให้ Max. no. of users และ Idle timeout เท่ากับ 5

5.3 การทดลองโปรแกรม FTP ไคลเอนต์ ติดต่อกับโปรแกรม Together Ftp Server

การทดลองโปรแกรม FTP ไคลเอนต์ติดต่อกับโปรแกรม Together Ftp Server เราใช้โปรแกรมไคลเอนต์ชื่อ “Cute FTP”, “Coffeecup free FTP”, “Bullet FTP” ซึ่งแต่ละการทดลองมีผลการทดลองดังนี้

5.3.1 โปรแกรม Cute FTP version 2.8

เป็นโปรแกรมไคลเอนต์ที่ใช้ทำการติดต่อกับโปรแกรม Together FTP Server ดังรูป 5.9

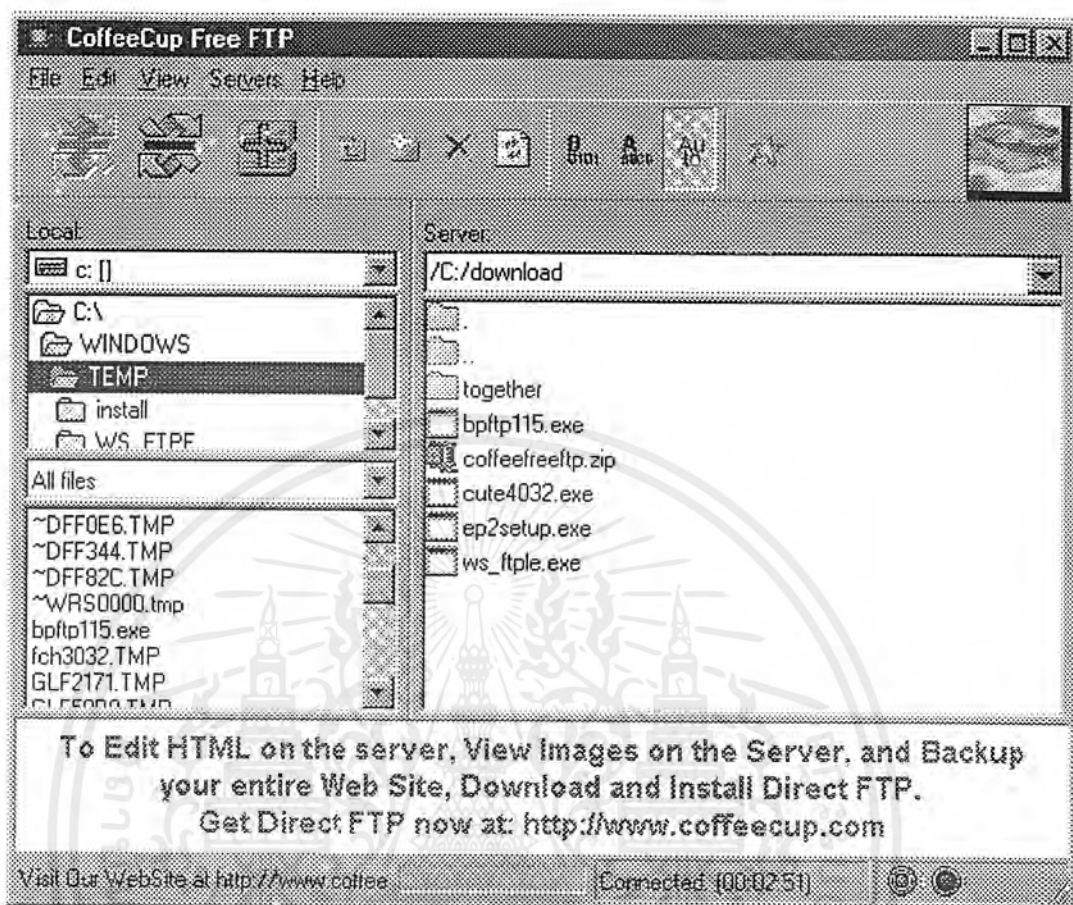


รูปที่ 5.9 แสดงจอภาพโปรแกรม Cute Ftp version 2.8 ที่ใช้ทดลอง

5.3.2 โปรแกรม Coffeecup FTP

เป็นโปรแกรมไคลเอนต์ที่ใช้ติดต่อกับโปรแกรม Together FTP Server ดังรูป 5.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

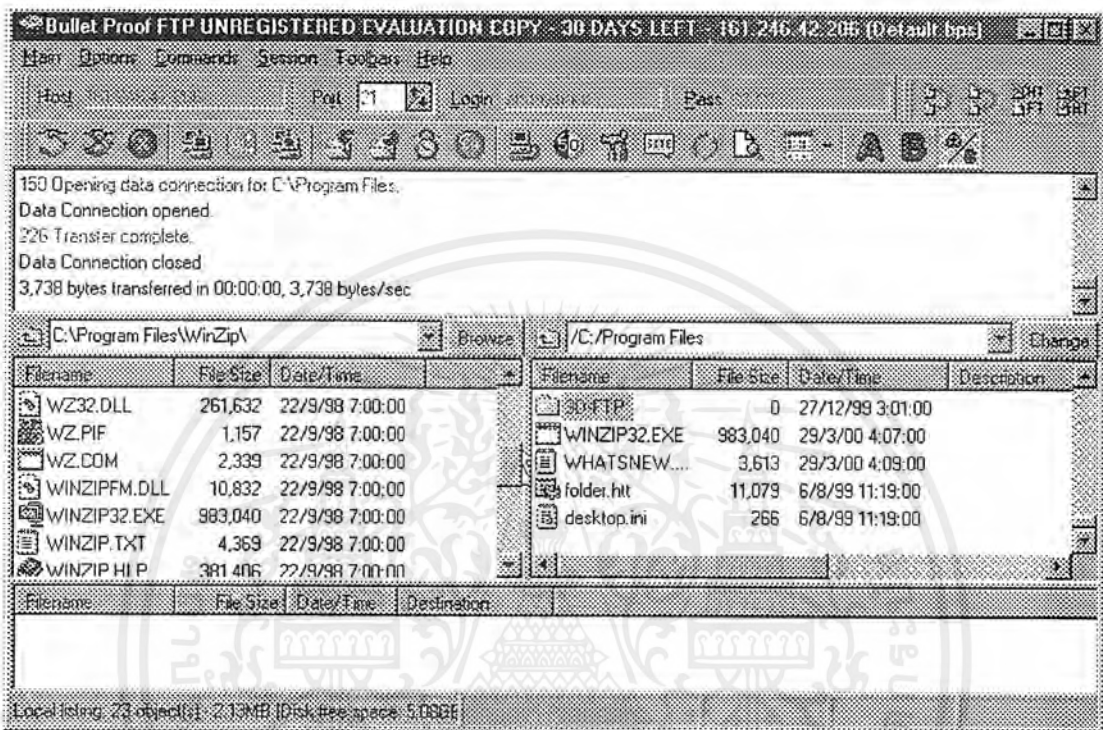


รูปที่ 5.10 แสดงจอภาพโปรแกรม Coffeecup Free Ftp ที่ใช้ทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 โปรแกรม Bullet FTP

เป็นโปรแกรมไคลเอนต์ที่ใช้ติดต่อกับ โปรแกรม Together FTP Server ดังรูป 5.11



รูปที่ 5.11 แสดงโปรแกรม Bullet Ftp ที่ใช้ในการทดลอง

5.3.4 สรุปผลการทดลองโปรแกรม FTP Client ที่ทำการติดต่อกับโปรแกรม Together Ftp

Server

ในการทดลองได้ใช้ โปรแกรม FTP Client 3 โปรแกรมในการทดลองติดต่อกับโปรแกรม Together Ftp Server ชื่อ “Cute Ftp”, “Coffee cup Free FTP”, “Bullet FTP” ได้ผลการทดลองดังตารางที่ 5.1

ตารางที่ 5.1 ตารางสรุปผลการทดลอง FTP ส่วนไคลเอนต์ที่ทดลองกับโปรแกรม Together Ftp Server

การทดลอง	CuteFtp Verion 2.8 ผลการทดลอง	Coffee cup Ftp ผลการทดลอง	Bullet Ftp ผลการทดลอง
Upload file	ได้	ได้	ได้
Download file	ได้	ได้	ได้
Delete file	ได้	ได้	ได้
เปลี่ยน directory	ได้	ได้	ได้
Up directory	ได้	ได้	ได้
Resume file	-	ได้	ได้
Rename file	ได้	ได้	ได้
Rename directory	ได้	ได้	ได้
Make directory	ได้	ได้	ได้
Delete directory	ได้	ได้	ได้

หมายเหตุ เครื่องหมาย “-” คือไม่สามารถทดลองได้เนื่องจากโปรแกรม Cute FTP version 2.8 เป็นโปรแกรมที่ไม่ได้ Register

บทที่ 6

สรุปผลและข้อเสนอแนะ

6.1 สรุปผล

ปัญหาพิเศษนี้ จัดทำขึ้นเพื่อศึกษาการทำงานของคอมพิวเตอร์ที่ติดต่อกันโดยผ่านทาง โพรโทคอล TCP/IP และเพื่อให้เข้าใจเกี่ยวกับการทำงานโปรแกรมโอนถ่ายไฟล์ข้อมูลฝั่งเซิร์ฟเวอร์ และไคลเอนต์ โดยการศึกษารวบรวมข้อมูลเพื่อใช้เป็นแนวทางในการออกแบบ และพัฒนาโปรแกรม โอนถ่ายไฟล์ข้อมูล ใช้วิธีค้นคว้าจากอินเทอร์เน็ต และหนังสือต่าง ๆ รวมทั้งทำการศึกษาโปรแกรมถ่ายโอนส่งผ่านข้อมูลฝั่งเซิร์ฟเวอร์ที่มีชื่ออยู่ในปัจจุบัน โดยมีเป้าหมายเพื่อการศึกษา และเพื่อประโยชน์ในการตัดสินใจออกแบบระบบ สำหรับการพัฒนาโปรแกรมถ่ายโอนส่งผ่านข้อมูลฝั่งเซิร์ฟเวอร์จะพัฒนาโดยใช้ภาษาจาวเวอร์ชัน 1.2.1 บนระบบปฏิบัติการวินโดวส์ 95/98/NT เริ่มต้นโดยได้ทำการทดลองสร้างโปรแกรมเปิดพอร์ตรอรับการติดต่อจากไคลเอนต์ จะพบว่าไคลเอนต์ และเซิร์ฟเวอร์สามารถติดต่อกันได้โดยผ่านพอร์ตที่เปิดไว้ ซึ่งได้นำโปรแกรมนี้นำไปเป็นพื้นฐานในการพัฒนาโอนถ่ายไฟล์ฝั่งเซิร์ฟเวอร์ได้ต่อมา

จากนั้นได้ทำการพัฒนาโปรแกรมถ่ายโอนข้อมูลฝั่งเซิร์ฟเวอร์ โดยสามารถใช้ได้กับโปรแกรม FTP ส่วนไคลเอนต์ชื่อ “Cute FTP version 2.8”, “Bullet FTP”, “Coffeecup Free FTP” สามารถรับไฟล์จากโปรแกรมฝั่งไคลเอนต์ และส่งไฟล์ไปยังโปรแกรมฝั่งไคลเอนต์ได้จริง สามารถรองรับคำสั่ง FTP พื้นฐานได้ โปรแกรมฝั่งไคลเอนต์ และโปรแกรมฝั่งเซิร์ฟเวอร์สามารถรันอยู่บนเครื่องเดียวกัน และทำงานติดต่อกันได้โดยผ่านอินเทอร์เน็ตหรือรันอยู่ต่างเครื่องโดยผ่านอินเทอร์เน็ตหรือเครือข่ายท้องถิ่นได้ ในกรณีที่การรับไฟล์ยังไม่เสร็จสมบูรณ์ในขณะที่ฝ่ายใดฝ่ายหนึ่งขาดการติดต่อ โปรแกรมฝั่งเซิร์ฟเวอร์สามารถรองรับการต่อไฟล์ได้ ทำให้ไม่เสียเวลาในการเริ่มต้นใหม่ ทั้งยังสามารถกำหนดเวลาสิ้นสุดการใช้งานของผู้ใช้เมื่อผู้ใช้ขาดการติดต่อ (Idle timeout) ได้ เมื่อหมดเวลา ผู้ใช้ก็จะถูกตัดออกจากระบบ ไม่สามารถทำงานต่อไปได้

6.2 ข้อเสนอแนะ

เนื่องจากโปรแกรมนี้พัฒนาโดยใช้จาวาเวอร์ชัน 1.2.1 พบว่าค่อนข้างยากในการทำอินเทอร์เฟซ (Interface) และการจัดการกับคลาสต่าง ๆ การใช้โปรแกรมที่มีเครื่องมือช่วยจะช่วยให้ง่ายขึ้น และประหยัดเวลามากกว่า

ในการจัดทำปัญหาพิเศษครั้งนี้ได้มีการทำ Multiple host ซึ่งได้พัฒนาให้สามารถมองเห็น โฮสต์อื่นได้ เหมือนเป็นไคลเอนต์เหมือนกันได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Sidnic Feit, **ARCHITECTURE, PROTOCOLS AND IMPLEMENTATION**, McGraw-Hill

International editions

Robin Burk, Martin Bligh, Thomas Lee, et al., **TCP/IP Blueprints**, Sams Publishing,

Ladd, Scott Robert, **Java algorithm**; New York; McGraw-Hill, c1998

Walnum, Clayton, **Java by example**, Indianapolis, IN: Que Corporation, c1996

Chan, Patrick, **Client/Server programming with Java and corba**, Reading, MA: Addison-Wesley, c1997

Gosling, James, **Java application programming interface**, Reading, MA: Addison-Wesley, c1996

Cohn, Mike, **Java developer's reference**, Indianapolis, IN: Sams.net, c1996

Laffera, Chris, **Advance Java**, Upper Saddle River, NJ: PrenticeHall, c1997

วีระศักดิ์ ชิงฉาวร , **Fundamental of java Programming Volume I**

วีระศักดิ์ ชิงฉาวร , **Fundamental of java Programming Volume II**