



การพัฒนาระบบการส่งถ่ายข้อมูล  
บนโปรโตคอล FTP และ HTTP โดยวิธีการเชิงวัตถุ  
Development of a File Transfer System  
on FTP and HTTP Protocols with Object-Oriented Method

ศุภกร รัศมีมณฑล  
Supakorn Ratsameemonthon

วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
มหาวิทยาลัยสงขลานครินทร์  
Master of Science Thesis in Computer Science  
Prince of Songkla University  
2541

๑

เลขหมู่	QA76.9.C55 ค74 2541 ค.2
Bib Key	151538

ชื่อวิทยานิพนธ์ การพัฒนาระบบส่งถ่ายข้อมูลบนโปรโตคอล FTP และ HTTP


โดยวิธีการเชิงวัตถุ

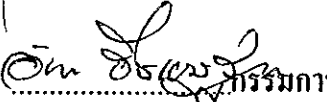
ผู้เขียน นาย ศุภกร รัตมีมณฑล

สาขาวิชา วิทยาการคอมพิวเตอร์


---

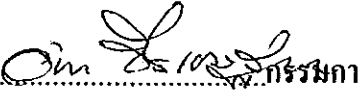
คณะกรรมการที่ปรึกษา

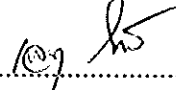
  
ดร. อ.จ. อ.จ. วิชาญ ประธานกรรมการ  
(ดร. อาจिन จิรัชิตพัฒนา)

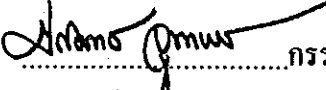
  
.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. อัจฉนา ชีรเชษฐมงคล)

คณะกรรมการสอบ

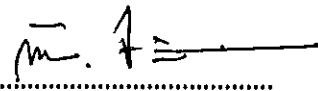
  
ดร. อ.จ. อ.จ. วิชาญ ประธานกรรมการ  
(ดร. อาจिन จิรัชิตพัฒนา)

  
.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. อัจฉนา ชีรเชษฐมงคล)

  
.....กรรมการ  
(อาจารย์ เลขา ไชยสร)

  
.....กรรมการ  
(อาจารย์ ปรีดิโมทย์ จูฑาพร)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้รับวิทยานิพนธ์ ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

  
.....  
(รองศาสตราจารย์ ดร. ก้าน จันท์พรหมมา)  
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	การพัฒนาระบบการส่งถ่ายข้อมูลบนโปรโตคอล FTP และ HTTP โดยวิธีการเชิงวัตถุ
ผู้เขียน	นาย สุภกร รัศมีมณฑล
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2541

### บทคัดย่อ

ในยุคแห่งโลกไร้พรมแดน หรือ ยุคโลกาภิวัตน์ คอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้นทุกขณะ มีการเชื่อมโยงคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่งหรือเชื่อมโยงเป็นเครือข่ายเล็ก ๆ และค่อย ๆ แดกกระจายเป็นเครือข่ายขนาดใหญ่ที่รู้จักกันในนาม “อินเทอร์เน็ต” ซึ่งมีจุดประสงค์หลักคือ การส่งถ่ายข้อมูลระหว่างกัน ช่วงเวลาที่ผ่านมามีโปรแกรมที่พัฒนาขึ้นเพื่อส่งถ่ายข้อมูลยังคงมีข้อจำกัดบางประการเช่น ไม่สามารถส่งถ่ายข้อมูลหลาย ๆ ช่องทางในเวลาเดียวกัน, มีการใช้ทรัพยากรของระบบคอมพิวเตอร์ในการส่งถ่ายข้อมูลได้อย่างไม่เต็มประสิทธิภาพ และสุดท้ายคือ ไม่สามารถส่งถ่ายข้อมูลได้อย่างต่อเนื่องภายใต้สภาวะวิกฤต เช่น ไฟฟ้าดับ หรือเครือข่ายที่ใช้งานอยู่เกิดขัดข้องในการทำงาน ในระหว่างที่ดำเนินการส่งถ่ายข้อมูล

โปรแกรมส่งถ่ายข้อมูลของงานวิทยานิพนธ์นี้ เป็นการศึกษการพัฒนาโปรแกรมบนเครือข่ายอินเทอร์เน็ตที่ทำงานบนเครื่องไมโครคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ (Microsoft Windows) โดยอาศัยเครื่องมือช่วยคือ วินโดวส์ซ็อกเก็ต (Windows Sockets) หรือ วินซ็อก (WINSOCK) ในการเชื่อมต่อกับเครือข่าย และในการส่งถ่ายข้อมูลต้องคำนึงถึงข้อกำหนดวิธีการส่งข้อมูลและการสื่อสาร ซึ่งในโครงการนี้ได้ใช้ File Transfer Protocol (FTP) และ HyperText Transport Protocol (HTTP) เป็นข้อกำหนด ส่วนการออกแบบและพัฒนาโปรแกรมได้ใช้เทคนิควิธีการโปรแกรมเชิงวัตถุ, เทคนิคกลไกสถานะ และความรู้จากทฤษฎีระบบปฏิบัติการ ซึ่งช่วยให้การพัฒนาดำเนินไปได้โดยง่าย และช่วยปรับปรุงแก้ไขข้อจำกัดต่าง ๆ ของการส่งถ่ายข้อมูลดังที่ได้กล่าวมา ส่วนสุดท้ายของการพัฒนาคือ การตรวจสอบ และประเมินประสิทธิภาพการทำงานของโปรแกรม ซึ่งจะรายงานประกอบกับวิทยานิพนธ์ฉบับนี้ด้วย

Thesis Title	Development of a File Transfer System on FTP and HTTP Protocols with Object-Oriented Method
Author	Mr Supakorn Ratsameemonthon
Major Program	Computer Science
Academic Year	1998

### Abstract

In the era of globalization, computers have much more important role to our daily life. Today these computers are interconnected to be a small network, and then the small networks are connected to be a bigger and bigger network so-called Internet. The main objective of Internet is to transfer information between each computer. For the past few decades, most programs for transferring data face some limitations; for example, the inability to transfer multiple files simultaneously, the inefficient of resource usage in transferring data and the inability to transfer data continuously under critical situations such as when power or network crashes during data transfer.

This proposed project is to study and develop a file transfer program running on the Microsoft Windows operating system and using Windows Sockets (WINSOCK) as a tool for connecting to computer networks. For data transfer, protocols for communicating and transmitting information should be taken into consideration. In this project, File Transfer Protocol (FTP) and HyperText Transport Protocol (HTTP) are employed. In addition, the object-oriented method and state machine technique are used to design and develop the program. The evaluation of the performance of the obtained program will be presented in the final part.

## กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จได้อย่างสมบูรณ์ ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่ายซึ่งผู้วิจัยรู้สึกซาบซึ้งใจ และขอกราบขอบพระคุณอย่างสูงมา ณ. โอกาสนี้ คือ

ดร. อาจिन จิรชีพพัฒนา อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษา ข้อเสนอแนะทางวิชาการ และแก้ไขปัญหาดังต่าง ๆ ตลอดจนตรวจทานวิทยานิพนธ์ให้แก่ผู้วิจัยมาโดยตลอด

ผู้ช่วยศาสตราจารย์ ดร. อัมมา วีระเชษฐมงคล อาจารย์ที่ปรึกษาร่วม ที่ได้กรุณาให้คำปรึกษาตรวจทานวิทยานิพนธ์ให้แก่ผู้วิจัย

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาช่วยตรวจและแก้ไขวิทยานิพนธ์ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ซึ่งได้สนับสนุนทุนสำหรับการศึกษาและทำวิจัย

บัณฑิตวิทยาลัย ที่ได้สนับสนุนทุนอุดหนุนการวิจัย

เจ้าหน้าที่ภาควิชาคณิตศาสตร์ทุกท่านที่ให้ความช่วยเหลือและอำนวยความสะดวกในการทำวิทยานิพนธ์

คุณพ่อ คุณแม่ พี่ชาย และน้องสาว ซึ่งได้สนับสนุนช่วยเหลือและให้กำลังใจในการทำวิทยานิพนธ์มาโดยตลอด

ศุภกร รัศมีมณฑล

## สารบัญ

	หน้า
บทคัดย่อ.....	(3)
Abstract.....	(4)
กิตติกรรมประกาศ.....	(5)
สารบัญ.....	(6)
รายการตาราง.....	(8)
รายการภาพประกอบ.....	(9)
บทที่	
1 บทนำ.....	1
1.1 วัตถุประสงค์.....	1
1.2 ขอบเขตการดำเนินงาน.....	2
1.3 ขั้นตอนและระยะเวลาการดำเนินงาน.....	2
1.4 สถานที่ และเครื่องมือที่ใช้.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 สรุปเนื้อหาในวิทยานิพนธ์.....	4
2 การพัฒนาระบบงานบนเครือข่ายอินเทอร์เน็ต.....	5
2.1 การสื่อสารบนเครือข่ายอินเทอร์เน็ต.....	5
2.1.1 แบบจำลอง OSI.....	5
2.1.2 โพรโทคอล TCP/IP.....	6
2.1.2.1 บริการในชั้นทรานสปอร์ต.....	7
2.1.2.2 บริการในชั้นเน็ตเวิร์ก.....	8
2.1.3 โพรโทคอลส่งถ่ายข้อมูล.....	12
2.1.3.1 โพรโทคอล FTP.....	12
2.1.3.2 โพรโทคอล HTTP.....	17
2.2 วินโดวส์ซ็อกเก็ต.....	21
2.2.1 แบบจำลองเครือข่ายวินโดวส์ซ็อก.....	21
2.2.2 รูปแบบการทำงานของวินโดวส์ซ็อก.....	22
3 การออกแบบ พัฒนา และการทดสอบระบบ.....	25

3.1	โครงสร้างระบบ .....	25
3.1.1	ส่วนติดต่อกับผู้ใช้.....	26
3.1.2	ส่วนการดำเนินงานของระบบ.....	27
3.1.2.1	ส่วนบริหารและจัดการงานส่งถ่ายข้อมูล .....	27
3.1.2.2	ส่วนการดำเนินงานส่งถ่ายข้อมูล .....	29
3.1.2.3	ส่วนบริการเครือข่าย .....	31
3.2	การพัฒนาาระบบ.....	32
3.2.1	ส่วนติดต่อกับผู้ใช้.....	33
3.2.1.1	การพัฒนาส่วนติดต่อกับผู้ใช้ .....	33
3.2.1.2	กระบวนการทำงานของส่วนติดต่อกับผู้ใช้ .....	34
3.2.1.3	ส่วนของโปรแกรม .....	38
3.2.2	ส่วนการดำเนินงานของระบบ.....	40
3.2.2.1	การพัฒนาส่วนการดำเนินงานของระบบ.....	40
3.2.2.2	กระบวนการทำงานของส่วนดำเนินงานของระบบ .....	44
3.2.2.3	ส่วนของโปรแกรม .....	63
3.3	การทดสอบระบบ .....	66
3.3.1	การทดสอบทั่วไป .....	66
3.3.2	การทดสอบประสิทธิภาพ .....	66
4	สรุป ปัญหา และข้อเสนอแนะ .....	71
4.1	สรุปผลการวิจัย .....	71
4.2	ปัญหาและอุปสรรค .....	71
4.3	ข้อเสนอแนะ .....	72
	บรรณานุกรม.....	73
	ภาคผนวก ก วินโดวส์ซ็อกเก็ต API.....	74
	ภาคผนวก ข คู่มือการติดตั้งและการใช้งาน.....	77
	ภาคผนวก ค การพัฒนาโปรแกรมด้วยวิธีการแบบเชิงวัตถุ.....	89
	ภาคผนวก ง ภาพประกอบของส่วนติดต่อกับผู้ใช้.....	92
	ภาคผนวก จ วิธีกลไกสถานะ .....	98
	ภาคผนวก ฉ แผนภาพกระแสข้อมูล .....	99
	ประวัติผู้เขียน .....	100

## รายการตาราง

ตาราง	หน้า
2-1 แสดงคำสั่ง FTP ที่ใช้ในงานวิทยานิพนธ์นี้ .....	14
3-1 คลาสของส่วนติดต่อกับผู้ใช้ .....	39
3-2 คลาสของส่วนการดำเนินงานของระบบ .....	63
3-3 แสดงแผนในการทดสอบโปรแกรม.....	67
3-4 แสดงอัตราเร็วในการรับส่งข้อมูล (Kb/Secs).....	68
ก-1 แสดงกลุ่มฟังก์ชันเปลี่ยนแปลงค่า .....	74
ก-2 แสดงกลุ่มฟังก์ชันฐานข้อมูล .....	75
ก-3 แสดงกลุ่มฟังก์ชันชีอกเกิด .....	75
ก-4 แสดงกลุ่มฟังก์ชันวินซ็อกขยายเพิ่มเติม.....	76
ข-1 แสดงการใช้งานรายการเมนู Edit ของวิน โคว์ฝ้าำมงานส่งถ่ายข้อมูล .....	86
ข-2 แสดงการใช้งานรายการเมนู View ของวิน โคว์ฝ้าำมงานส่งถ่ายข้อมูล .....	86
ง-1 แสดงองค์ประกอบของวิน โคว์ฝ้าำม.....	93
ง-2 แสดงองค์ประกอบของวิน โคว์ฝ้าำมกำหนดสถานะแวดล้อมของ โปรแกรม.....	95
ง-3 แสดงองค์ประกอบของวิน โคว์ฝ้าำมรับงานส่งถ่ายข้อมูล.....	96
ง-4 แสดงรายการเมนู Edit ของวิน โคว์ฝ้าำมงานส่งถ่ายข้อมูล.....	97
ง-5 แสดงรายการเมนู View ของวิน โคว์ฝ้าำมงานส่งถ่ายข้อมูล.....	97



## รายการภาพประกอบ

ภาพประกอบ	หน้า
2-1 แบบจำลอง OSI.....	6
2-2 โพรโทคอล TCP/IP เปรียบเทียบกับแบบจำลอง OSI.....	7
2-3 การเชื่อมต่อแบบสองทิศทางซึ่งเป็นบริการแบบ TCP.....	8
2-4 การเชื่อมต่อแบบทิศทางเดียวซึ่งเป็นบริการแบบ UDP.....	8
2-5 เลขที่อยู่ IP ทั้ง 5 ประเภท.....	10
2-6 แบบจำลองการทำงานของสถานีลูกข่าย-สถานีบริการ FTP.....	12
2-7 แบบจำลองการทำงานของสถานลูกข่าย-สถานบริการ HTTP.....	17
2-8 ตัวอย่างคำร้องขอและคำตอบ.....	21
2-9 แสดงการเปรียบเทียบระหว่างแบบจำลอง OSI กับ แบบจำลองวินชีอก.....	22
2-10 รูปแบบการทำงานแบบกึ่งกัน.....	23
2-11 รูปแบบการทำงานแบบอิสระ.....	24
2-12 รูปแบบการทำงานแบบอซิงโครนัส.....	25
3-1 ผังโครงสร้างระบบ.....	25
3-2 ผังโครงสร้างของส่วนติดต่อกับผู้ใช้.....	26
3-3 การทำงานของส่วนบริหารและจัดการงานส่งถ่ายข้อมูล.....	28
3-4 กลไกสถานะของการดำเนินงานในส่วนส่งถ่ายข้อมูลบน โพรโทคอล FTP.....	29
3-5 กลไกสถานะของการดำเนินงานในส่วนส่งถ่ายข้อมูลบน โพรโทคอล HTTP.....	31
3-6 โครงสร้างของส่วนบริการเครือข่าย.....	32
3-7 แผนภาพกระแสข้อมูลของส่วนติดต่อกับผู้ใช้.....	35
3-8 ผังโครงสร้างกลาสของส่วนติดต่อกับผู้ใช้.....	38
3-9 การทำงานของส่วนบริหารและจัดการงานส่งถ่ายข้อมูล.....	40
3-10 รูปแบบทั่วไปของการพัฒนาส่วนการดำเนินงานส่งถ่ายข้อมูล.....	42
3-11 แผนภาพกระแสข้อมูลของระบบส่งถ่ายข้อมูล.....	44
3-12 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS TRANSFER REQUEST.....	44
3-13 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS FTP REQUEST.....	45
3-14 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS DATA TRANSFER.....	46
3-15 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS HTTP REQUEST.....	47

3-16	แผนภาพกระแสข้อมูลของกระบวนการ PROCESS CONNECTION.....	48
3-17	ผังโครงสร้างคลาสของส่วนการดำเนินงานของระบบ.....	63
3-18	แสดงเส้นทางของการทดสอบโปรแกรม .....	67
3-19	กราฟสรุปผลการทดสอบครั้งที่ 1-4.....	68
3-20	กราฟสรุปผลการทดสอบครั้งที่ 5-8.....	69
3-21	กราฟสรุปผลการทดสอบครั้งที่ 9-12.....	69
3-22	กราฟสรุปผลการทดสอบครั้งที่ 13-16.....	69
ข-1	วินโดว์กำหนดสถานะแวดล้อมของโปรแกรม .....	77
ข-2	วินโดว์หลัก.....	79
ข-3	วินโดว์ให้ใส่ชื่อและรหัสผ่าน .....	80
ข-4	วินโดว์ใส่ชื่อแฟ้มที่ต้องการจัดเก็บ.....	80
ข-5	กรอบข้อความสอบถามยืนยันการยกเลิกการส่งถ่ายข้อมูล .....	81
ข-6	วินโดว์ให้ใส่ชื่อไคลเอนท์ที่ต้องการย้ายไปทำงาน.....	81
ข-7	วินโดว์ให้ใส่ชื่อไคลเอนท์ที่ต้องการสร้างชิ้นใหม่ .....	82
ข-8	กรอบข้อความยืนยันก่อนที่จะลบไคลเอนท์.....	82
ข-9	กรอบข้อความยืนยันก่อนที่จะลบแฟ้มข้อมูล.....	82
ข-10	วินโดว์ให้ใส่ชื่อแฟ้มอันใหม่.....	83
ข-11	วินโดว์รับงานส่งถ่ายข้อมูล .....	84
ข-12	วินโดว์ให้เลือกชื่อแฟ้มที่ต้องการส่ง.....	84
ข-13	แสดงวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล.....	85
ข-14	แสดงข้อมูลของงานส่งถ่ายข้อมูล .....	86
ค-1	แสดงคุณสมบัติของ Encapsulation.....	90
ค-2	แสดงการสืบทอดคุณสมบัติจากคลาสพื้นฐานไปยังคลาสอนุพันธ์ .....	91
ค-3	แสดงการทำงานของ Polymorphism.....	91
ง-1	วินโดว์หลัก .....	92
ง-2	วินโดว์กำหนดสถานะแวดล้อมของโปรแกรม.....	95
ง-3	วินโดว์รับงานส่งถ่ายข้อมูล.....	96
ง-4	แสดงวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล .....	97

## บทที่ 1

### บทนำ

ในการส่งถ่ายข้อมูลระหว่างเครื่องคอมพิวเตอร์จะต้องมีข้อกำหนดของวิธีการรับส่งข้อมูลในการสื่อสารระหว่างเครื่องคอมพิวเตอร์บนเครือข่าย ที่เรียกว่าโปรโตคอล (Protocol) ซึ่งในงานวิทยานิพนธ์นี้ได้ใช้โปรโตคอล FTP และ HTTP ซึ่งเป็นโปรโตคอลมาตรฐานและนำมาใช้งานมากที่สุดบนเครือข่ายอินเทอร์เน็ต (Internet) การพัฒนาโปรแกรมโดยใช้โปรโตคอลทั้งสองนี้ทำให้ได้หลากหลายวิธี สำหรับโปรแกรมของงานนี้จะใช้เทคนิคกลไกสถานะ (State Machine) มาช่วยในการพัฒนา นอกจากนี้การพัฒนาโปรแกรมได้ใช้เทคนิควิธีการโปรแกรมเชิงวัตถุ (Object-Oriented Programming) มาช่วย ซึ่งเทคนิคนี้เป็นตัวช่วยเสริมเทคนิคระบบกลไกสถานะ ทำให้การพัฒนาโปรแกรมทำได้สะดวก และการปรับปรุงโปรแกรมทำได้ง่าย

สำหรับประสิทธิภาพการทำงานของโปรแกรมเป็นสิ่งสำคัญที่จะต้องคำนึงถึง ซึ่งประสิทธิภาพของโปรแกรมระบบส่งถ่ายข้อมูลจะมีความสามารถในการส่งถ่ายข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่ให้บริการได้หลาย ๆ เครื่องภายในเวลาเดียวกัน การส่งถ่ายข้อมูลพร้อม ๆ กันนี้โปรแกรมที่ทำงานจะต้องมีคุณสมบัติที่สำคัญสองประการคือ ประการแรกความน่าเชื่อถือในแง่ของการรับหรือส่งข้อมูลจะต้องมีความผิดพลาดน้อยที่สุดหรือไม่มีเลย ประการที่สองคือ การเข้าใช้ทรัพยากรของระบบคอมพิวเตอร์เพื่อทำการส่งถ่ายข้อมูลจะต้องดำเนินไปอย่างมีประสิทธิภาพ จากคุณสมบัติทั้งสองนี้ได้บรรจุไว้ในโปรแกรมที่ได้พัฒนาในงานวิทยานิพนธ์นี้

#### 1.1 วัตถุประสงค์

ออกแบบและพัฒนาโปรแกรมระบบส่งถ่ายข้อมูลโดยใช้วิธีการโปรแกรมแบบเชิงวัตถุ ซึ่งโปรแกรมที่พัฒนาขึ้นนี้จะทำงานบนเครื่องไมโครคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์รุ่นที่ 3.1 (Microsoft Windows 3.1) เพื่อทำหน้าที่เป็นสถานีลูกข่าย (Client) ที่ส่งถ่ายข้อมูลบนโปรโตคอล FTP และ HTTP ผ่านทางเครือข่ายอินเทอร์เน็ตกับสถานีบริการ (Server)

## 1.2 ขอบเขตการดำเนินงาน

1. พัฒนาโปรแกรมส่งถ่ายข้อมูลที่สามารถส่งถ่ายข้อมูลกับเครื่องคอมพิวเตอร์ที่เป็นเครื่องบริการได้หลาย ๆ เครื่องพร้อมกันในเวลาเดียวกัน

2. พัฒนาการบริหารและจัดการการส่งถ่ายข้อมูลที่เกิดขึ้นพร้อม ๆ กันจากที่กล่าวไว้ในหัวข้อที่ 1 ให้เข้าใช้ทรัพยากรของระบบคอมพิวเตอร์ที่กำลังใช้งานอยู่อย่างมีประสิทธิภาพ

3. พัฒนาโปรแกรมถ่ายข้อมูลที่สามารถกู้คืน (Recovery) การทำงานของโปรแกรมให้กลับมาดำเนินการส่งถ่ายต่อจากเดิมได้ เมื่อเกิดเหตุการณ์วิกฤต (ต.ย.เช่น ไฟฟ้าดับ) ขึ้นและเหตุการณ์ได้ดำเนินผ่านไปแล้ว

## 1.3 ขั้นตอนและระยะเวลาการดำเนินงาน

### • ขั้นตอนการดำเนินงาน

1. ดำเนินการศึกษาในหัวข้อต่าง ๆ ดังนี้

- ศึกษาวิธีการพัฒนาโปรแกรมสื่อสารบนเครือข่ายอินเทอร์เน็ตที่ใช้วินโดวส์ซ็อกเก็ต

- ศึกษาข้อกำหนดของโปรโตคอล FTP และ HTTP ที่จะนำมาใช้งาน

- ศึกษาข้อจำกัดในการส่งถ่ายข้อมูลของโปรแกรมส่งถ่ายข้อมูลที่เคยมีผู้พัฒนาขึ้นก่อนหน้านี้

2. วิเคราะห์ ทดสอบ และหาวิธีในการพัฒนาระบบ

3. ออกแบบ อินพุตและเอาต์พุต และโมดูล (module) การทำงานเพื่อเตรียมใช้กับเทคนิควิธีการ โปรแกรมแบบเชิงวัตถุ

4. พัฒนาระบบงาน

5. ทดสอบ แก้ไข และปรับปรุงระบบ

6. จัดทำรายงาน

- ระยะเวลาการดำเนินงาน

ใช้เวลาทั้งสิ้นประมาณ 1 ปี 3 เดือน ดังรายละเอียดตามตารางต่อไปนี้

ขั้นตอนการพัฒนาโปรแกรม การส่งถ่ายข้อมูลบนโปรโตคอล FTP / HTTP (2539-41)																
	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
ศึกษา																
วิเคราะห์																
ออกแบบ																
พัฒนาโปรแกรม																
ทดสอบ																
จัดทำรายงาน																

#### 1.4 สถานที่ และเครื่องมือที่ใช้

- สถานที่

ห้องปฏิบัติการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

- เครื่องมือที่ใช้

1. เครื่องคอมพิวเตอร์ IBM Compatible ใช้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์

##### รุ่นที่ 3.1

2. ตัวแปลภาษา Borland C++ 4.5

3. โปรแกรมช่องทางการสื่อสารวินโดวส์ซ็อกเก็ตรุ่นที่ 1.1 สำหรับการพัฒนา

- WINSOCK.H - header file
- WINSOCK.LIB - import library
- WINSOCK.DLL - run time component
- WINSOCK.HLP - help file

4. โปรแกรมอรรถประโยชน์

- Object-Oriented Design Tool (With Class CASE tool) - สำหรับการออกแบบ
- Trumpet Winsock Version 3.0 - สำหรับเชื่อมต่อกับเครือข่าย
- TracePlus/Winsock Version 3.00 - สำหรับตรวจสอบการสื่อสารกับเครือข่าย

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำโปรแกรมที่พัฒนาไปใช้ประกอบการเรียนการสอนในเนื้อหาบางส่วนของวิชาที่เกี่ยวข้องกับการสื่อสารข้อมูลบนเครือข่ายได้
2. ผู้วิจัยมีความเข้าใจในการดำเนินงานและการพัฒนาโปรแกรมที่ทำงานบนเครือข่ายมากขึ้น
3. ได้โปรแกรมส่งถ่ายข้อมูลที่ทำงานได้อย่างมีประสิทธิภาพ และน่าเชื่อถือ

### 1.6 สรุปเนื้อหาในวิทยานิพนธ์

บทที่ 1 กล่าวถึง วัตถุประสงค์ ขอบเขต ขั้นตอน และระยะเวลาในการทำวิทยานิพนธ์ และประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 กล่าวถึง การสื่อสารบนเครือข่ายอินเทอร์เน็ต และวินโดวส์ซีเอกเก็ต

บทที่ 3 กล่าวถึง การออกแบบและพัฒนาระบบส่งถ่ายข้อมูล

บทที่ 4 จะเป็นการทดสอบการทำงานของโปรแกรม การสรุปผลของงานวิทยานิพนธ์ และ ข้อเสนอแนะต่าง ๆ

## บทที่ 2

### การพัฒนาระบบงานบนเครือข่ายอินเทอร์เน็ต

การพัฒนาระบบงานบนเครือข่ายอินเทอร์เน็ต ผู้พัฒนามีความจำเป็นต้องทำความเข้าใจถึงข้อกำหนดของวิธีการรับส่งข้อมูลในการสื่อสารระหว่างเครื่องคอมพิวเตอร์บนเครือข่าย ที่เรียกว่าโปรโตคอล ซึ่งจะกล่าวถึงโปรโตคอลที่ใช้บนเครือข่ายอินเทอร์เน็ตไว้ในหัวข้อแรก หัวข้อถัดมาจะอธิบายถึงโปรแกรมวินโดวส์ซ็อกเก็ต ซึ่งเป็นโปรแกรมที่เป็นช่องทางการสื่อสารสำหรับการพัฒนาระบบงานบนเครือข่ายอินเทอร์เน็ตที่ทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์

#### 2.1 การสื่อสารบนเครือข่ายอินเทอร์เน็ต

แบบจำลองการเชื่อมต่อระหว่างระบบเปิด “โอเอสไอ” (OSI) (Open System Interconnect Network Reference Model) จะช่วยทำให้มองเห็นการสื่อสารบนเครือข่ายได้อย่างชัดเจนขึ้น การติดต่อสื่อสารจากคอมพิวเตอร์เครื่องหนึ่งไปยังคอมพิวเตอร์อีกเครื่องหนึ่งบนเครือข่ายคอมพิวเตอร์จำเป็นต้องใช้โปรโตคอลในการสื่อสารระหว่างกัน สำหรับอินเทอร์เน็ตนั้นจะใช้โปรโตคอล TCP/IP เป็นหลักและเมื่อมองในระดับโปรแกรมประยุกต์ การสื่อสารระหว่างโปรแกรมประยุกต์จากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกคอมพิวเตอร์อีกเครื่องหนึ่ง ก็มีความจำเป็นต้องใช้โปรโตคอลเช่นเดียวกัน สำหรับระบบงานของวิทยานิพนธ์นี้ได้เลือกใช้งานสองโปรโตคอลด้วยกันคือ โปรโตคอล FTP และ โปรโตคอล HTTP

##### 2.1.1 แบบจำลอง OSI

แบบจำลอง OSI จะแสดงถึงการมองในระดับบนของระบบเครือข่ายทั้งหมด แบบจำลอง OSI จะลดความซับซ้อนของเครือข่ายไปสู่องค์ประกอบที่สำคัญและแสดงให้เห็นว่าองค์ประกอบเหล่านี้เข้ากันได้เป็นโครงสร้างสามัญ โดยการใช้โครงสร้างดังกล่าวทำให้สามารถสร้างข้อกำหนด, แสดงแนวคิดของเครือข่าย และ บัญญัติคำศัพท์เฉพาะขึ้นได้

International Standards Organization (ISO) ได้จัดตั้งอนุกรรมการขึ้นในปี 1977 เพื่อออกแบบสถาปัตยกรรมเครือข่ายมาตรฐาน จุดมุ่งหมายของอนุกรรมการคือ กำหนด และระบุชุดของหน้าที่ทั่วไปที่สามารถควบคุมการติดต่อสื่อสารของเครือข่ายระหว่างคอมพิวเตอร์ อนุกรรมการได้สร้างแบบจำลอง OSI ขึ้นดังแสดงในภาพประกอบ 2-1 จากรูปสี่เหลี่ยมด้านซ้ายของภาพประกอบคือ โปรโตคอลสแตค (Stack) หรือกลุ่มของโปรโตคอลที่สัมพันธ์กันได้ถูกอุปนัยเป็นสแตคของกล่องหลาย ๆ ใบที่นำมาแสดงในแบบจำลองอย่างเป็นลำดับชั้น กล่าวคือในระดับชั้นที่ต่ำจะสนับสนุนลำดับชั้นที่สูงกว่าด้วยบริการที่จัดเตรียมไว้

<u>Layer#</u>	<u>Layer Name</u>	<u>Description of Layer Services</u>
7	Application	Provides the user interface and implements the services to complete the application's purpose (e.g., file transfer, electronic mail, etc.).
6	Presentation	Formats data (e.g., encrypts or decrypts, compresses or decompresses, or converts between different representations) so its meaning is preserved.
5	Session	Opens a session (a "virtual connection") between two network hosts, controls the session between the two endpoints, then closes the session.
4	Transport	Transfers data reliably or unreliably. Reliable transfer involves creation of a connection (a "virtual circuit") and then destruction of the connection on completion.
3	Network	Address packets and routes them to provide end-to-end communications between two network hosts, through intermediate hosts.
2	Data Link	Creates and controls the physical links of communication between two endpoints, multiplexes links competing for a shared interface, "packetizes" a bitstream.
1	Physical	Provides the electrical connections to a transmission medium, with which one bit can be sent and recognized as one bit by the receiver.

ภาพประกอบ 2-1 แบบจำลอง OSI

( ที่มา: Comer, Douglas., 1991 : 143 )

แบบจำลอง OSI แบ่งโปรโตคอลออกเป็น 7 ลำดับชั้น แต่ละลำดับชั้นจะมีบทบาทหน้าที่ของตัวเองในการให้บริการตามชื่อของลำดับชั้นที่กำหนดไว้ แต่ละลำดับจะเข้าถึงบริการของลำดับชั้นที่ต่ำกว่าโดยใช้จุดเชื่อมต่อซึ่งอยู่ที่ขอบเขตบนของกล่องของลำดับชั้นนั้น ๆ จุดเชื่อมต่อนี้จะประกอบด้วยฟังก์ชันบริการให้เรียกใช้หลายฟังก์ชัน

การทำงานของแบบจำลอง OSI จะทำงานแบบลดหลั่นกันลงมา ลำดับชั้นที่อยู่เหนือกว่าจะขึ้นกับชั้นที่อยู่ต่ำลงมา การทำงานในลำดับชั้นที่อยู่บนจะส่งงานให้กับลำดับชั้นล่างโดยผ่านจุดเชื่อมต่อ และความเกี่ยวพันระหว่างลำดับชั้นจะเกิดขึ้นเฉพาะส่วนของงานบริการที่ลำดับชั้นที่ต่ำกว่ากระทำขึ้นเท่านั้น

### 2.1.2 โปรโตคอล TCP/IP

โปรโตคอล TCP/IP ( Transmission Control Protocol/Internet Protocol ) เป็นโปรโตคอลหลักสองโปรโตคอลที่รู้จักกันดีในนามของ Internet Suite TCP/IP ได้จัดเตรียมการเชื่อมโยงเครือข่ายคอมพิวเตอร์หลาย ๆ ระบบ โดยผ่านทางสื่อสัญญาณชนิดต่าง ๆ



ความคงทนของ TCP/IP คือความเข้ากันได้กับโครงสร้างพื้นฐานของเครือข่ายในระดับชั้นต่ำลงมา จากความจริงที่ว่าโปรโตคอลนี้มีพื้นฐานอยู่บนมาตรฐานเปิดกว้าง จึงส่งผลให้โปรโตคอลนี้เป็นที่รู้จักกัน ชุดของโปรโตคอล TCP/IP ที่สัมพันธ์กับแบบจำลอง OSI ดังแสดงในภาพประกอบ 2-2

Layer#	Layer Name	TCP/IP Protocol Suite
7	Application	Application
6	Presentation	
5	Session	
4	Transport	TCP and UDP
3	Network	IP, ARP, and ICMP
2	Data Link	Data Link
1	Physical	Physical

ภาพประกอบ 2-2 โปรโตคอล TCP/IP เปรียบกับแบบจำลอง OSI

( ที่มา: Comer, Douglas., 1991 : 146 )

จากภาพประกอบ 2-2 ชุดของโปรโตคอล TCP/IP เมื่อเปรียบเทียบกับแบบจำลอง OSI ในระดับชั้นทรานสปอร์ต (Transport) และในระดับชั้นเน็ตเวิร์ก (Network) รายละเอียดของโปรโตคอลทั้งสองมีดังนี้

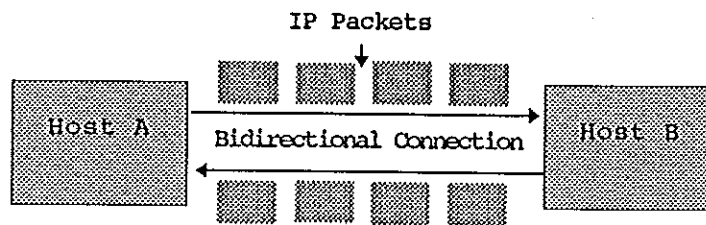
#### 2.1.2.1 บริการในชั้นทรานสปอร์ต

ประกอบด้วยโปรโตคอลที่ให้บริการ 2 โปรโตคอลด้วยกันดังนี้

- Connection-oriented service: TCP

บริการนี้อาจเรียกอีกอย่างหนึ่งว่า Stream Service บริการนี้ออกแบบมาสำหรับการส่งถ่ายข้อมูลในปริมาณมาก ๆ และมีการจัดเตรียมการตรวจสอบความผิดพลาด, การกู้ข้อมูล และการจัดเรียงข้อมูล ก่อนที่จะมีการส่งข้อมูลจะต้องจัดตั้งการเชื่อมต่อแบบสองทิศทางระหว่างสถานีต้นทาง และปลายทาง ดังแสดงในภาพประกอบ 2-3 เพื่อให้เกิดความน่าเชื่อถือได้ว่าสถานีปลายทางจะได้รับข้อมูลที่ปราศจากความผิดพลาดและได้รับอย่างเป็นลำดับ

การส่งข้อมูลจะแบ่งข้อมูลที่ส่งออกเป็นเซกเมนต์ (Segment) โดยแต่ละเซกเมนต์จะประกอบด้วยค่าตรวจสอบผลรวม (Checksum) และสารสนเทศที่บ่งบอกถึงลำดับข้อมูล (Sequence Number) จากนั้นจะนำเซกเมนต์ข้อมูลไปหุ้มด้วย IP แพคเกจ (Packet) ซึ่ง IP จะนำไปใช้ในการกำหนดทิศทางการส่งไปยังสถานีปลายทาง ค่าตรวจสอบผลรวมและสารสนเทศบอกลำดับเซกเมนต์ข้อมูล จะนำไปใช้โดยสถานีปลายทางในการตรวจสอบความถูกต้องและลำดับของข้อมูล เมื่อใดก็ตามที่ IP แพคเกจสูญหาย สถานีต้นทางจะส่งข้อมูลเดิมซ้ำตามความเหมาะสม ถึงแม้ว่าจะเกิดการขัดจังหวะจากการส่งข้อมูลซ้ำของสถานีต้นทางก็จะไม่ส่งผลกระทบต่อารรับข้อมูลของโปรแกรมประยุกต์ที่สถานีปลายทาง

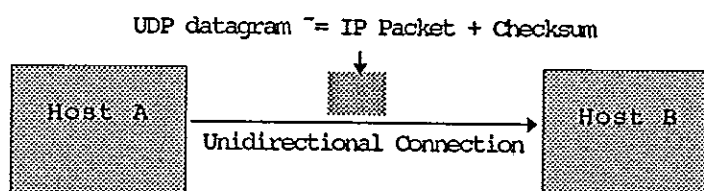


ภาพประกอบ 2-3 การเชื่อมต่อแบบสองทิศทางซึ่งเป็นบริการแบบ TCP

#### • Connectionless Service:UDP

บริการนี้อาจเรียกอีกอย่างหนึ่งว่า Datagram Service บริการนี้ออกแบบมาสำหรับการส่งข้อมูลในจำนวนน้อย ๆ ในทิศทางเดียวดังแสดงในภาพประกอบ 2-4 ก่อนที่จะทำการส่งข้อมูลไม่จำเป็นต้องมีการจัดการเชื่อมต่อระหว่างสถานีงาน และไม่เข้มงวดในการตรวจสอบความผิดพลาดในการส่งถ่ายข้อมูล โปรแกรมประยุกต์ที่เหมาะสมกับการใช้บริการ UDP จะต้องทำงานในลักษณะร้องขอไปและรอการตอบรับกลับมา (Request/Response)

การตรวจสอบข้อมูลที่ได้รับมาจะใช้กลไกง่าย ๆ คือ การตรวจสอบค่าผลรวมจะไม่มี การส่งข้อมูลซ้ำถ้าหากว่าข้อมูลเกิดสูญหาย หรือผิดพลาด หรือลำดับของแพคเกจข้อมูลผิดพลาด เมื่อไรก็ตามที่เกิดความผิดพลาดขึ้น UDP จะไม่แก้ไขแต่จะปล่อยให้เป็นการรับผิดชอบของโปรแกรมที่อยู่ในระดับ โปรโตคอลที่อยู่สูงขึ้นไป



ภาพประกอบ 2-4 การเชื่อมต่อแบบทิศทางเดียวซึ่งเป็นบริการแบบ UDP

### 2.1.2.2 บริการในชั้นเน็ตเวอร์ก

ประกอบด้วยโปรโตคอลที่ให้บริการ 3 โปรโตคอลด้วยกันดังนี้

- โปรโตคอล IP

โปรโตคอล IP ที่ใช้และพูดถึงในงานวิทยานิพนธ์นี้เป็นโปรโตคอล Version 4 บริการของโปรโตคอลนี้เป็นกุญแจสำคัญของการทำงานบนเครือข่าย โปรโตคอลนี้จะจัดเตรียมการให้เลขที่อยู่ (Addressing), การกำหนดเส้นทาง (Routing), การแบ่งแยก (Fragmentation) และการรวบรวมแพ็คเกจ (Assembly) การกำหนดเลขที่อยู่ให้กับแพ็คเกจจะทำให้สามารถส่งแพ็คเกจไปยังสถานี่งานอื่น ๆ ได้ การกำหนดเส้นทางหมายถึง การเลือกเส้นทางสำหรับการส่ง การแบ่งแยกและการรวบรวมแพ็คเกจทำให้สามารถส่งแพ็คเกจขนาดใหญ่ผ่านไปยังเครือข่ายที่ใช้ขนาดแพ็คเกจที่เล็กได้ และด้วยความสามารถนี้ทำให้สามารถเชื่อมต่อเครือข่ายที่มีสื่อในการทำงานที่ต่างกันได้ ซึ่งเป็นจุดสำคัญของโปรโตคอล TCP/IP

อย่างไรก็ตาม บริการนี้จะเป็นสิ่งที่พรางตาต่อโปรแกรมประยุกต์ เมื่อโปรแกรมประยุกต์ส่งแพ็คเกจจะไม่สามารถกำหนดทิศทางกรส่ง หรือการแบ่งแยกแพ็คเกจได้ มีเพียงแต่อุปกรณ์เชื่อมโยง หรือเราเตอร์ (Router) ที่เชื่อมต่อระหว่างสถานี่งาน ซึ่งทำหน้าที่กำหนดทิศทางและแบ่งแยกแพ็คเกจด้่าจำเป็น และเมื่อโปรโตคอลสแตกฝ่ายรับได้รับก็จะรวบรวมแพ็คเกจอีกครั้งเพื่อส่งข้อมูลต่อให้แก่โปรแกรมที่อยู่ในระดับโปรโตคอลที่อยู่สูงขึ้นไปใช้ในการทำงานต่อไป

ขนาดของเลขที่อยู่ IP ที่ปรากฏในแพ็คเกจประกอบด้วย 4 ไบต์ ซึ่งมีการใช้ในโปรแกรมเป็นสัญลักษณ์ของชุดอักขระตัวเลขที่เรียกว่า Dotted-decimal ค่าของ 4 บิตในลำดับสูงใช้ในการแบ่งแยกเลขที่อยู่เป็น 5 กลุ่มคือ A,B,C,D และ E แสดงคังภาพประกอบ 2-5 แต่ละเลขที่อยู่ IP ถูกจัดตั้งเป็นส่วนของ netid และ hostid และ กลุ่มของเลขที่อยู่ IP นี้จะใช้ในการแบ่งขนาดของ netid และ hostid ดังที่กล่าวมา

MSB	LSB					
0	netid(7)      hostid(24)		class A			
1	0	netid(14)      hostid(16)	class B			
1	1	0	netid(21)      hostid(8)	class C		
1	1	1	0	multicast address	class D	
1	1	1	1	0	reserved	class E

ภาพประกอบ 2-5 เลขที่อยู่ IP ทั้ง 5 ประเภท

( ที่มา: Comer, Douglas., 1991 : 62 )

เครือข่ายส่วนใหญ่จะใช้ Subnetbits ในการเพิ่มขนาดของเครือข่ายในกลุ่มเลขที่อยู่แบบ A และ B ทำให้ใกล้เคียงกับกลุ่ม C ซึ่งมีเครือข่ายย่อยที่แบ่งโดยเราเตอร์มาก จะทำให้มีจำนวนโฮสต์ (Host) น้อยลง และสามารถแบ่งแยกปัญหาที่เกิดขึ้นได้ง่าย สำหรับขอบเขตของเลขที่อยู่ในกลุ่ม D มีไว้สำหรับเลขที่อยู่แบบมัลติคาสต์ (Multicast) กล่าวคือ จะมีโฮสต์มากกว่าหนึ่งสถานีที่สามารถรับแพคเกจได้พร้อม ๆ กัน คล้ายกับการส่งแพคเกจโดยใช้เลขที่อยู่แบบกระจายข่าว (Broadcast) เพียงแต่ว่าในแบบมัลติคาสต์จะดีกว่า

- Internet Control Message Protocol : ICMP

โปรโตคอลนี้ดูประหนึ่งว่าอยู่ในระดับชั้นโปรโตคอลทรานสปอร์ต เมื่อมันทำงานอยู่เหนือโปรโตคอล IP แต่มันก็ถูกจัดให้อยู่ในระดับชั้นโปรโตคอลเน็ตเวิร์ก เช่นเดียวกับโปรโตคอล IP ในทางปฏิบัติโปรโตคอลนี้ เป็นโปรโตคอลสนับสนุนไม่ใช่โปรโตคอลส่งถ่ายข้อมูล การทำงานของโปรโตคอลนี้จะใช้ส่งการควบคุม, ข้อผิดพลาด และข่าวสารสนทนาระหว่างเครือข่ายโฮสต์

ชนิดของข่าวสารของโปรโตคอล ICMP มีอยู่มากมาย มีอยู่ส่วนน้อยที่ไม่ส่งผลกระทบต่อโปรแกรมประยุกต์ ข่าวสารของโปรโตคอลส่วนใหญ่จะส่งผลกระทบต่อโปรแกรมอย่างชัดเจน ถ้าหากโปรแกรมประยุกต์เลือกใช้ซ็อกเก็ตชนิด UDP หรือ TCP

- Address Resolution Protocol : ARP

อุปกรณ์เชื่อมต่อเครือข่ายจะมีเลขที่อยู่เชิงกายภาพ (Physical Address) ที่บรรจุอยู่ในหน่วยความจำถาวร (ROM) ของแผงวงจรเครือข่ายมาจากโรงงานโดยตรง โปรโตคอลนี้จะทำหน้าที่เข้าคู่ (Mapping) จากเลขที่อยู่ IP ไปยังเลขที่อยู่เชิงกายภาพ

โปรโตคอล TCP/IP จะต้องค้นหาเลขที่อยู่เชิงกายภาพของสถานีปลายทาง ก่อนที่จะส่งแพคเกจออกไป ดังนั้นเมื่อไรก็ตามที่โปรแกรมประยุกต์เริ่มติดต่อกับสถานีปลายทางผ่านสื่อที่ใช้โปรโตคอล ARP การทำงานของโปรโตคอลนี้จะเกิดขึ้น เมื่อกระบวนการค้นหาเลขที่อยู่เชิงกายภาพเสร็จสิ้นลง การทำงานของโปรแกรมประยุกต์ก็จะดำเนินต่อไป เลขที่อยู่เชิงกายภาพของอุปกรณ์เชื่อมต่อไม่เป็นสิ่งจำเป็นสำหรับโปรแกรมประยุกต์ มันเป็นหน้าที่ของโปรโตคอลสแตค TCP/IP ในการดูแลเรื่องเหล่านี้

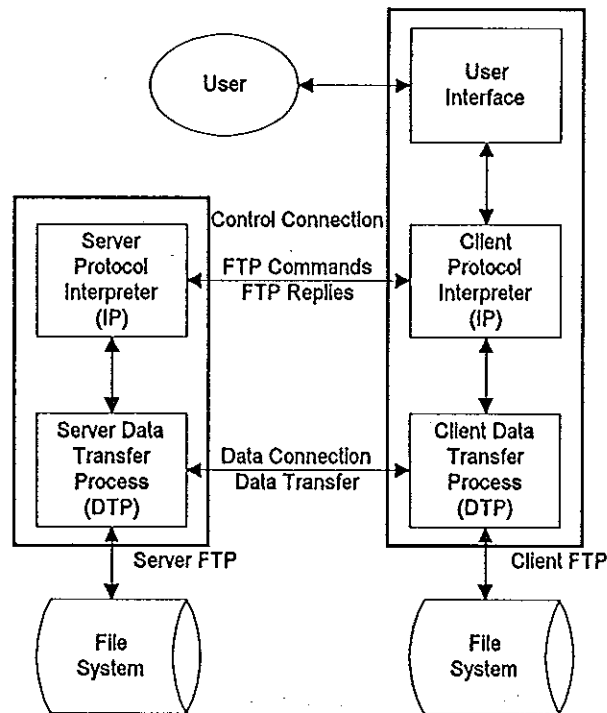
ประการสุดท้าย โปรโตคอล ARP จะไม่ใช้กับ Serial Line Internet Protocol (SLIP) และ Point-to-Point Protocol (PPP) เพราะโปรโตคอลเหล่านี้จะไม่ใช้เลขที่อยู่เชิงกายภาพ เนื่องจากการเชื่อมต่อแบบจุดต่อจุดผ่านทางสายนำสัญญาณแบบ Serial แต่การใช้เลขที่อยู่ IP ยังคงมีอยู่

### 2.1.3 โพรโทคอลส่งถ่ายข้อมูล

จากแบบจำลอง OSI การติดต่อสื่อสารในระดับโปรโตคอลชั้นบนสุด หรือระดับโปรแกรมประยุกต์นั้น ในงานวิทยานิพนธ์นี้ได้ใช้ โปรโตคอล FTP และ โปรโตคอล HTTP

#### 2.1.3.1 โพรโตคอล FTP

กลไกการทำงานของ โปรโตคอลนี้สามารถอธิบายโดยใช้แผนภาพของแบบจำลองดังแสดงในภาพประกอบ 2-6



ภาพประกอบ 2-6 แบบจำลองการทำงานของ สถานีลูกข่าย-สถานีบริการ FTP

( ที่มา: Postel, J. B. and Reynolds, J. K., 1985 )

จากภาพประกอบ 2-6 สถานีลูกข่ายมีการเชื่อมต่อกับสถานีบริการ 2 ช่องทางด้วยกัน ช่องทางแรกเป็นช่องเชื่อมต่อควบคุม (Control Connection) ส่วนช่องทางที่สองเป็นช่องเชื่อมต่อข้อมูล (Data Connection) ช่องทางการควบคุมมีไว้สำหรับส่งคำสั่ง (Command) ไปยังสถานีบริการและใช้รับผลลัพธ์ของคำสั่ง (Replies) ที่สถานีบริการตอบกลับมา ตัวแก้ไขข้อมูลจริง ๆ จะส่งถ่ายทางช่องเชื่อมต่อข้อมูล

เมื่อพิจารณาในส่วนของสถานีลูกข่ายจะประกอบด้วยโมดูลการทำงาน 3 โมดูลด้วยกัน คือ โมดูลแรกจะใช้ในการติดต่อกับผู้ใช้ โมดูลที่สองใช้ในการแปลโปรโตคอล และ โมดูลสุดท้ายใช้ในการส่งถ่ายข้อมูล

โมดูลติดต่อกับผู้ใช้สร้างขึ้นภายใต้สภาวะแวดล้อมแบบกราฟฟิคเพื่อรับคำสั่งจากผู้ใช้ และแสดงผลลัพธ์ของคำสั่ง

โมดูลแปลโปรโตคอลทำหน้าที่แปลคำสั่ง และควบคุมโมดูลส่งถ่ายข้อมูล โมดูลนี้จะสร้างคำสั่งที่ได้จากผู้ใช้และส่งคำสั่งไปยังสถานีบริการ เมื่อสถานีบริการได้รับจะประมวลผลคำสั่ง และส่งผลลัพธ์ในรูปแบบของรหัสตอบกลับไป สถานีลูกข่ายจะนำรหัสตอบรับที่ได้มาแปลความหมาย และปฏิบัติคำสั่งโปรแกรมที่สอดคล้องกับรหัสตอบไป

โมดูลส่งถ่ายข้อมูลจะทำงานติดต่อกับระบบแฟ้มข้อมูลเพื่อทำการอ่านเขียนข้อมูลที่ส่งไปหรือรับมาผ่านทางช่องทางส่งถ่ายข้อมูล โดยโมดูลนี้อยู่ภายใต้การควบคุมของโมดูลแปลโปรโตคอลว่าจะให้เริ่มต้นทำงานเมื่อไรและจะให้ทำการส่งหรือรับข้อมูล เมื่อโมดูลนี้ทำการส่งถ่ายข้อมูลเสร็จสมบูรณ์แล้ว จะส่งผลลัพธ์ของการทำงานให้แก่โมดูลแปลโปรโตคอลต่อไป

#### • ช่องเชื่อมต่อควบคุม

สถานีลูกข่ายจะสร้างการเชื่อมต่อแบบ TCP ไปยังสถานีบริการผ่านทางพอร์ต (Port) มาตรฐานหมายเลข 21 การเชื่อมต่อระหว่างกันนี้เรียกว่าช่องเชื่อมต่อควบคุม สถานีลูกข่ายจะใช้ช่องทางนี้ในการส่งคำสั่งและรับผลลัพธ์ของคำสั่ง ผลลัพธ์ของบางคำสั่งอาจจะมีหลายบรรทัดอยู่ในรูปของรหัส ASCII แต่มีจำนวนไม่มากนัก

สำหรับการส่งถ่ายข้อมูลปริมาณมากเช่น แฟ้มข้อมูล หรือแฟ้มข้อมูลที่เก็บชื่อแฟ้มภายในไดเรกทอรี (Directory) จะส่งถ่ายผ่านทางช่องเชื่อมต่อข้อมูล

#### • ช่องเชื่อมต่อข้อมูล

คำสั่งของ FTP มีเพียง 4 คำสั่งเท่านั้นที่ต้องสร้างช่องเชื่อมต่อข้อมูล คือ STOR (ส่งแฟ้มข้อมูล), APPE (ส่งแฟ้มข้อมูลไปเขียนต่อท้าย), RETR (รับแฟ้มข้อมูล) และ LIST (รับแฟ้มข้อมูลที่เก็บชื่อแฟ้มที่อยู่ในไดเรกทอรี) โดยแบบฉบับแล้ว สถานีลูกข่ายจะใช้คำสั่ง PORT ส่งสารสนเทศของซ็อกเก็ตที่ได้จากการเปิดใช้ช่องเชื่อมต่อข้อมูล (เลขที่อยู่ IP และ หมายเลขพอร์ต) ไปยังสถานีบริการ จากนั้นสถานีบริการจะทำการเชื่อมต่อมายังเลขที่อยู่ และหมายเลขพอร์ตตามที่ระบุไว้ สถานีลูกข่ายจะทำตัวเป็นสถานีบริการชั่วคราวที่จะคอยรับการสร้างช่องเชื่อมต่อข้อมูล และทำการส่งถ่ายข้อมูลทราบใดที่ช่องเชื่อมต่อข้อมูลยังเชื่อมต่อกันอยู่

การปฏิบัติการใช้ช่องเชื่อมต่อข้อมูลจะเป็นหน้าที่ของฝ่ายที่ส่งข้อมูล เมื่อการส่งถ่ายข้อมูลเสร็จสมบูรณ์แล้ว หรืออาจกล่าวได้ว่าสำหรับคำสั่ง STOR หรือ APPE สถานีถูกถ่ายจะเป็นฝ่ายยุติ และสำหรับคำสั่ง RETR หรือ LIST จะเป็นหน้าที่ของสถานีบริการ

- คำสั่งของ FTP

คำสั่งจะอยู่ในรูปของรหัส ASCII ข้อความตามข้อกำหนดของ RFC 854 และแต่ละข้อความจะจบด้วยรหัส ASCII หมายเลข 10 และ 13 ตามลำดับ สำหรับชนิดของตัวพิมพ์ของคำสั่งตามข้อกำหนด RFC 959 จะไม่คำนึงถึง ซึ่งอาจใช้ได้ทั้งตัวพิมพ์ใหญ่ หรือเล็กผสมกันก็ได้

ตาราง 2-1 จะแสดงคำสั่งกลุ่มหนึ่งจากคำสั่งทั้งหมดของ โปรโตคอลที่ใช้ในงานวิทยานิพนธ์นี้

ตาราง 2-1 แสดงคำสั่ง FTP ที่ใช้ในงานวิทยานิพนธ์นี้

คำสั่ง	ความหมาย
ABOR	ยกเลิกคำสั่งที่ทำงานค้างอยู่
APPE {filename}	ส่งแฟ้มข้อมูลไปเขียนต่อท้ายแฟ้มข้อมูลที่สถานีบริการ
CWD {directory}	ย้ายการทำงานไปยังไดเรกทอรีที่ต้องการ
DELE {filename}	ลบแฟ้มข้อมูล
LIST ([path][fileset])	ต้องการทราบชื่อแฟ้มข้อมูลของไดเรกทอรีที่กำลังใช้งานอยู่
MKD	สร้างไดเรกทอรี
PASS {password}	ส่งรหัสผ่านของผู้ใช้
PORT {socket name}	ส่งเลขที่อยู่ IP และหมายเลขพอร์ตของช่องเชื่อมต่อข้อมูลที่เปิดใช้
PWD	ต้องการทราบชื่อไดเรกทอรีปัจจุบัน
QUIT	ออกจากการใช้งานสถานีบริการ
REST {marker}	กำหนดตำแหน่งไบต์ของแฟ้มข้อมูลที่ต้องการอ่านหรือเขียนกับสถานีบริการ
RETR {filename}	ต้องการรับแฟ้มข้อมูล
RMD {directory}	ลบไดเรกทอรี
RNFR {filename}	ส่งชื่อแฟ้มข้อมูลอันเก่าที่ต้องการเปลี่ยนชื่อ
RNTO {filename}	ส่งชื่อแฟ้มข้อมูลอันใหม่ที่ต้องการเปลี่ยนชื่อ



200 Type set to A.

LIST

150 Opening ASCII mode data connection for /bin/ls.

226 Transfer complete.

QUIT

จากตัวอย่างสถานีลูกข่ายจะได้รับผลตอบ 220 หลังจากสร้างช่องเชื่อมต่อควบคุม หมายเลข 2 ตัวแรกบ่งบอกว่าคำสั่งทำงานสำเร็จ ส่วนหมายเลข 2 ตัวที่สองบ่งบอกว่าผลลัพธ์เกี่ยวข้องกับ การเชื่อมต่อ และหมายเลขลำดับที่ 3 คือ 0 จะสรุปว่าสถานีบริการพร้อมที่จะรับคำสั่งแล้ว

คำสั่งแรกที่สถานีลูกข่ายส่งไปคือ USER โดยมีชื่อผู้ใช้ที่ส่งไปคือ USER ซึ่งได้ผลลัพธ์เท่ากับ 331 หมายเลข 3 ตัวแรกบ่งบอกว่าสถานีบริการต้องการสารสนเทศเพิ่มเติม ส่วน หมายเลข 3 ตัวที่สองบ่งบอกว่าผลลัพธ์เกี่ยวข้องกับ การตรวจสอบการเข้าใช้งาน และหมายเลข 1 ตัวสุดท้ายจะสรุปว่าชื่อผู้ใช้ที่ส่ง ไปถูกต้องแต่ต้องการรหัสผ่านเพิ่มเติม

คำสั่ง PASS ต้องการรหัสผ่านที่ส่งไปคือ PASS และได้ผลลัพธ์กลับมาเป็น 230 ซึ่งบ่งบอกว่าสามารถเข้าใช้งานสถานีบริการได้

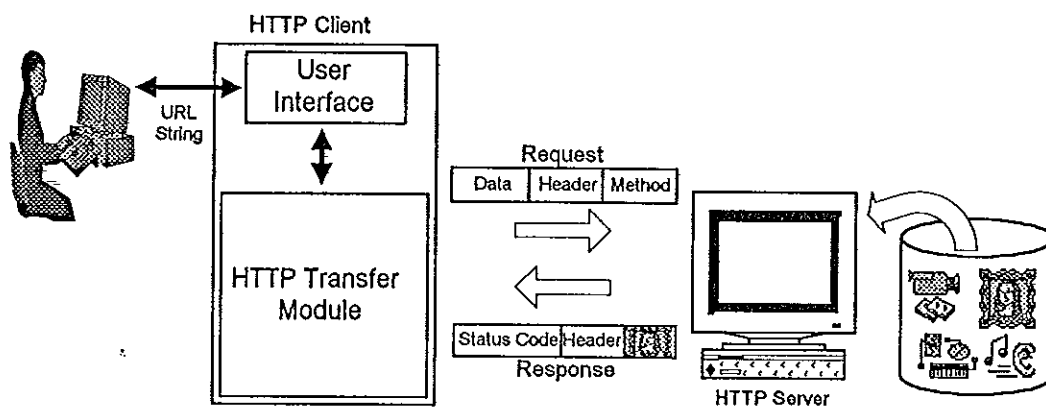
คำสั่งถัดมาคือ PWD ใช้เพื่อต้องการทราบชื่อไคลเอนท์ปัจจุบัน ผลลัพธ์ที่ได้รับตัวแรกคือ 2 บ่งบอกว่าคำสั่งทำงานเสร็จสมบูรณ์ และ หมายเลขที่สองคือ 5 บ่งบอกว่าคำสั่งและผลลัพธ์เกี่ยวข้องกับระบบเพิ่มข้อมูล ชื่อของไคลเอนท์ปัจจุบันจะอยู่ภายในเครื่องหมาย อัฒประภาศ

สิ่งที่จะกระทำถัดไปคือ ต้องการทราบชื่อเพิ่มข้อมูลที่เก็บอยู่ในไคลเอนท์ปัจจุบันซึ่ง ต้องใช้คำสั่ง 3 คำสั่งคือ PORT, TYPE และ LIST คำสั่ง PORT จะบอกให้สถานีบริการทราบถึง เลขที่อยู่ IP และหมายเลขพอร์ตที่สถานีบริการต้องเชื่อมต่อ คำสั่ง TYPE จะบอกให้สถานีบริการ ทราบถึงประเภทของข้อมูลที่จะส่ง คำสั่ง LIST จะบอกให้สถานีบริการใช้ช่องเชื่อมต่อข้อมูลใน การส่งเพิ่มข้อมูลที่เก็บชื่อเพิ่มข้อมูล

เมื่อส่งถ่ายข้อมูลเสร็จสมบูรณ์แล้วและสถานีบริการ ได้ยุติช่องเชื่อมต่อข้อมูล สถานี บริการจะส่งผลตอบ 226 ซึ่งจะบ่งบอกว่าการส่งถ่ายข้อมูลเสร็จสมบูรณ์ แต่อาจจะเป็นไปได้ว่า สถานีลูกข่ายจะได้รับผลตอบนี้ก่อนที่จะได้รับข้อมูลจากช่องเชื่อมต่อข้อมูลได้ทั้งหมดก่อนก็ได้

### 2.1.3.2 โพรโทคอล HTTP

รูปแบบการทำงานของโปรโตคอลจะเป็นลักษณะของการส่งคำร้องขอ (Request) ไปและรอรับข้อมูลตอบ (Response) กลับมา ระหว่างสถานีลูกข่ายกับสถานีบริการ หลังจากที่สถานีลูกข่ายได้รับข้อมูลตอบกลับแล้ว การติดต่อสื่อสารระหว่างกันจะสิ้นสุดลง ซึ่งสามารถอธิบายกลไกการทำงานโดยใช้แผนภาพของแบบจำลอง ดังแสดงในภาพประกอบ 2-7



ภาพประกอบ 2-7 แบบจำลองการทำงานของสถานีลูกข่าย-สถานีบริการ HTTP

จากภาพประกอบ 2-7 โปรแกรมสถานีลูกข่ายจะรับชื่อของทรัพยากรที่ผู้ใช้ต้องการ ซึ่งอยู่ในรูปแบบของ URL คำที่ได้รับนี้จะถูกนำไปใช้ในการสร้างการเชื่อมต่อกับสถานีบริการ เมื่อสามารถทำการเชื่อมต่อระหว่างกันได้แล้ว สถานีลูกข่ายจะส่งคำร้องขอไปให้แก่สถานีบริการ

ในฝั่งของสถานีบริการเมื่อได้รับคำร้องขอแล้ว จะนำค่า URL ที่ระบุไว้ในข้อมูลร้องขอมาใช้จัดเตรียมทรัพยากร และดำเนินการจัดส่งคำตอบกลับไปให้แก่สถานีลูกข่าย เมื่อสถานีบริการจัดส่งคำตอบกลับไปจนหมดแล้วก็จะยุติการเชื่อมต่อกับสถานีให้บริการ

ในส่วน of สถานีลูกข่ายเมื่อได้รับคำตอบแล้วจะนำคำตอบมาตรวจสอบว่ามีทรัพยากรที่ต้องการหรือไม่ ถ้ามีจะจัดเก็บลงเพิ่มข้อมูลที่จัดเตรียมไว้ ถ้าไม่มีจะแจ้งให้ผู้ใช้ทราบ หลังจากนั้นจะยุติการเชื่อมต่อระหว่างกัน

รายละเอียดของคำร้องขอและคำตอบของโปรโตคอลมีดังนี้

- คำร้องขอ

คำร้องขอประกอบด้วยสารสนเทศดังต่อไปนี้

- Request method

ชนิดของ Request method ที่สถานีลูกข่ายใช้งานมีดังนี้

GET – มักใช้ในการร้องขอทรัพยากรที่ต้องการ

HEAD – ใช้ในการร้องขอเฉพาะ HEADER ของทรัพยากร

POST – ใช้เพื่อให้ส่ง Request data ซึ่งข้อมูลที่จะเป็นส่วนหนึ่งของทรัพยากรที่

ระบุไว้ใน URL

PUT – ใช้เพื่อแทนที่ทรัพยากรของสถานีบริการที่ระบุไว้ใน URL ด้วย Request data

- Request header

Request header สามารถเลือกใช้ได้ ส่วนที่ใช้กันหลัก ๆ มีดังนี้

Accept – ระบุชนิดของทรัพยากรที่สถานีใช้บริการยอมรับได้

Authorization – ใช้เมื่อต้องการระบุชื่อผู้ใช้ และ รหัสผ่าน

User-agent – ชื่อและรุ่นของโปรแกรมของสถานีลูกข่าย

Range – ระบุช่วงไบต์ของทรัพยากรที่ต้องการให้สถานีบริการส่งถ่ายมาให้

Host – ระบุสถานีบริการที่เป็นต้นกำเนิดของทรัพยากร (Resource) ใช้ในกรณีที่

เชื่อมต่อผ่าน สถานีบริการพร็อกซี (Proxy Server) ที่ใช้โปรโตคอล HTTP 1.1

- Request data

ถ้าสถานีลูกข่ายใช้ Request method แบบ POST หรือ PUT จะมีการส่ง Request data ตามมา แต่ถ้าเป็นแบบ GET หรือ HEAD จะไม่มีการส่ง Request data

- คำตอบ

คำตอบประกอบด้วยสารสนเทศดังนี้

— Status code

เมื่อสถานีลูกข่ายส่งคำร้องขอมา สิ่งหนึ่งที่สถานีบริการจะต้องส่งกลับมาก็คือ Status code ซึ่งเป็นตัวเลข 3 หลัก แบ่งออกเป็น 4 ประเภทคือ ช่วงระหว่าง 200-299 จะบ่งบอกถึง ผลสำเร็จของการดำเนินการ, 300-399 จะบ่งบอกว่าไม่สามารถจัดหาทรัพยากรได้ เนื่องจากว่ามีการเคลื่อนย้ายทรัพยากรที่ต้องการไปยังที่อื่น, ช่วง 400-499 แสดงถึงความผิดปกติของสถานีลูกข่าย และช่วง 500 ขึ้นไป จะบอกให้ทราบว่าสถานีบริการ ไม่สามารถทำตามคำร้องขอได้

— Response header

ประกอบด้วยข้อมูลที่เกี่ยวข้องกับของสถานีบริการและทรัพยากรที่ส่งมา Header ที่ใช้บ่อย ๆ มีดังนี้

- Server – บอกชื่อและรุ่นสถานีบริการ
- Date – เวลาปัจจุบันตามมาตรฐาน GMT
- Last-modified – วันที่ล่าสุดที่มีการแก้ไขทรัพยากร
- Expires – วันหมดอายุของทรัพยากร
- Content-length – ความยาวของ Response data มีหน่วยเป็นไบต์
- Content-type – ชนิดของ MIME (RFC 1521 และ 1522) ของ Response data
- WWW-authenticate – จะใช้เมื่อ Status code เท่ากับ 401 เพื่อบ่งบอกสถานีลูกข่ายให้กำหนด ชื่อผู้ใช้ และ รหัสผ่าน ในการเข้าใช้ทรัพยากร

— Response data

สถานีบริการจะส่ง Response data ถัดจากส่วนของ Header ซึ่งข้อมูลที่ส่งไปจะมีขนาดของไบต์ข้อมูลตามที่ได้ระบุไว้ใน Content-length ของ Response header

ฝ่ายหอสมุด  
ศูนย์หนังสือ วรรณกรรมวิเศษ

• Universal resource locator:URL

การเข้าใช้ทรัพยากรหลากหลายชนิดบนอินเทอร์เน็ต มีความจำเป็นจะต้องระบุสถานบริการปลายทาง ชื่อทรัพยากรที่ต้องการ และระบุถึงโปรโตคอลที่ใช้ในการเข้าใช้ทรัพยากร การใช้ URL จะทำให้ออกได้ว่าจะใช้ทรัพยากรได้ที่ไหน ต้องการทรัพยากรอะไร และ เข้าใช้ทรัพยากรอย่างไร การใช้ URL จะต้องเป็นไปตามวากยสัมพันธ์ (Syntax) ที่กำหนดไว้ใน Draft RFC จาก www.cern.ch โดยวากยสัมพันธ์ของการใช้งานสามารถแสดงในรูปแบบของ Backus-Naur form ได้ดังนี้

{service}://{host}[:port]/[path/.../][file name]

- service : ชื่อของบริการมาตรฐานที่ระบุถึงโปรโตคอลที่ใช้ในงานในงานวิทยานิพนธ์นี้คือ http กับ ftp
- host : ชื่อสถานบริการปลายทาง หรือ เลขที่อยู่ IP ของสถานบริการ
- port : หมายเลขพอร์ตของสถานบริการ
- path : ชื่อเส้นทางที่ระบุถึงไครเทอริที่เก็บทรัพยากรบนสถานบริการ

ตัวอย่าง แสดงการร้องขอของทรัพยากร bookshop.html ไปยังสถานบริการ  
www.bookshop.canadasia.com และ Response ที่ได้จากการร้องขอ

Request method	Host name
↓	↓
GET <u>http://www.bookshop.canadasia.com.sg/bookshop.html</u> HTTP/1.0	
↑ Universal Resource Locators (URL)	
User-Agent: Mozilla/2.02 (win16;1)	
Pragma: no-cache	
Host: www.bookshop.canadasia.com.sg	
Accept: image/gif, */*	
----- Blank Line (hex = 0D0A0D0A) -----	
Status code	Request header
↓	↓
HTTP/1.1 200 OK.	REQUEST
Date: Sun, 18 May 1997 19:43:17 GMT	RESPONSE
Server: Apache/1.2b7	
Content-Type: text/html	
Last-Modified: Sat, 30 Nov1996 17:23:25 GMT	Response header
Content-Length: 2929	↓
----- Blank Line (hex = 0D0A0D0A) -----	
<html>.<head><title>About The Singapore Bookshop</title></head><BODY bgcolor=#ffeedd link=#ff5000 vlink=#801000><blockquote><ul><li><h1>The Singapore Bookshop</h1><p>here is	Response data

ภาพประกอบ 2-8 ตัวอย่างคำร้องขอและคำตอบ

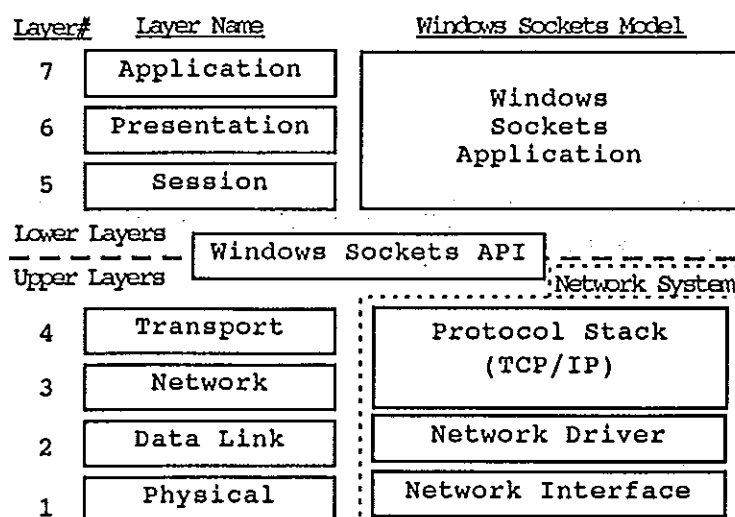
## 2.2 วินโดวส์ซ็อกเก็ต

วินโดวส์ซ็อกเก็ตคือ ส่วนเชื่อมต่อสำหรับการพัฒนาโปรแกรมบนเครือข่ายภายใต้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ ซึ่งส่วนเชื่อมต่อนี้ในมุมมองของนักพัฒนาโปรแกรมก็คือตัวเชื่อมโยงโปรแกรมประยุกต์ของวินโดวส์ซ็อกเก็ต (Windows Socket API) หรือ วินซ็อก API สำหรับงานวิทยานิพนธ์นี้ได้ใช้วินซ็อก API รุ่นที่ 1.1 จากส่วนเชื่อมต่อนี้สามารถอธิบายความสัมพันธ์กับระบบเครือข่ายได้โดยใช้แบบจำลอง OSI ในส่วนของการนำวินซ็อก API ไปใช้พัฒนาโปรแกรมนั้นมีรูปแบบให้เลือกใช้ 3 รูปแบบด้วยกัน ซึ่งจะกล่าวในหัวข้อถัดไป

### 2.2.1 แบบจำลองเครือข่ายวินซ็อก

แบบจำลองเครือข่ายวินซ็อก (Winsock Network Model) ได้เพิ่มเติมขยายจากแบบจำลอง OSI ดังแสดงในภาพประกอบ 2-9 ประกอบด้วย 3 ส่วนหลัก ๆ ดังนี้

- winsock application: จะจัดเตรียมหน้าที่การทำงานของลำดับชั้นที่ 5-7 เรียกว่าลำดับชั้นบน (Upper Layer)
- network system: จะจัดเตรียมหน้าที่การทำงานของลำดับชั้นที่ 1-4 เรียกว่าลำดับชั้นล่าง (Lower Layer )
- winsock API : จะทำหน้าที่เป็นสะพานเชื่อมระหว่าง ลำดับชั้นบนและล่าง



ภาพประกอบ 2-9 แสดงการเปรียบเทียบระหว่างแบบจำลอง OSI กับ แบบจำลองวินซ็อก

( ที่มา: Hall, Martin. et al., 1992 )

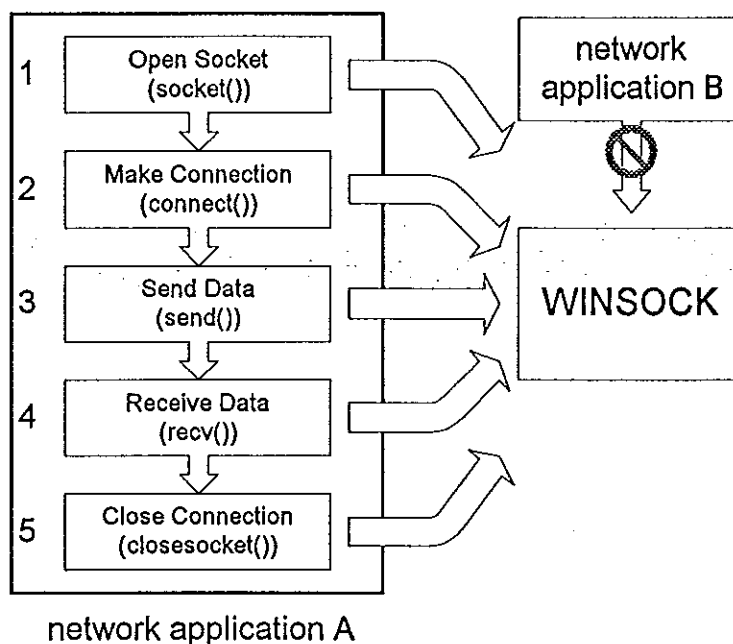
จากภาพประกอบ 2-9 ในส่วนของลำดับชั้นบนประกอบด้วยโปรแกรมประยุกต์ โปรแกรมเดี่ยวที่มีส่วนติดต่อกับผู้ใช้ เมื่อโปรแกรมประยุกต์ในระดับชั้นบนต้องการส่งสารสนเทศ จะกระทำผ่านวินซ็อก API ซึ่งจะส่งสารสนเทศต่อไปยังระดับชั้นล่างต่อไป ในส่วนของระดับชั้นล่างจะมองเห็นสารสนเทศที่ได้รับเป็นเพียงข้อมูลดิบเท่านั้น และจะส่งต่อไปยังจุดหมายปลายทางต่อไป ส่วนการรับสารสนเทศของโปรแกรมประยุกต์ก็จะดำเนินไปในการทำงานเดี่ยวกัน

### 2.2.1 รูปแบบการทำงานของวินซ็อก

การพัฒนาโปรแกรมบนเครือข่ายโดยใช้วินซ็อก API มีรูปแบบการทำงานของวินซ็อก ให้เลือกใช้ 3 รูปแบบด้วยกันคือ แบบกีดกัน (Blocking), แบบอิสระ (Nonblocking) และ แบบอซิงโครนัส (Asynchronous) แต่ละรูปแบบที่เลือกใช้จะมีผลต่อการพัฒนาโปรแกรมและการทำงานของโปรแกรมที่พัฒนาขึ้นแตกต่างกันออกไป

- แบบกีดกัน

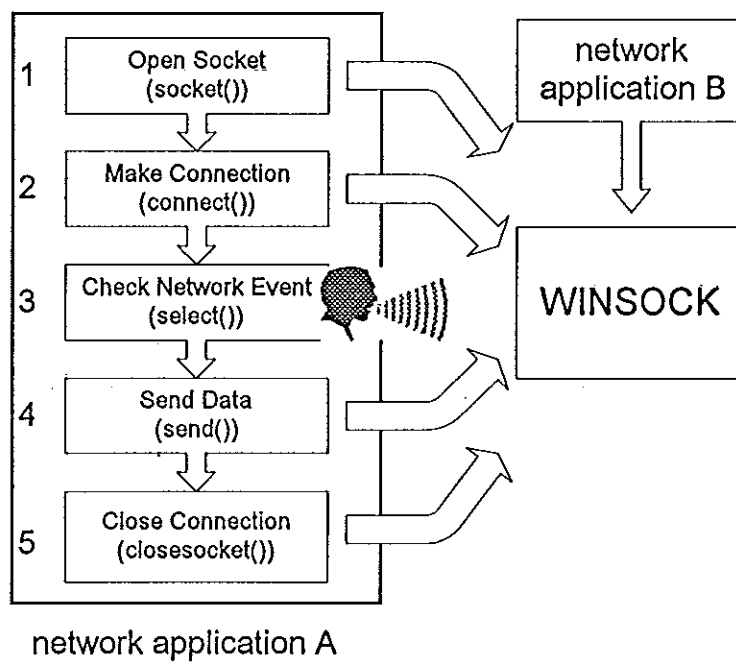
รูปแบบนี้มีความสะดวกและเรียบง่ายในการพัฒนาโปรแกรม โดยทำการเรียกใช้ ฟังก์ชัน (Function) วินซ็อก API ดำเนินไปอย่างเรียงลำดับ แสดงการทำงานดังภาพประกอบ 2-10 ข้อเสียของรูปแบบนี้คือ เมื่อเรียกใช้งานฟังก์ชันวินซ็อกแล้ว จะต้องรอนกว่าฟังก์ชันจะทำงานเสร็จ ซึ่งจะทำให้โปรแกรมประยุกต์อื่น ๆ ไม่สามารถเรียกใช้ฟังก์ชันวินซ็อกได้



ภาพประกอบ 2-10 รูปแบบการทำงานแบบกีดกัน

- แบบอิสระ

รูปแบบนี้ได้แก้ไขข้อเสียของรูปแบบก่อนหน้านี้ โดยทำการตรวจสอบเหตุการณ์ทางเครือข่าย (ต.ย. เชื่อมต่อกับสถานีปลายทางได้แล้ว หรือ มีข้อมูลส่งเข้ามา) ก่อนที่จะเรียกใช้ฟังก์ชันวินซ็อกในลำดับถัดไป แสดงการทำงานดังภาพประกอบ 2-11 แต่รูปแบบนี้ก็ยังคงมีปัญหา คือ การกำหนดระยะเวลาที่เหมาะสมในการตรวจสอบเหตุการณ์ทางเครือข่าย ,

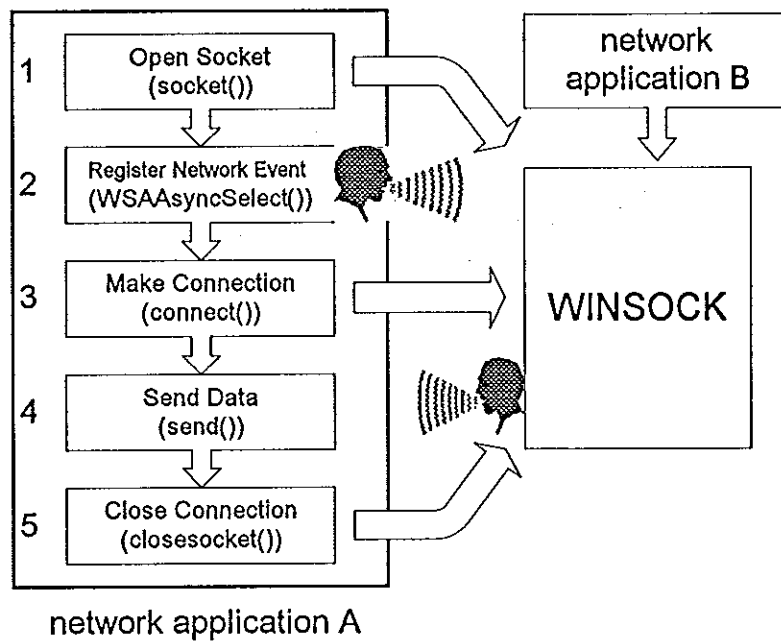


ภาพประกอบ 2-11 รูปแบบการทำงานแบบอิสระ



- แบบอซิงโครนัส

งานวิทยานิพนธ์นี้ได้เลือกใช้รูปแบบนี้ เพราะเป็นรูปแบบที่ดีที่สุด ซึ่งได้แก้ไขข้อเสียของสองรูปแบบที่กล่าวมา แต่ก็ยังคงมีข้อจำกัดคือมีความซับซ้อนในการพัฒนาโปรแกรม การทำงานของรูปแบบนี้จะใช้ฟังก์ชันวินซ็อกอซิงโครนัสอีกทีหนึ่งชั้น (Winsock Asynchronous Extension Functions) ปังบอกให้วินซ็อกส่งข้อความ (Message) มาให้แก่โปรแกรมประยุกต์ผ่านทางระบบจัดส่งข้อความของระบบปฏิบัติการวินโดวส์เมื่อเกิดเหตุการณ์ทางเครือข่ายขึ้น เพื่อที่จะได้เรียกใช้งานฟังก์ชันวินซ็อกในลำดับถัดไป แสดงการทำงานดังภาพประกอบ 2-12



ภาพประกอบ 2-12 รูปแบบการทำงานแบบอซิงโครนัส

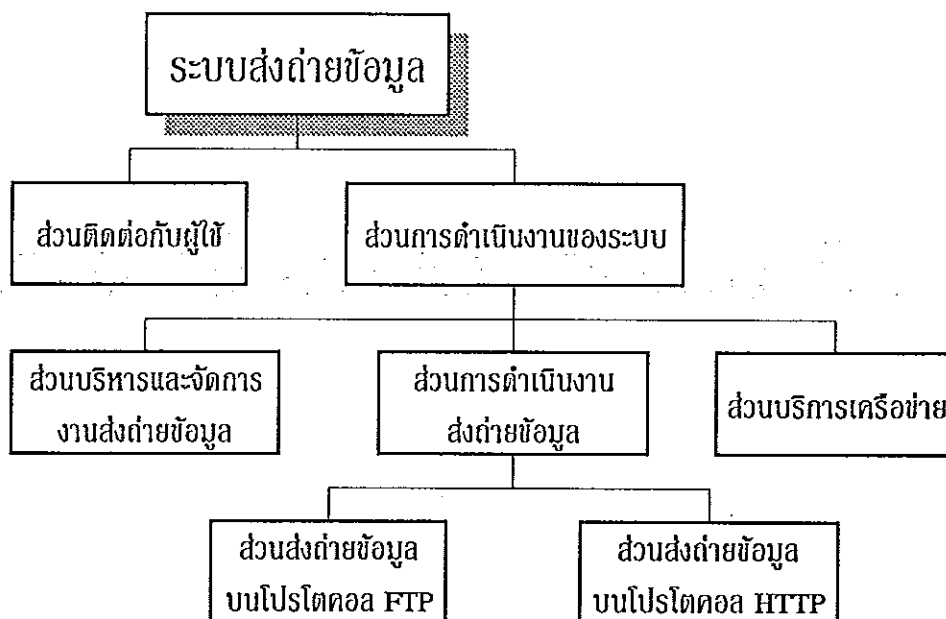
## บทที่ 8

### การออกแบบ พัฒนา และทดสอบระบบ

การออกแบบระบบส่งถ่ายข้อมูลนี้ได้แบ่งงานออกเป็น ส่วน ๆ ซึ่งแต่ละส่วนจะดำเนินงานประสานและสอดคล้องกัน ในการออกแบบแต่ละส่วนจะแสดงให้เห็นถึงหน้าที่สำคัญ ๆ ที่จะนำไปใช้ในการพัฒนาต่อไป ในส่วนของการพัฒนาได้นำสิ่งที่ออกแบบไว้มาดำเนินการพัฒนาด้วยวิธีโปรแกรมเชิงวัตถุและวิธีกลไกสถานะ และได้แสดงให้เห็นถึงกระบวนการดำเนินงานของงานที่พัฒนาในรูปแบบของแผนภาพที่แสดงความสัมพันธ์ของการทำงานร่วมกัน

#### 8.1 โครงสร้างระบบ

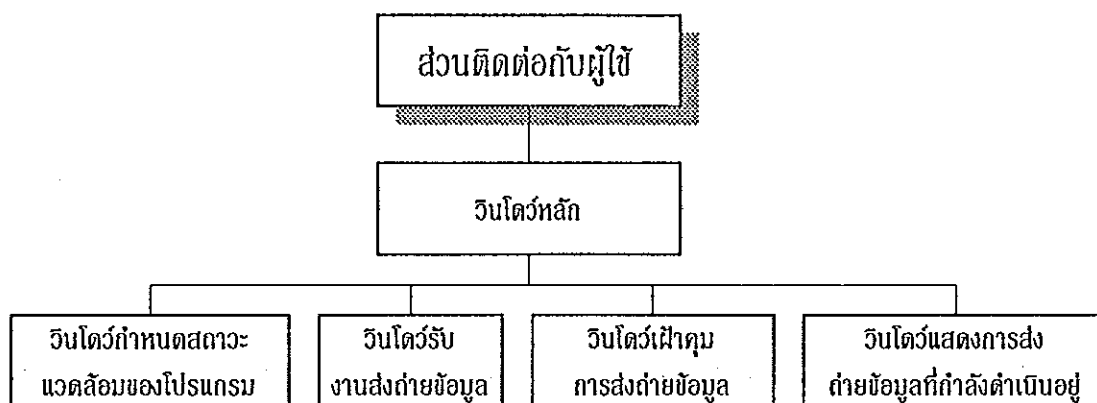
ระบบการส่งถ่ายข้อมูลในงานวิทยานิพนธ์นี้ถูกออกแบบเป็นส่วน ๆ แสดงผังโครงสร้างระบบในภาพประกอบ 3-1 แต่ละส่วนจะมีความสัมพันธ์ในการทำงานดังนี้คือ เริ่มจากผู้ใช้งานป้อนข้อมูลที่จะทำการส่งถ่ายให้กับส่วนติดต่อกับผู้ใช้ จากข้อมูลที่ได้รับนี้จะนำไปใช้ในการดำเนินงานส่งถ่ายข้อมูล ซึ่งเป็นเป็นหน้าที่ของส่วนบริหารและจัดการงานส่งถ่ายข้อมูลที่ใช้ส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP และส่วนส่งถ่ายข้อมูลบนโปรโตคอล HTTP ในการสร้างงานส่งถ่ายขึ้น และจากส่วนส่งถ่ายข้อมูลทั้งสองนี้จะใช้ส่วนบริการเครือข่ายเพื่อเชื่อมต่อกับเครือข่ายในการส่งถ่ายข้อมูลต่อไป



ภาพประกอบ 8-1 ผังโครงสร้างระบบ

### 8.1.1 ส่วนติดต่อกับผู้ใช้

ส่วนติดต่อกับผู้ใช้ของระบบการส่งถ่ายข้อมูลถูกออกแบบให้อยู่ในรูปกรอบหน้าต่างหรือที่เรียกกันว่า “วินโดว์” (Window) โดยวินโดว์ของส่วนติดต่อกับผู้ใช้ยังประกอบด้วยวินโดว์สำหรับดำเนินงานต่าง ๆ ดังแสดงในภาพประกอบ 3-2



ภาพประกอบ 3-2 พังโครงสร้างของส่วนติดต่อกับผู้ใช้

#### • วินโดว์หลัก

วินโดว์นี้ได้ออกแบบให้เป็นวินโดว์หลักที่มีวินโดว์ย่อยอื่น ๆ ที่จะกล่าวในหัวข้อถัดไปเป็นวินโดว์รอง วินโดว์หลักนี้ได้มีหน้าที่การทำงานแบ่งออกเป็นส่วน ๆ ดังนี้

1. ส่วนเมนูระบบ ประกอบด้วยรายการเมนูให้ผู้ใช้เปิดใช้วินโดว์เพื่อกำหนดสถานะแวดล้อมของโปรแกรม, วินโดว์รับงานส่งถ่ายข้อมูล และวินโดว์เฝ้าคุมการส่งถ่ายข้อมูล
2. ส่วนรับข้อมูลสำหรับการส่งถ่าย ข้อมูลที่รับแบ่งออกเป็นสองส่วนคือ ข้อมูลสำหรับส่งถ่ายบนโปรโตคอล FTP และ ข้อมูลสำหรับส่งถ่ายบนโปรโตคอล HTTP
3. ส่วนแสดงผลหลักซ์ของการทำงานของโปรแกรม ข้อมูลที่แสดงเป็นข้อมูลที่ได้จากติดต่อกับสถานีบริการ FTP จากหัวข้อย่อยที่ 5 ข้างล่าง และผลลัพธ์จากการดำเนินงานส่งถ่ายข้อมูลจากส่วนบริหารและจัดการงานส่งถ่ายข้อมูล
4. ส่วนติดต่อกับระบบเพิ่มข้อมูลของระบบ      หน้าที่ส่วนนี้จะทำการสร้างเคลื่อนย้าย และปรับปรุง ระบบเพิ่มข้อมูลของระบบ

5. ส่วนติดต่อกับระบบแก้ไขข้อมูลของสถานีบริการ FTP หน้าในส่วนนี้จะทำการสร้าง เคลื่อนย้าย และปรับปรุง ระบบแก้ไขข้อมูลของสถานีบริการ FTP โดยการติดต่อกับสถานีบริการจะดำเนินการผ่านทาง ส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP

6. ส่วนจัดเตรียมข้อมูลที่จะนำไปสร้างการส่งถ่ายข้อมูล ซึ่งข้อมูลที่จะนำมาจัดเตรียมได้มาจากหัวข้อข้างต้น

- วินโดว์กำหนดสถานะแวดล้อมของโปรแกรม

วินโดว์นี้สำหรับรับข้อมูลที่จะนำไปใช้ร่วมกับการดำเนินงานการส่งถ่ายข้อมูล

- วินโดว์รับงานส่งถ่ายข้อมูล

วินโดว์นี้สำหรับรับข้อมูลที่จะนำไปดำเนินงานการส่งถ่ายข้อมูล

- วินโดว์เฝ้าคุมการส่งถ่ายข้อมูล

วินโดว์นี้เป็นส่วนหนึ่งของส่วนบริหารและจัดการงานส่งถ่ายข้อมูล หน้าของวินโดว์คือแสดงรายละเอียดของข้อมูลของการส่งถ่ายข้อมูลทั้งหมดที่กำลังดำเนินอยู่ และทำการปรับเปลี่ยนรายละเอียดของข้อมูลตามความต้องการของผู้ใช้

- วินโดว์แสดงการส่งถ่ายข้อมูลที่กำลังดำเนินอยู่

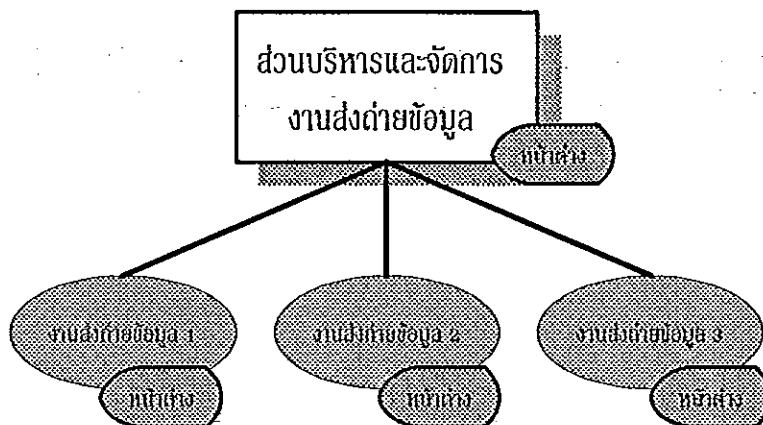
วินโดว์นี้เป็นส่วนหนึ่งของส่วนบริหารและจัดการงานส่งถ่ายข้อมูล หน้าของวินโดว์คือแสดงรายละเอียดของงานส่งถ่ายข้อมูลที่ต้องการงานหนึ่งที่กำลังดำเนินอยู่ โดยวินโดว์นี้จะไปปรากฏเป็นส่วนหนึ่งในวินโดว์หลัก

### 3.1.2 ส่วนการดำเนินงานของระบบ

จากภาพประกอบ 3-1 ส่วนการดำเนินงานของระบบถูกแบ่งออกเป็น 3 ส่วนด้วยกัน โดยทั้งสามส่วนมีรายละเอียดการดำเนินงานออกแบบดังนี้

#### 3.1.2.1 ส่วนบริหารและจัดการงานส่งถ่ายข้อมูล

การดำเนินงานในส่วนนี้ถูกออกแบบให้มีการทำงานได้หลาย ๆ งานพร้อมกัน โดยแต่ละงานจะถูกดำเนินงานในรูปวินโดว์ ดังแสดงในภาพประกอบ 3-3



ภาพประกอบ 8-8 การทำงานของส่วนบริหารและจัดการงานส่งถ่ายข้อมูล

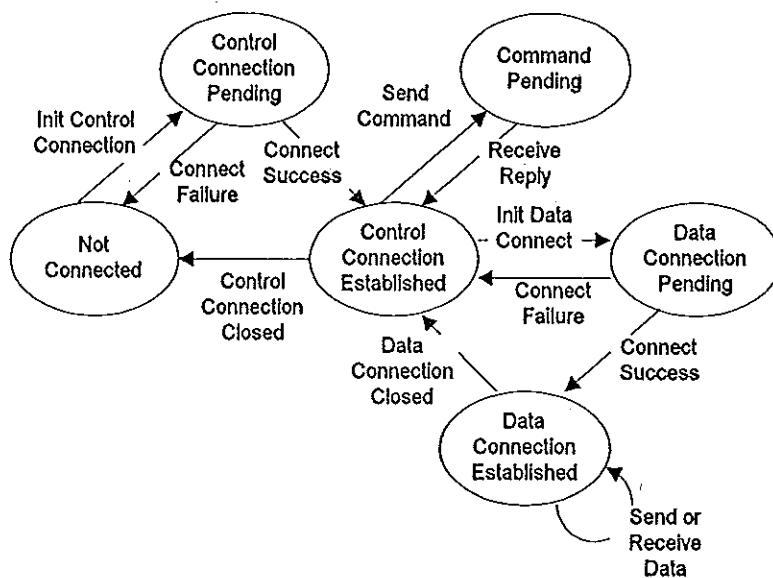
จากภาพประกอบ 3-3 ข้อมูลจากส่วนติดต่อกับผู้ใช้ที่จะนำมาใช้สร้างงานส่งถ่ายข้อมูลจะถูกส่วนบริหารและจัดการนำมาสร้างเป็นงานส่งถ่ายข้อมูลขึ้นมาซึ่งอาจจะมีมากกว่าหนึ่งงานก็ได้ การบริหารและจัดการในส่วนนี้หมายถึง การสร้างงานส่งถ่ายข้อมูลภายใต้กฎเกณฑ์ที่ว่า “งานใดที่ส่งถ่ายข้อมูลได้เร็วจะได้ทำงานก่อน” ส่วนงานที่ส่งถ่ายช้าจะถูกพักการทำงานไว้ชั่วคราวและกลับมาทำงานใหม่อีกครั้ง จากวิธีการบริหารและจัดการดังที่กล่าวมานี้ได้สร้างเป็นวินโดว์แสดงการทำงานไว้ ซึ่งก็คือวินโดว์เฝ้าคุมการส่งถ่ายข้อมูลดังที่กล่าวไว้ในหัวข้อ 3.1.1 และจากงานส่งถ่ายข้อมูลแต่ละงานก็จะมีวินโดว์ที่แสดงสถานะของการส่งถ่ายข้อมูลของตัวเอง ซึ่งก็คือวินโดว์แสดงการส่งถ่ายข้อมูลที่กำลังดำเนินอยู่

### 8.1.2.2 ส่วนการดำเนินงานส่งถ่ายข้อมูล

การดำเนินงานแบ่งออกเป็น 2 ส่วนตามโปรโตคอลที่เลือกใช้

- ส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP

การออกแบบการดำเนินงานในส่วนนี้สามารถใช้วิธีกลไกสถานะ ซึ่งมีสัญลักษณ์และกฎเกณฑ์การใช้แสดงดังภาคผนวก จ วิธีการนี้จะทำให้สิ่งที่ออกแบบมาทำงานสอดคล้องกับรูปแบบการทำงานแบบอซิงโครนัสของวินช็อกตามที่กล่าวไว้ในหัวข้อ 2.2.1 กลไกสถานะของส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP ดังแสดงในภาพประกอบ 3-4



ภาพประกอบ 8-4 กลไกสถานะของการดำเนินงานในส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP

จากภาพประกอบ 3-4 สามารถอธิบายการดำเนินงานของกลไกสถานะได้ดังนี้

— สถานะ

**Not Connected** – สถานะที่ไม่มีการเชื่อมต่อกับสถานีบริการใด ๆ

**Control Connection Pending** – สถานะที่รอการเชื่อมต่อกับสถานีบริการเพื่อสร้างช่องเชื่อมต่อควบคุม

**Control Connection Established** – สถานะที่มีการสร้างช่องเชื่อมต่อควบคุมแล้ว

**Command Pending** – สถานะที่รอผลลัพธ์ของคำสั่ง FTP จากสถานีบริการ

**Data Connection Pending** – สถานะที่รอการเชื่อมต่อกับสถานีบริการเพื่อสร้างช่องเชื่อมต่อข้อมูล

**Data Connection Establish** – สถานะที่มีการสร้างช่องเชื่อมต่อข้อมูลได้แล้ว

– การเปลี่ยนสถานะ

**Init Control Connection** – สร้างช่องเชื่อมต่อควบคุม

**Connect Failure** – สร้างช่องเชื่อมต่อควบคุมไม่สำเร็จ

**Connect Success** – สร้างช่องเชื่อมต่อควบคุมสำเร็จ

**Send Command** – ส่งคำสั่งของ FTP ไปยังสถานีบริการด้วยช่องเชื่อมต่อควบคุม

**Receive Reply** – รับผลลัพธ์ของคำสั่ง FTP จากช่องเชื่อมต่อควบคุม

**Init Data Connect** – สร้างช่องเชื่อมต่อข้อมูล

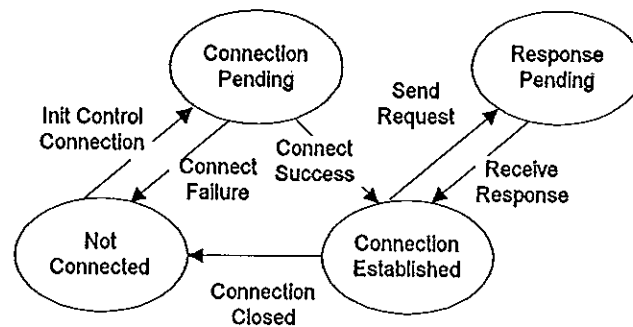
**Send or Receive Data** – ส่งหรือรับข้อมูลจากช่องเชื่อมต่อข้อมูล

**Data Connection Closed** – ยุติการใช้ช่องเชื่อมต่อควบคุม

**Control Connection Closed** – ยุติการใช้ช่องเชื่อมต่อข้อมูล

- ส่วนส่งถ่ายข้อมูลบนโปรโตคอล HTTP

การออกแบบได้ดำเนินการเช่นเดียวกันกับส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP โดยมีกลไกสถานะดังแสดงในภาพประกอบ 3-5



ภาพประกอบ 3-5 กลไกสถานะของการดำเนินงานในส่วนส่งถ่ายข้อมูลบนโปรโตคอล HTTP

จากภาพประกอบ 3-5 สามารถอธิบายการดำเนินงานของกลไกสถานะได้ดังนี้

— สถานะ

**Not Connected** – สถานะที่ไม่มี การเชื่อมต่อกับสถานีบริการ

**Connection Pending** – สถานะที่รอการเชื่อมต่อกับสถานีบริการ

**Connection Established** – สถานะที่มีการสร้างการเชื่อมต่อกับสถานีบริการได้แล้ว

**Response Pending** – สถานะที่รอข้อมูลตอบกลับ

— การเปลี่ยนสถานะ

**Init Connection** – สร้างการเชื่อมต่อกับสถานีบริการ

**Connect Failure** – สร้างการเชื่อมต่อไม่สำเร็จ

**Connect Success** – สร้างการเชื่อมต่อสำเร็จ

**Send Request** – ส่งข้อมูลร้องขอไปยังสถานีบริการ

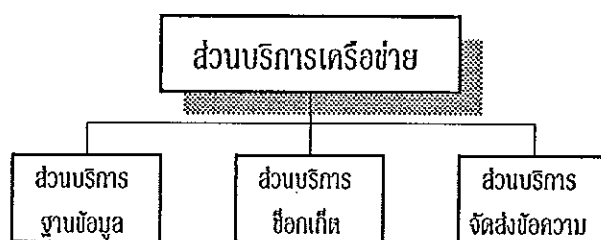
**Receive Response** – รับข้อมูลตอบกลับ

**Connection Closed** – ยุติการการเชื่อมต่อกับสถานีบริการ



### 3.1.2.8 ส่วนบริการเครือข่าย

การดำเนินงานนอกแบบได้แบ่งงานออกเป็น 3 ส่วนด้วยกัน สามารถแสดงในรูปแบบของผังโครงสร้างดังแสดงในภาพประกอบ 3-6



ภาพประกอบ 3-6 โครงสร้างของส่วนบริการเครือข่าย

จากภาพประกอบ 3-6 ได้ออกแบบงานบริการแต่ละส่วนให้มีหน้าที่การทำงานดังนี้

- งานบริการฐานข้อมูล

หน้าที่สำคัญของส่วนงานนี้คือ การให้บริการแก่ส่วนการดำเนินงานส่งถ่ายข้อมูลเพื่อค้นหาเลขที่อยู่ IP ของสถานีบริการ โดยงานบริการฐานข้อมูลนี้จะเรียกใช้วินซ็อก API กลุ่มฐานข้อมูลในการทำงานอีกต่อหนึ่งดังที่ได้แสดงไว้ในภาคผนวก ก และทำหน้าที่รับข้อความจากงานบริการจัดส่งข้อความและส่งต่อข้อความที่ได้รับไปให้แก่ส่วนการดำเนินงานส่งถ่ายข้อมูล

- งานบริการซ็อกเก็ต

งานบริการนี้จะให้บริการการเชื่อมต่อกับสถานีบริการ และให้บริการรับหรือส่งข้อมูลกับสถานีบริการที่เชื่อมต่อกันได้แล้ว โดยงานบริการซ็อกเก็ตนี้จะเรียกใช้วินซ็อก API กลุ่มซ็อกเก็ตในการทำงานอีกต่อหนึ่ง นอกจากนี้ยังทำหน้าที่ถ่ายทอดข้อความที่ได้รับจากงานบริการจัดส่งข้อความไปยังส่วนการดำเนินงานส่งถ่ายข้อมูล

- งานบริการจัดส่งข้อความ

จากหัวข้อที่ 2.2.1 เรื่องรูปแบบการทำงานของวินซ็อกอซิง โครนัส จากที่กล่าวไว้ว่าเมื่อเกิดเหตุการณ์ทางเครือข่ายขึ้นวินซ็อกจะส่งเหตุการณ์ทางเครือข่ายมาให้แก่โปรแกรมประยุกต์ผ่านทางระบบจัดส่งข้อความของระบบปฏิบัติการ ซึ่งหน้าที่ของงานบริการนี้คือการรับข้อความของเหตุการณ์ทางเครือข่ายจากระบบปฏิบัติการและส่งต่อไปให้แก่งานบริการฐานข้อมูลและงานบริการซ็อกเก็ตดังที่ได้กล่าวผ่านมา

## 3.2 การพัฒนาระบบ

การพัฒนาระบบได้ใช้วิธีการ โปรแกรมแบบเชิงวัตถุ โดยใช้โปรแกรมแปลภาษา Borland C++ พัฒนาโครงสร้างระบบที่ออกแบบไว้ขึ้นเป็นคลาส (Class) ต่าง ๆ ขึ้นมา และได้นำคลาสบางส่วนและเครื่องมือการพัฒนาของโปรแกรมแปลภาษามาใช้งาน ซึ่งคลาสที่นำมาใช้ส่วนใหญ่จะอยู่ในส่วนติดต่อกับผู้ใช้ ส่วนการพัฒนาในส่วนการดำเนินงานของระบบจะไม่มี การนำคลาสดังกล่าวมาใช้แต่จะมีการเรียกใช้วินโดว์ API

### 3.2.1 ส่วนติดต่อกับผู้ใช้

การพัฒนาส่วนติดต่อกับผู้ใช้นี้ได้พัฒนาโดยใช้คลาสที่ทำหน้าเป็นส่วนเชื่อมต่อกับผู้ใช้ จากโปรแกรมแปลภาษา ซึ่งคลาสเหล่านี้สามารถศึกษาการใช้งานได้จากคู่มือการพัฒนาโปรแกรม ภาพของวินโดว์ที่พัฒนาขึ้นได้แสดงไว้ในภาคผนวก ง ส่วนการดำเนินงานพัฒนามีดังนี้

#### 3.2.1.1 การพัฒนาส่วนติดต่อกับผู้ใช้

การพัฒนาส่วนติดต่อกับผู้ใช้แบ่งออกเป็น 5 ส่วนดังนี้

- วินโดว์หลัก

การพัฒนาวินโดว์นี้ได้พัฒนาให้เป็นคลาสหลักของโปรแกรม และได้กำหนดคลาสนี้ให้มีคุณสมบัติเป็นวินโดว์ตามวิธีการพัฒนาจากคู่มือของโปรแกรมแปลภาษา ภายในพื้นที่ของวินโดว์ได้ใช้คลาสของโปรแกรมแปลภาษาที่ทำหน้าที่เป็นส่วนติดต่อนำมาจัดวางไว้ ดังแสดงในภาพประกอบ ง-1 และแต่ละองค์ประกอบของส่วนติดต่อกับผู้ใช้ของวินโดว์แสดงไว้ในตาราง ง-1 นอกจากนี้เรียกใช้งานคลาสที่เป็นส่วนติดต่อกับผู้ใช้ที่กล่าวมาแล้วยังได้พัฒนาคลาสอื่น ๆ ขึ้นมาเพื่อทำงานร่วมกับคลาสส่วนติดต่อนี้ ซึ่งคลาสที่พัฒนาขึ้นเองนี้จะกล่าวถึงในหัวข้อ 3.2.1.3 ในส่วนของหน้าที่การทำงานต่าง ๆ ที่กำหนดไว้ในช่วงของการออกแบบได้พัฒนาขึ้นเป็นโมดูลของคลาสหลักขึ้นมาทำงานตามที่ได้กำหนดไว้

- วินโดว์กำหนดสถานะแวดล้อมของโปรแกรม

การพัฒนาวินโดว์นี้ได้พัฒนาโดยใช้เครื่องมือของโปรแกรมแปลภาษาที่ชื่อว่า Resourch Workshop ซึ่งได้ภาพของหน้าต่างดังแสดงในภาพประกอบ ๓-2 และแต่ละองค์ประกอบของส่วนติดต่อกับผู้ใช้ของวินโดว์ดังแสดงในตาราง ๓-2 จากวินโดว์นี้ได้สร้างคลาสขึ้นมาเพื่อทำงานร่วมกัน ซึ่งคลาสที่พัฒนามันเองนี้จะกล่าวถึงหัวข้อ 3.2.1.3

- วินโดว์รับงานส่งถ่ายข้อมูล

การพัฒนาวินโดว์มีการดำเนินการเช่นเดียวกับการพัฒนาวินโดว์จากหัวข้อก่อนหน้านี้ โดยมีภาพของวินโดว์ดังแสดงในภาพประกอบ ๓-3 และแต่ละองค์ประกอบของส่วนติดต่อกับผู้ใช้ของวินโดว์ดังแสดงในตาราง ๓-3

- วินโดว์เฝ้าคุมการส่งถ่ายข้อมูล

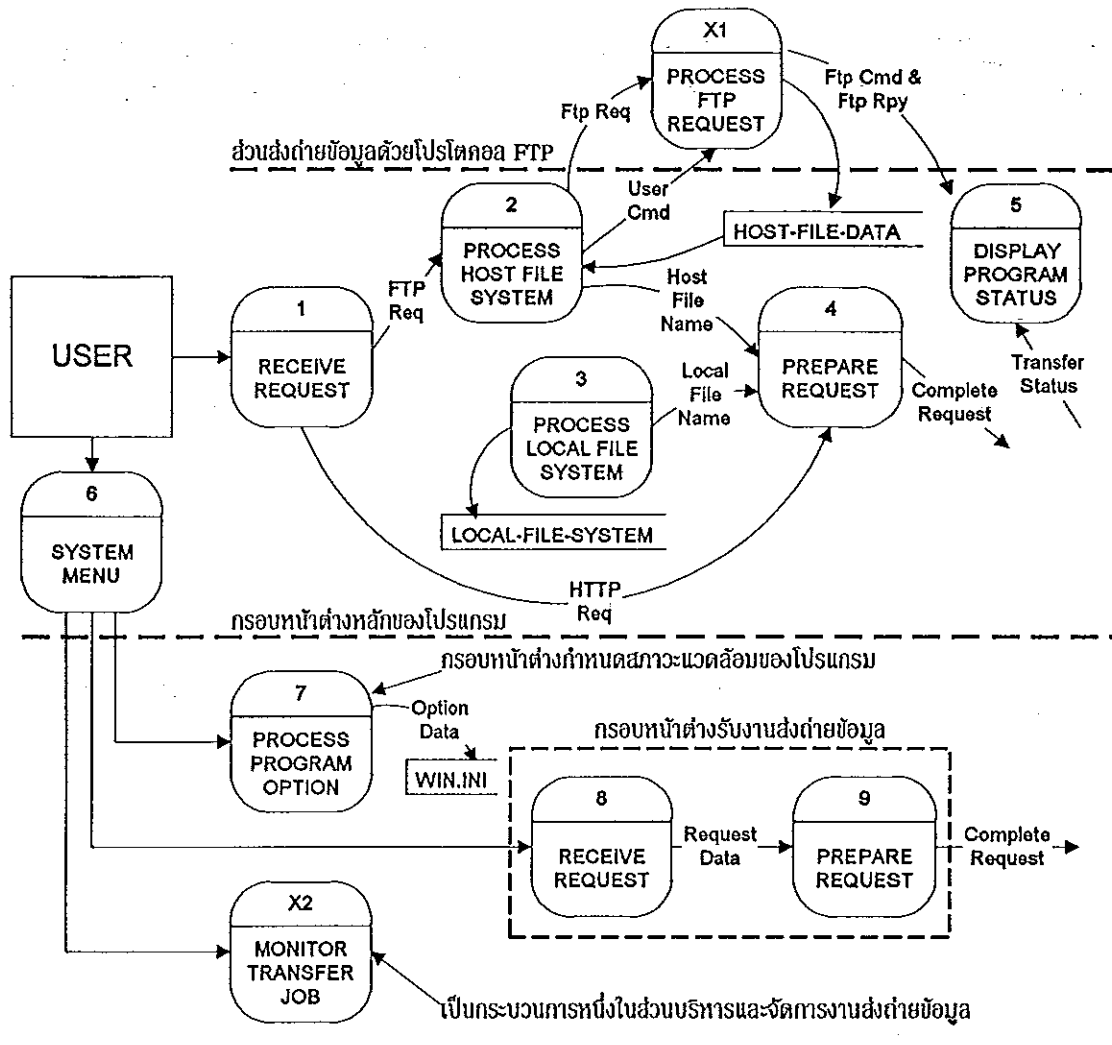
การพัฒนาวินโดว์ได้สร้างขึ้นเป็นคลาสที่มีการกำหนดคุณสมบัติให้เป็นวินโดว์เพื่อที่จะสามารถแสดงข้อความต่าง ๆ ได้ ซึ่งมีภาพของวินโดว์ดังแสดงในภาพประกอบ ๓-4 นอกจากนี้คลาสที่พัฒนามันได้เรียกใช้คลาสเมนูของโปรแกรมแปลภาษาเพื่อสร้างเมนู ซึ่งมีรายละเอียดของเมนูดังแสดงในตาราง ๓-4 และ ๓-5 รายละเอียดการทำงานและการพัฒนาคลาสที่กล่าวไว้ในตอนต้นจะอธิบายการพัฒนาไว้ในหัวข้อส่วนบริการและจัดการงานส่งถ่ายข้อมูล

- วินโดว์แสดงการส่งถ่ายข้อมูลที่กำลังดำเนินอยู่

การพัฒนาวินโดว์ได้สร้างขึ้นเป็นคลาสซึ่งจะกล่าวถึงการทำงานและการพัฒนาไว้ในหัวข้อส่วนบริการและจัดการงานส่งถ่ายข้อมูล คลาสที่พัฒนามันนี้ได้กำหนดคุณสมบัติให้เป็นวินโดว์เพื่อที่จะสามารถแสดงข้อความ หรือวาดภาพได้ ซึ่งมีภาพของวินโดว์ดังแสดงในภาพประกอบ ๓-3 หมายเลข 23

### 3.2.1.2 กระบวนการทำงานของส่วนติดต่อกับผู้ใช้

รายละเอียดการดำเนินงานของส่วนติดต่อกับผู้ใช้สามารถแทนด้วยภาพกระแสข้อมูล ( DFD: Data Flow Diagram ) ดังแสดงในภาพประกอบ 3-7 ซึ่งมีกฎเกณฑ์การใช้ดังแสดงในภาคผนวก ๓



ภาพประกอบ 8-7 แผนภาพกระแสข้อมูลของส่วนติดต่อกับผู้ใช้

จากภาพประกอบ 3-7 สามารถอธิบายการดำเนินงานของแผนภาพได้ดังนี้

• คำอธิบายของกระบวนการ

กระบวนการที่ 1

ชื่อกระบวนการ RECEIVE REQUEST

คำอธิบาย รับข้อมูลจากผู้ใช้ที่จะนำไปใช้สร้างการส่งถ่ายข้อมูล ถ้าข้อมูลที่รับได้เป็นข้อมูลสำหรับการส่งถ่ายบนโปรโตคอล FTP จะส่งข้อมูลที่ได้ไปให้แก่กระบวนการที่ 2 แต่ถ้าหากเป็นโปรโตคอล HTTP จะส่งไปให้แก่กระบวนการที่ 4

- กระบวนการที่ 2  
ชื่อกระบวนการ PROCESS HOST FILE SYSTEM  
คำอธิบาย สร้างการเชื่อมต่อไปยังสถานีบริการ FTP และเข้าใช้ระบบเพิ่มข้อมูลของสถานีบริการผ่านทางส่วนติดต่อกับผู้ใช้ การดำเนินการที่กล่าวมาจะไม่มี การส่งถ่ายเพิ่มข้อมูลใด ๆ และจะกระทำผ่านส่วนส่งถ่ายข้อมูลบน โพรโทคอล FTP
- กระบวนการที่ 3  
ชื่อกระบวนการ PROCESS LOCAL FILE SYSTEM  
คำอธิบาย เข้าใช้ระบบเพิ่มข้อมูลของระบบที่ใช้อยู่ โดยดำเนินการผ่านส่วนติดต่อกับผู้ใช้
- กระบวนการที่ 4  
ชื่อกระบวนการ PREPARE REQUEST  
คำอธิบาย จัดเตรียมข้อมูลสำหรับนำไปใช้ในการส่งถ่ายข้อมูล โดยถ้าเป็นการส่งถ่ายด้วย โพรโทคอล FTP จะนำชื่อเพิ่มจากกระบวนการที่ 2 และ 3 มาใช้ในการจัดเตรียมด้วย ส่วนการส่งถ่ายด้วย โพรโทคอล HTTP สามารถจัดเตรียมข้อมูลที่ได้จากกระบวนการที่ 1 ได้เลย
- กระบวนการที่ 5  
ชื่อกระบวนการ DISPLAY PROGRAM STATUS  
คำอธิบาย แสดงข้อความการติดต่อกับสถานีบริการ FTP และแสดงผลพร้อมของการส่งถ่ายข้อมูลจากส่วนบริหารและจัดการงานส่งถ่ายข้อมูล
- กระบวนการที่ 6  
ชื่อกระบวนการ SYSTEM MENU  
คำอธิบาย กระบวนการนี้เป็นการดำเนินการของโปรแกรมแปลภาษา ซึ่งจะถ่ายทอดคำสั่งจากผู้ใช้เพื่อเปิดใช้งานวินโดวที่ต้องการต่อไป
- กระบวนการที่ 7  
ชื่อกระบวนการ PROCESS PROGRAM OPTION  
คำอธิบาย รับข้อมูลจากผู้ใช้ที่จะนำไปใช้ร่วมกับการดำเนินการส่งถ่ายข้อมูล และจัดเก็บข้อมูลที่ได้รับไว้ในเพิ่มข้อมูลระบบ win.ini

กระบวนการที่ 8

ชื่อกระบวนการ RECEIVE REQUEST

คำอธิบาย รับข้อมูลจากผู้ใช้ที่จะนำไปใช้ดำเนินการส่งถ่ายข้อมูล

กระบวนการที่ 9

ชื่อกระบวนการ PREPARE REQUEST

คำอธิบาย จัดเตรียมข้อมูลสำหรับนำไปใช้ในการส่งถ่ายข้อมูล ทั้งที่เป็นการส่งถ่ายด้วยโปรโตคอล FTP และการส่งถ่ายด้วยโปรโตคอล HTTP

กระบวนการที่ X1

ชื่อกระบวนการ PROCESS FTP REQUEST

คำอธิบาย การพัฒนาส่วนส่งถ่ายข้อมูลด้วยโปรโตคอล FTP

กระบวนการที่ X2

ชื่อกระบวนการ MONITOR TRANSFER JOB

คำอธิบาย การพัฒนาส่วนบริหารและจัดการงานส่งถ่ายข้อมูล

- คำอธิบายตัวจัดเก็บข้อมูล

HOST-FILE-DATA = \*แก้ไขข้อความที่แสดงชื่อแฟ้มและชื่อไคลเรททอรีของสถานีบริการ FTP\*

WIN.INI = \*คู่มือของระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์\*

- คำอธิบายของกระแสข้อมูล

FTP Req =

URL + USER-NAME + PASSWORD

HTTP Req =

URL

Ftp Req =

TRANS-METHOD + URL + USER-NAME + PASSWORD + LOCAL-FILENAME +  
BYTEDONE

User Cmd = \*คำสั่งในการเข้าใช้ระบบแฟ้มของสถานีบริการ FTP ที่ได้จากวินโดวส์หลัก\*

Host File Name = \*ชื่อแฟ้มของสถานีบริการ FTP ที่ต้องการส่งถ่ายมา\*

Local File Name = \*ชื่อเพิ่มของระบบเพิ่มที่ใช้งานอยู่ที่ต้องการส่งถ่ายไปยังสถานีบริการ\*

HTTP Req =

URL

Complete Request =

JOBID + PROTOCOL + TRANS-METHOD + URL + USER-NAME + PASSWORD +  
 LOCAL-FILENAME + JOBSTATE + WAITING-TIME + BYTEDONE + FILE-SIZE +  
 STARTUP-TIME

Option Data =

WWW-Proxy-Name + Editor-Name + Download-Dir-Name + E-mail + Scan-Time +  
 Max-Connection + Recv-Bytes + Send-Bytes

Request Data =

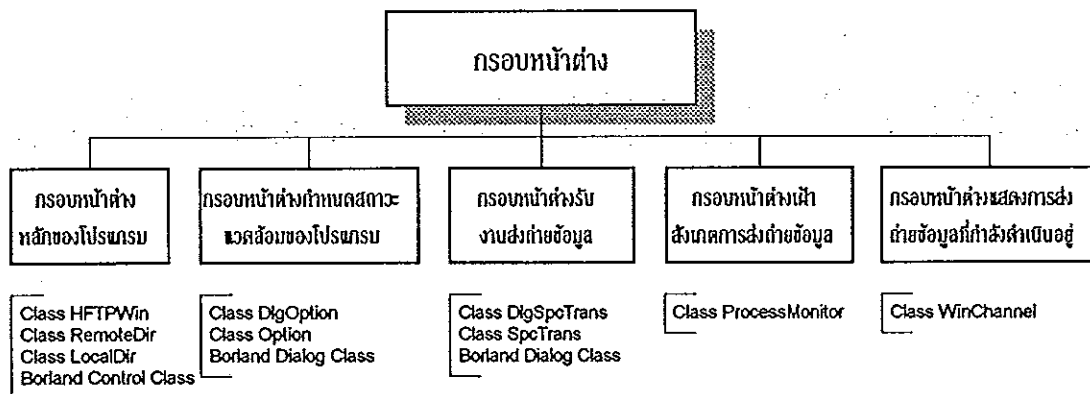
PROTOCOL + TRANS-METHOD + URL + USER-NAME + PASSWORD +  
 LOCAL-FILENAME + STARTUP-TIME

- คำอธิบายหน่วยของข้อมูล

ดูคำอธิบายหน่วยของข้อมูลจากหัวข้อ 3.2.2.2

### 3.2.1.3 ส่วนของโปรแกรม

การพัฒนาในส่วนการติดต่อกับผู้ใช้ นี้ ได้ดำเนินการสร้างเป็นคลาสขึ้นมาซึ่งสามารถแสดงในรูปของผังโครงสร้างดังแสดงในภาพประกอบ 3-8



ภาพประกอบ 3-8 ผังโครงสร้างคลาสของส่วนติดต่อกับผู้ใช้

จากภาพประกอบ 3-8 การพัฒนาและหน้าที่ของแต่ละคลาสของแต่ละวินโดว์ แสดงดัง  
ในตาราง 3-1

ตาราง 3-1 คลาสของส่วนติดต่อกับผู้ใช้

คลาส	คำอธิบาย
HFTPWin	เป็นคลาสหลักของโปรแกรมที่ทำหน้าที่ประสานการทำงานกับคลาสอื่น ๆ ในการทำงานเป็นระบบส่งถ่ายข้อมูล
RemoteDir	เป็นคลาสที่ติดต่อกับส่วนติดต่อกับผู้ใช้เพื่อแสดงการเข้าใช้และเข้าใช้ระบบแฟ้มของสถานีบริการ FTP ที่กำลังใช้งานอยู่
LocalDir	เป็นคลาสที่ติดต่อกับส่วนติดต่อกับผู้ใช้เพื่อแสดงการเข้าใช้และเข้าใช้ระบบแฟ้มของเครื่องคอมพิวเตอร์ที่กำลังใช้งานอยู่
Borland Control Class	เป็นคลาสส่วนติดต่อกับผู้ใช้ของโปรแกรมแปลภาษา
DlgOption	เป็นคลาสที่ติดต่อกับคลาสวินโดว์ชนิด Dialog ของโปรแกรมแปลภาษา
Option	เป็นคลาสที่จัดเก็บรายละเอียดของข้อมูลที่จะกำหนดสถานะแวดล้อมของโปรแกรม
Borland Dialog Class	เป็นคลาสส่วนติดต่อกับผู้ใช้ของโปรแกรมแปลภาษาที่เป็นวินโดว์ชนิด Dialog
DlgSpcTrans	เป็นคลาสที่ติดต่อกับคลาสวินโดว์ชนิด Dialog ของโปรแกรมแปลภาษา
SpcTrans	เป็นคลาสที่จัดเก็บรายละเอียดของข้อมูลที่จะนำไปใช้สร้างงานส่งถ่ายข้อมูล
ProcessMonitor	เป็นคลาสที่สร้างขึ้นโดยกำหนดให้มีคุณสมบัติเป็นวินโดว์ เพื่อให้สามารถแสดงข้อมูลต่าง ๆ ได้ ดูรายละเอียดเพิ่มเติมในหัวข้อ 3.2.2.3
WinChannel	เป็นคลาสที่สร้างขึ้นโดยกำหนดให้มีคุณสมบัติเป็นวินโดว์ เพื่อให้สามารถแสดงข้อมูลต่าง ๆ ได้ ดูรายละเอียดเพิ่มเติมในหัวข้อ 3.2.2.3



### 3.2.2 ส่วนการดำเนินงานของระบบ

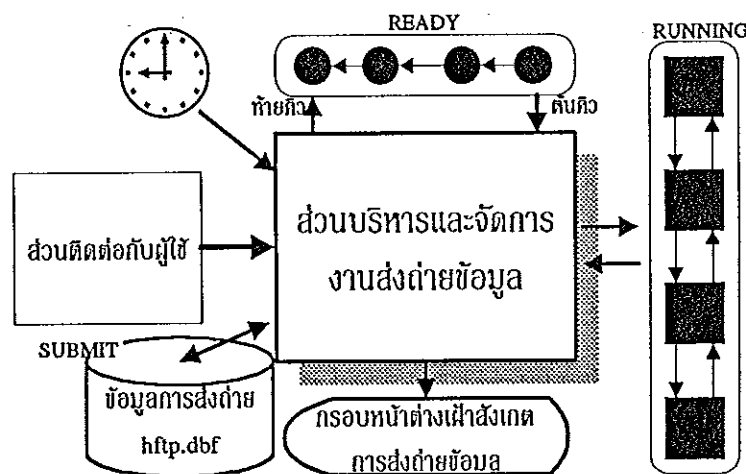
การพัฒนาในส่วนนี้จะนำสิ่งที่ได้ออบแบบไว้มาดำเนินการพัฒนาโดยมีกระบวนการทำงานของระบบในส่วนนี้แสดงในรูปของแผนภาพกระแสข้อมูลดังแสดงภาพประกอบ 3-11 ส่วนการดำเนินการพัฒนาได้แบ่งออกเป็นหัวข้อดังนี้

#### 3.2.2.1 การพัฒนาส่วนการดำเนินงานของระบบ

การพัฒนาแบ่งออกเป็น 3 ส่วนดังนี้

- ส่วนบริหารและจัดการงานส่งถ่ายข้อมูล

การดำเนินงานในส่วนนี้ได้พัฒนาเป็นกลุ่มของคลาสขึ้นมาทำงานร่วมกัน ซึ่งสามารถแสดงการทำงานดังแสดงในภาพประกอบ 3-9



ภาพประกอบ 3-9 การทำงานของส่วนบริหารและจัดการงานส่งถ่ายข้อมูล

จากภาพประกอบ 3-9 คลาสที่ทำหน้าที่ในส่วนของการบริหารและจัดการจะนำข้อมูล ที่นำมาสร้างการส่งถ่ายจากส่วนติดต่อกับผู้ใช้มาจัดเก็บไว้ในแฟ้ม hftp.dbf แบบเรียงลำดับและนำไปสร้างเป็นงานส่งถ่ายขึ้น งานส่งถ่ายข้อมูลที่สร้างขึ้นนี้จะมีสถานะการทำงานอยู่ 3 สถานะด้วยกันดังนี้

**SUBMIT** – งานส่งถ่ายข้อมูลที่อยู่ในสถานะนี้เป็นเพียงข้อมูลที่เก็บอยู่ในแฟ้ม hftp.dbf แบบเรียงลำดับเท่านั้น ไม่มีการดำเนินงานส่งถ่ายข้อมูลใด ๆ

**READY** – งานส่งถ่ายข้อมูลที่อยู่ในสถานะนี้มีความพร้อมที่จะส่งถ่ายข้อมูล งานในสถานะนี้เป็นเพียงเลขที่ประจำตัวของงาน โดยมีโครงสร้างการจัดเก็บงานแบบ Linked List ที่มีการเข้าใช้ข้อมูลแบบคิว จากภาพประกอบ 3-9 งานในสถานะนี้คือ ภาพวงกลมสีดำ

**RUNNING** – งานส่งถ่ายข้อมูลที่อยู่ในสถานะนี้ อยู่ในสถานะที่กำลังดำเนินการส่งถ่ายข้อมูลอยู่ งานในสถานะนี้ได้พัฒนาขึ้นเป็นคลาสที่จะดำเนินการส่งถ่ายข้อมูลขึ้นจริง ๆ โดยมีโครงสร้างการจัดเก็บคลาสนี้หรืองานส่งถ่ายข้อมูลแบบ Double Linked List จากภาพประกอบ 3-9 งานในสถานะนี้คือ ภาพสี่เหลี่ยมจัตุรัสสีดำ

จากงานส่งถ่ายข้อมูลที่สร้างขึ้นครั้งแรกนี้จะอยู่ในสถานะ SUBMIT ซึ่งก็คือการจัดเก็บไว้ในแฟ้ม hftp.dbf และจะสร้างเป็นงานที่พร้อมที่จะส่งถ่ายข้อมูลหรืองานที่อยู่ในสถานะ READY การดำเนินการขั้นถัดไปจะเป็นการบริหารและจัดการงานที่สร้างขึ้นให้ได้ทำการส่งถ่ายข้อมูล ซึ่งได้พัฒนาเป็นโมดูลของคลาสบริหารและจัดการขึ้นมา โดยมีการทำงานดังนี้คือ เมื่อได้รับสัญญาณพิกจากระบบตามที่ได้กำหนดเป็นระยะเวลาไว้ จะมีการตรวจสอบไปยังงานทุกงานที่อยู่ในสถานะ RUNNING ว่ามีความล่าช้าในการส่งถ่ายข้อมูลหรือไม่ ถ้าเกิดความล่าช้าจะยุติการทำงานของงานนั้นและย้ายงานไปสู่สถานะ READY จากนั้นนำงานที่อยู่ในสถานะ READY ที่ถูกย้ายมาก่อนหน้านี้มาทำการส่งถ่ายข้อมูลต่อ และจบการทำงานในการบริหารและจัดการ และเมื่อได้รับสัญญาณพิกาลูกใหม่การดำเนินการดังที่กล่าวมาจะเกิดขึ้นอีกครั้งหนึ่ง การแสดงกิจกรรมของการจัดการและบริหารที่กล่าวมาข้างต้นนี้ได้พัฒนาเป็นคลาสขึ้นอีกคลาสหนึ่งที่มีวินโดว์แสดงกิจกรรมดังกล่าวไว้ นอกจากคลาสนี้จะแสดงข้อมูลตามที่กล่าวมาแล้วยังมีรายการเมนูเพื่อติดต่อกับผู้ใช้ให้สามารถปรับเปลี่ยนข้อมูลของการส่งถ่ายที่เก็บไว้ในแฟ้ม hftp.dbf ได้อีกด้วย

จากงานส่งถ่ายข้อมูลงานหนึ่งที่อยู่ในสถานะ RUNNING เมื่อได้ดำเนินการส่งถ่ายข้อมูลเสร็จสิ้นแล้วคลาสนี้ดำเนินการส่งถ่ายข้อมูลจะตรวจสอบว่าได้ดำเนินการส่งถ่ายข้อมูลครบสมบูรณ์หรือไม่ ซึ่งแบ่งแยกการตรวจสอบออกเป็น 2 โปรโตคอล แต่มีวิธีการตรวจสอบเหมือนกันคือ ตรวจสอบจากจำนวนไบต์ข้อมูลที่ได้ดำเนินการส่งถ่ายไปแล้วกับจำนวนไบต์ข้อมูลทั้งหมดที่ต้องดำเนินการส่งถ่าย ซึ่งการหาจำนวนไบต์ข้อมูลหลังสุดนี้ได้กล่าวเอาไว้ในหัวข้อ 2.1.3 เมื่อตรวจสอบแล้วพบว่าส่งถ่ายไม่ครบจะสร้างงานส่งถ่ายใหม่ที่อยู่ในสถานะ READY อีกครั้งหนึ่ง

ในระหว่างที่มีการดำเนินงานส่งถ่ายข้อมูล และได้เกิดไฟฟ้าดับขึ้น ผู้ใช้สามารถกลับมาดำเนินการส่งถ่ายข้อมูลใหม่ต่อจากเดิมได้โดยกระทำผ่านรายการเมนูของวินโดว์เฝ้าคุมการส่งถ่ายข้อมูล ซึ่งคลาสที่ดูแลวินโดว์นี้จะนำข้อมูลของการส่งถ่ายจากแฟ้ม hftp.dbf ไปสร้างเป็นงานที่พร้อมที่จะส่งถ่ายข้อมูลหรืองานที่อยู่ในสถานะ READY ต่อไป

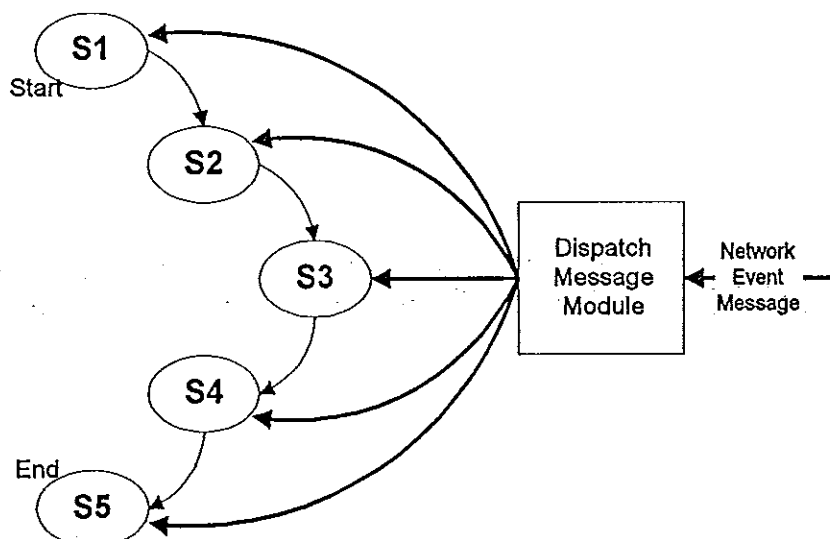
จากการดำเนินการพัฒนาดังที่กล่าวมาทั้งหมดที่กล่าวมานี้สามารถแสดงกระบวนการทำงานในรูปของแผนภาพกระแสข้อมูลดังแสดงในภาพประกอบ 3-12

- ส่วนการดำเนินงานส่งถ่ายข้อมูล

การดำเนินงานได้นำสิ่งที่ออกแบบไว้ด้วยวิธีกลไกสถานะจากหัวข้อที่ 3.1.2.2 มาดำเนินการพัฒนา ซึ่งแบ่งออกเป็น 2 หัวข้อดังนี้

- ส่วนส่งถ่ายข้อมูลบนโปรโตคอล FTP

ในส่วนนี้ได้แบ่งการพัฒนาออกเป็น 2 ส่วนคือ ส่วนเชื่อมต่อควบคุม และส่วนเชื่อมต่อข้อมูลหรืออาจเปรียบเทียบการทำงานของทั้งสองส่วนนี้ได้เท่ากับ โมดูล IP และ โมดูล DTP จากหัวข้อที่ 2.1.3.1 ซึ่งทั้ง 2 ส่วนนี้ได้แยกพัฒนาออกเป็น 2 คลาสด้วยกัน แต่ได้พัฒนาด้วยวิธีกลไกสถานะแบบเดียวกัน ซึ่งสามารถแสดงรูปแบบทั่วไปของการพัฒนาดังแสดงในภาพประกอบ 3-10



ภาพประกอบ 3-10 รูปแบบทั่วไปของการพัฒนาส่วนการดำเนินงานส่งถ่ายข้อมูล

จากภาพประกอบ 3-10 รูปวงรีที่มีตัวอักษร S กำกับอยู่จะแทนสถานะ ๆ หนึ่ง ซึ่งได้พัฒนาเป็นโมดูลของคลาสที่มีจำนวนโมดูลเท่ากับจำนวนสถานะ การเปลี่ยนจากสถานะเดิมไปยังสถานะใหม่จะเป็นไปตามเหตุการณ์ของเครือข่ายที่เกิดขึ้น ซึ่งสามารถรับรู้เหตุการณ์ที่เกิดขึ้นได้จากโมดูลของคลาสที่ทำหน้าที่รับข้อความของเหตุการณ์จากส่วนบริการเครือข่ายและส่งต่อให้แก่แต่ละสถานะ เมื่อสถานะได้รับข้อความจะทำงานตามหน้าที่ที่ได้กำหนดไว้ซึ่งจะมีการเรียกใช้งานบริการเครือข่ายอีกต่อหนึ่งในการทำงาน จากรูปแบบทั่วไปของการพัฒนานี้ได้นำมาพัฒนาในส่วนเชื่อมต่อควบคุมแสดงในรูปของแผนภาพกระแสนิเวศข้อมูลดังแสดงในภาพประกอบ 3-13 ส่วนการพัฒนาส่วนเชื่อมต่อข้อมูลดังแสดงในภาพประกอบ 3-14

#### — ส่วนส่งถ่ายข้อมูลบนโปรโตคอล HTTP

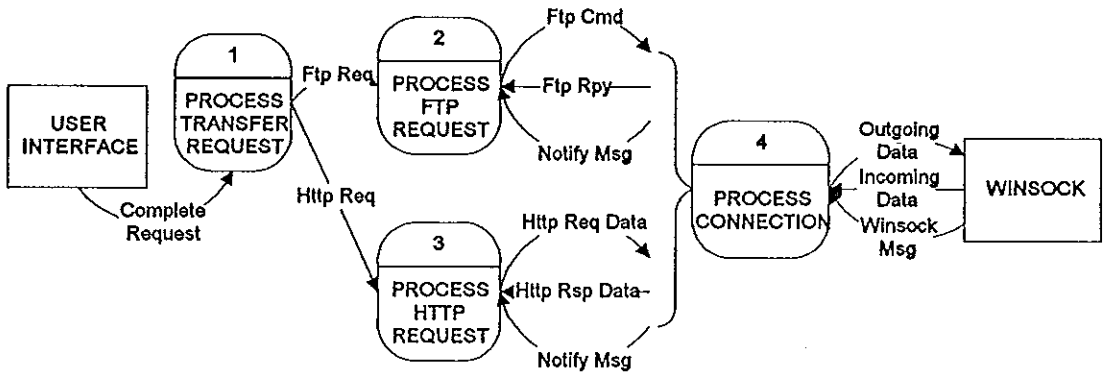
ดำเนินการพัฒนาด้วยวิธีเดียวกันกับการพัฒนาส่วนส่งถ่ายข้อมูลจากหัวข้อก่อนหน้านี้ โดยพัฒนาเป็นคลาสใหญ่เพียงคลาสเดียวที่มีกระบวนการทำงานที่สามารถแสดงเป็นแผนภาพกระแสนิเวศข้อมูลดังแสดงในภาพประกอบ 3-15

#### ● ส่วนบริการเครือข่าย

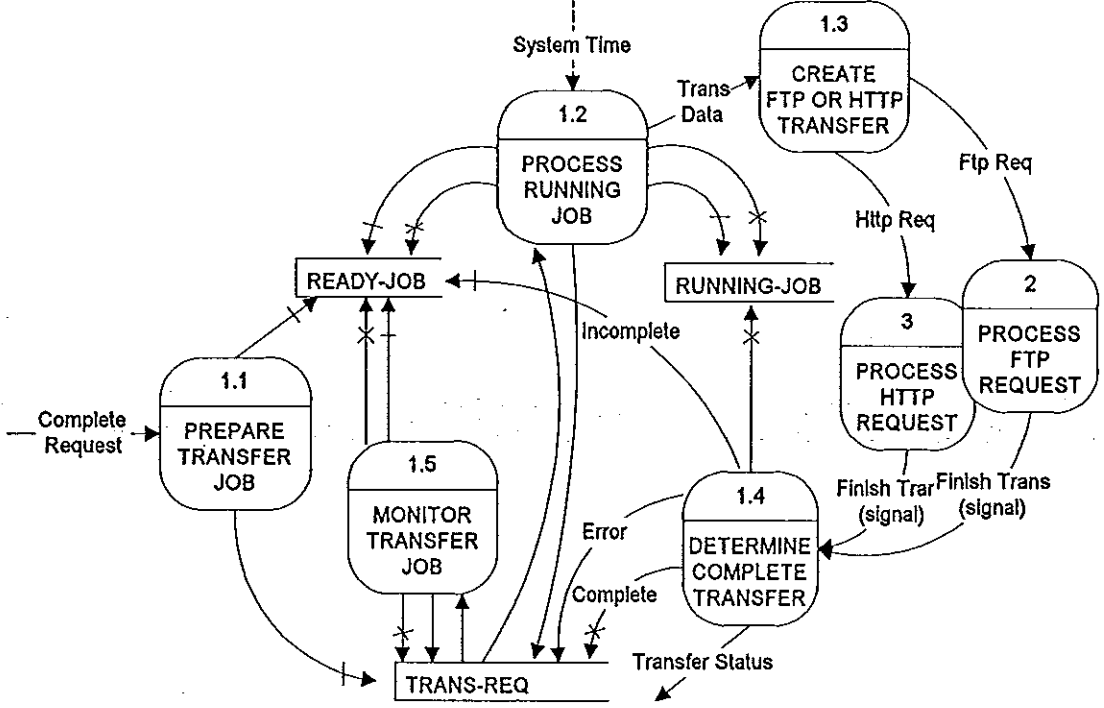
การดำเนินงานได้นำส่วนที่ออกไว้มานำมาพัฒนาได้เป็นคลาส 3 คลาสด้วยกันคือ คลาสที่ให้บริการค้นหาเลขที่อยู่ IP, คลาสที่ให้บริการด้านการสื่อสารที่ใช้ชื่อเกิด และคลาสที่ทำหน้าที่จัดส่งข้อความ ซึ่งแต่ละคลาสจะประกอบด้วยโมดูลที่ทำหน้าที่ตามที่ได้อธิบายไว้ในส่วนของการออกแบบ รายละเอียดของกระบวนการทำงานของแต่ละคลาสทั้งหมดนี้ที่ทำงานสัมพันธ์กันแสดงในรูปของของแผนภาพกระแสนิเวศข้อมูลดังแสดงในภาพประกอบ 3-16

### 3.2.2.2 กระบวนการทำงานของส่วนดำเนินงานของระบบ

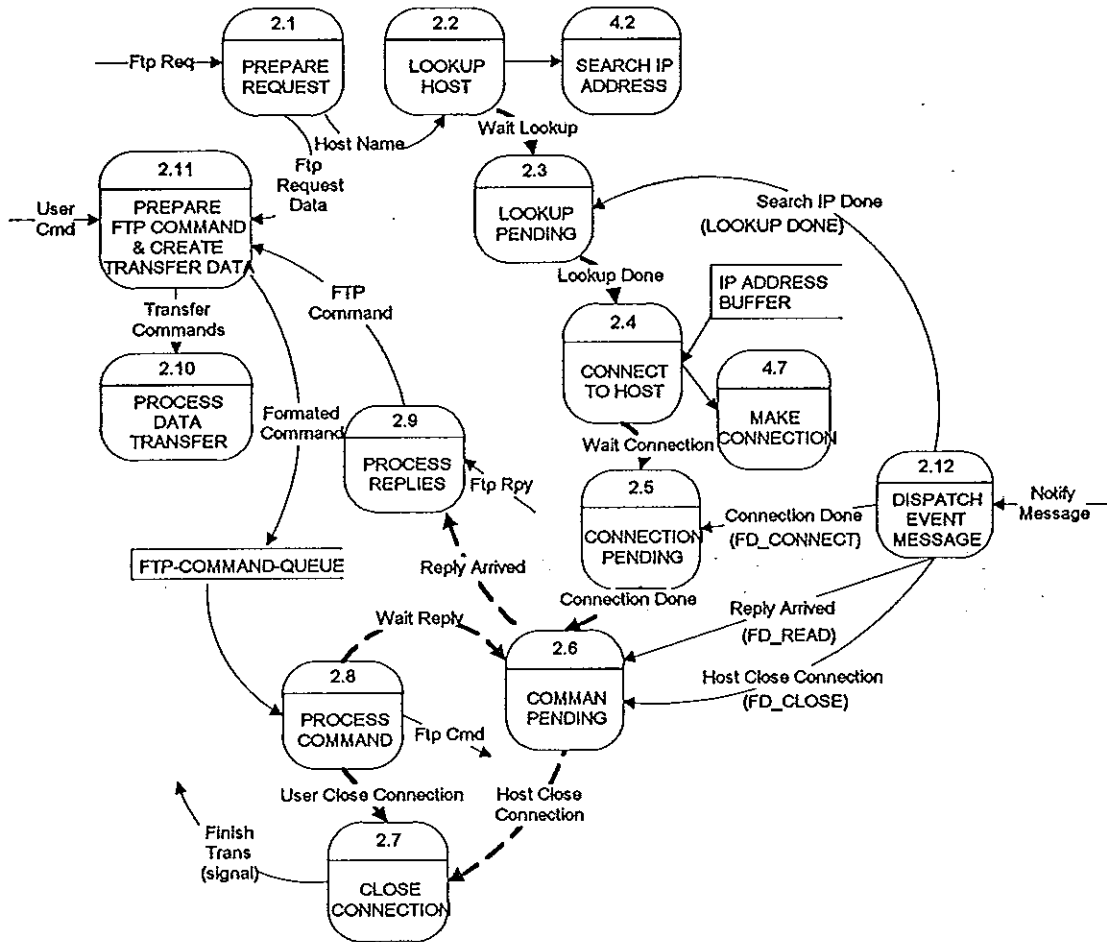
รายละเอียดการดำเนินงานของส่วนดำเนินงานของระบบสามารถแทนด้วยภาพกระแสข้อมูลดังแสดงในภาพประกอบ 3-11



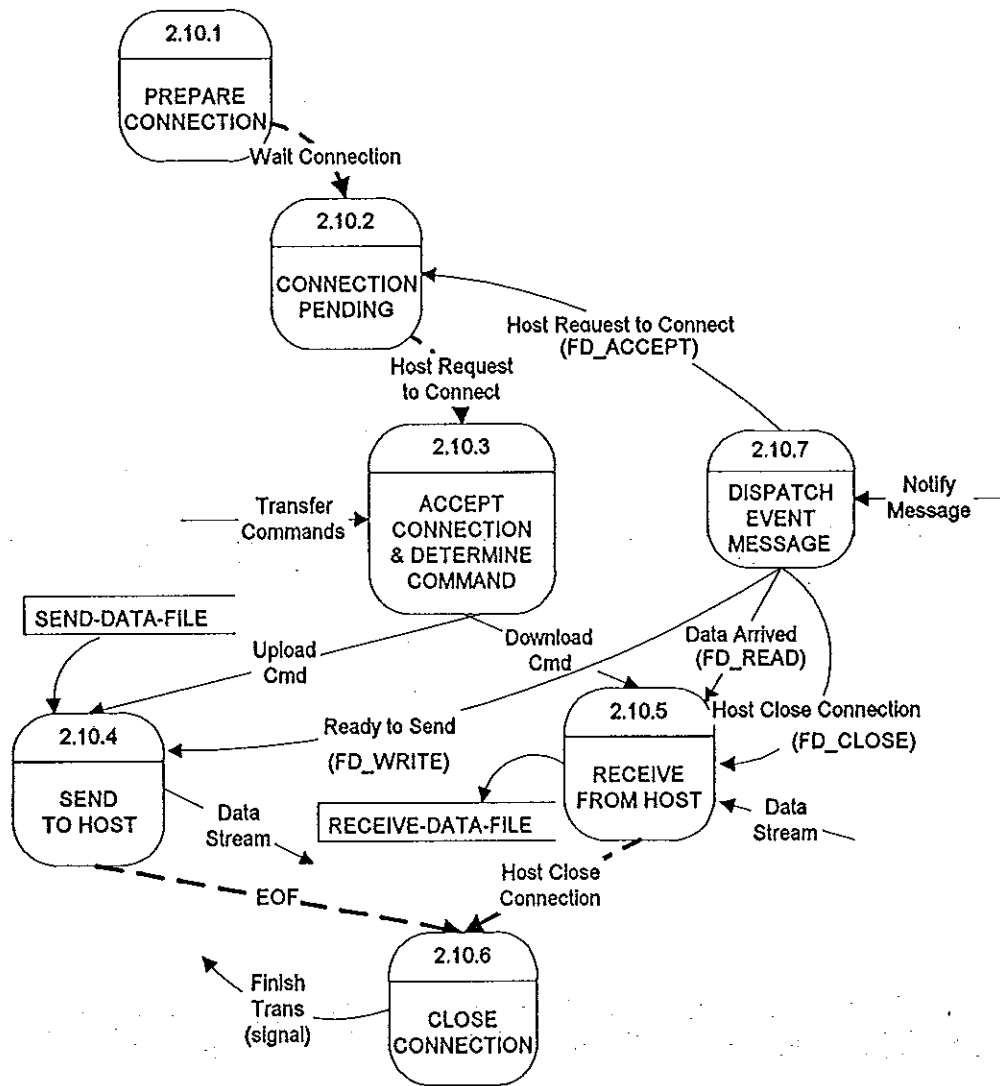
ภาพประกอบ 3-11 แทนภาพกระแสข้อมูลของระบบส่งถ่ายข้อมูล



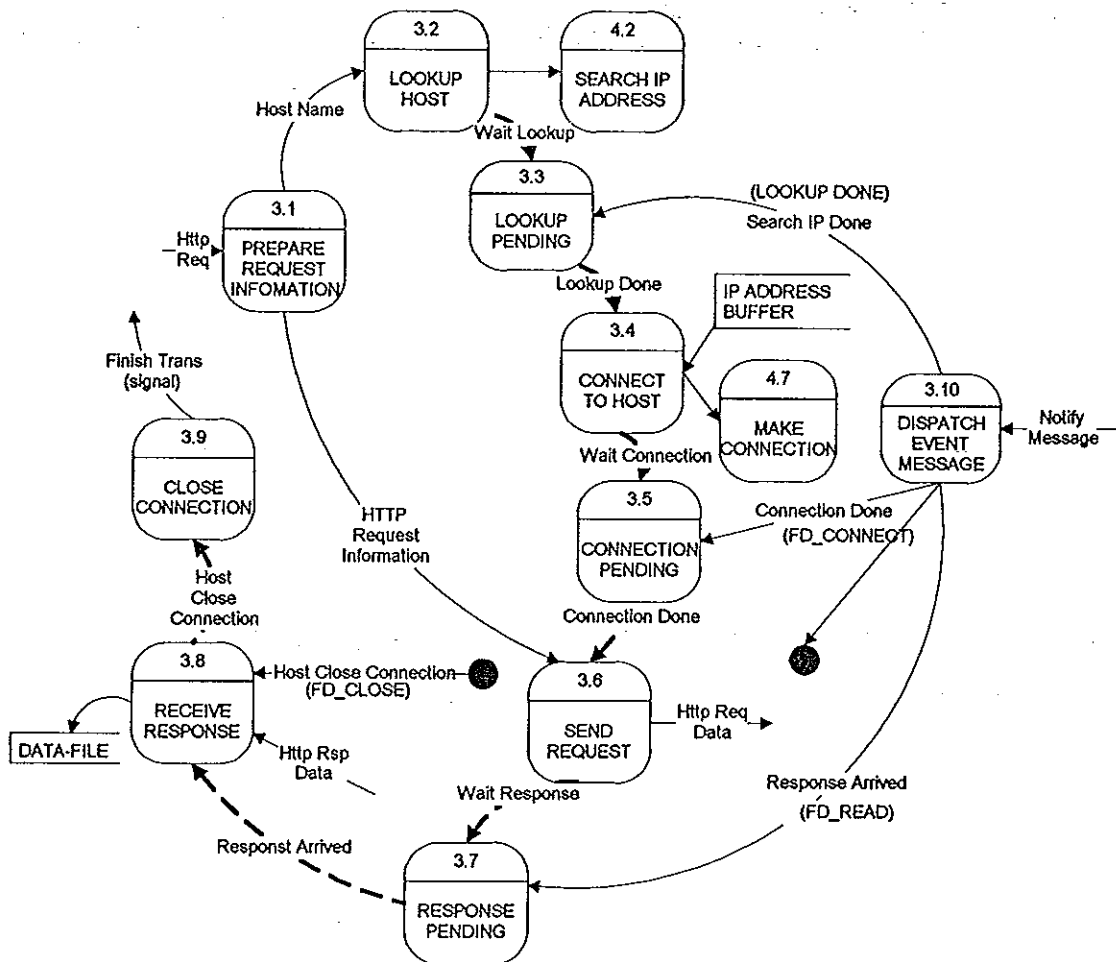
ภาพประกอบ 3-12 แทนภาพกระแสข้อมูลของกระบวนการ PROCESS TRANSFER REQUEST



ภาพประกอบ 8-13 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS FTP REQUEST

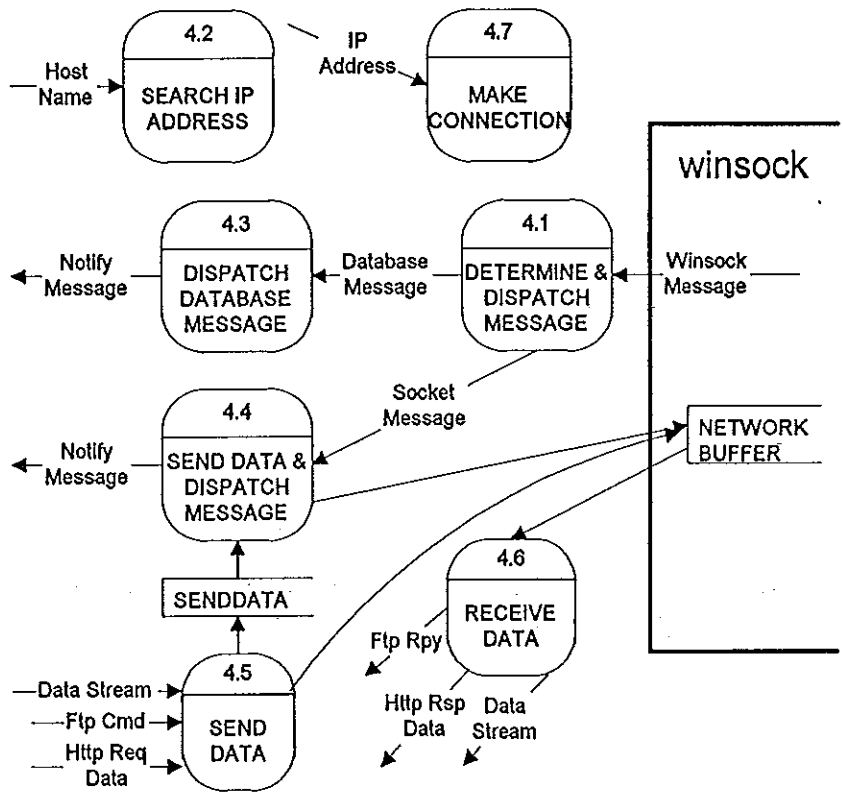


ภาพประกอบ 3-14 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS DATA TRANSFER



ภาพประกอบ 8-15 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS HTTP REQUEST





- 4.1 Dispatch Message Service จัดส่งข้อความจากวินซ็อกไปยังงานบริการฐานข้อมูลและงานบริการซ็อกเก็ต
- 4.2 - 4.3 Database Task Service ให้บริการค้นหาเลขที่อยู่ IP
- 4.4 - 4.7 Socket Task Service ให้บริการที่ต้องใช้ซ็อกเก็ต เช่น ส่งหรือรับข้อมูล สร้างการเชื่อมต่อ

ภาพประกอบ 3-16 แผนภาพกระแสข้อมูลของกระบวนการ PROCESS CONNECTION

จากภาพประกอบ 3-11 ถึง 3-16 สามารถอธิบายการดำเนินงานของแผนภาพได้ดังนี้

- คำอธิบายของกระบวนการ

จากภาพประกอบ 3-11

กระบวนการที่ 1

ชื่อกระบวนการ PROCESS TRANSFER REQUEST

คำอธิบาย ส่วนบริหารและจัดการงานส่งถ่ายข้อมูลจะรับข้อมูลจากส่วนติดต่อกับผู้ใช้และนำข้อมูลที่ได้ไปดำเนินการส่งถ่ายข้อมูล โดยถ้าหากข้อมูลที่ได้รับได้เป็นข้อมูลสำหรับการส่งถ่ายบนโปรโตคอล FTP จะส่งข้อมูลที่ได้ไปให้แก่กระบวนการที่ 2 แต่ถ้าหากเป็นโปรโตคอล HTTP จะส่งไปให้แก่กระบวนการที่ 4

กระบวนการที่ 2

ชื่อกระบวนการ PROCESS FTP REQUEST

คำอธิบาย ดำเนินการส่งถ่ายข้อมูลบนโปรโตคอล FTP ผ่านทางกระบวนการที่ 4

กระบวนการที่ 3

ชื่อกระบวนการ PROCESS HTTP REQUEST

คำอธิบาย ดำเนินการส่งถ่ายข้อมูลบนโปรโตคอล HTTP ผ่านทางกระบวนการที่ 4

กระบวนการที่ 4

ชื่อกระบวนการ PROCESS CONNECTION

คำอธิบาย ส่วนบริการเครือข่าย ให้บริการค้นหาเลขที่อยู่ IP , งานบริการสื่อสาร และบริการจัดส่งข้อความจากวินซ็อกไปยังกระบวนการที่ 2 และ 3

จากภาพประกอบ 3-12

กระบวนการที่ 1.1

ชื่อกระบวนการ PREPARE TRANSFER JOB

คำอธิบาย สร้างงานส่งถ่ายข้อมูลที่อยู่ในสถานะพร้อมที่จะทำงานไว้ใน READY-JOB และจัดเก็บสารสนเทศของงานไว้ใน TRANS-REQ

## กระบวนการที่ 1.2

ชื่อกระบวนการ PROCESS RUNNING JOB

คำอธิบาย กระบวนการนี้จะทำงานตามสัญญาณนาฬิกาของระบบ การทำงาน เริ่มจากย้ายงานส่งถ่ายข้อมูลที่ส่งถ่ายล่าช้าจาก RUNNING-JOB ไปยัง READY-JOB จากนั้นย้ายงานส่งถ่ายข้อมูลที่นำมาไว้ก่อนหน้านี้จาก READY-JOB ไปไว้ที่ RUNNING-JOB และสร้างการส่งถ่ายข้อมูลโดยส่งข้อมูลการส่งถ่ายไปให้แก่กระบวนการที่ 1.3

## กระบวนการที่ 1.3

ชื่อกระบวนการ CREATE FTP OR HTTP TRANSFER

คำอธิบาย สร้างการส่งถ่ายข้อมูลโดยใช้โปรโตคอล FTP หรือ HTTP

## กระบวนการที่ 1.4

ชื่อกระบวนการ DETERMINE COMPLETE TRANSFER

คำอธิบาย ตรวจสอบการส่งถ่ายข้อมูลว่าได้ดำเนินการส่งถ่ายข้อมูลเสร็จสมบูรณ์หรือไม่ โดยมีการดำเนินงานตามลำดับ ดังนี้

1. ย้ายงานส่งถ่ายข้อมูลออกจาก RUNNING-JOB
2. ถ้าส่งถ่ายข้อมูลไม่เสร็จ สร้างงานส่งถ่ายไว้ใน READY-JOB
3. ถ้าเกิดความผิดพลาดในการส่งถ่ายข้อมูล จะปรับปรุงสารสนเทศของงานส่งถ่ายใน TRANS-REQ ที่จะแสดงให้ผู้ใช้เห็นว่ามีความผิดพลาดเกิดขึ้น
4. ถ้าส่งถ่ายข้อมูลเสร็จสมบูรณ์ นำสารสนเทศของงานส่งถ่ายออกจาก TRANS-REQ

## กระบวนการที่ 1.5

ชื่อกระบวนการ MONITOR TRANSFER JOB

คำอธิบาย มีการทำงานตามที่อธิบายไว้ในหัวข้อ 3.1.1 ในหัวข้อของการออกแบบวินโดว์เฝ้าคุมการส่งถ่ายข้อมูล

## จากภาพประกอบ 8-18

กระบวนการที่ 2.1

ชื่อกระบวนการ PREPARE REQUEST

คำอธิบาย จัดเตรียมข้อมูลที่จะดำเนินการส่งถ่าย โดยส่งชื่อสถานีบริการไปให้แก่กระบวนการที่ 2.2 และส่งข้อมูลที่ต้องการดำเนินการส่งถ่ายไปให้แก่กระบวนการที่ 2.11

กระบวนการที่ 2.2

ชื่อกระบวนการ LOOKUP HOST

คำอธิบาย ค้นหาเลขที่อยู่ IP ของสถานีบริการ FTP และเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.3 เพื่อรอผลลัพธ์ของการค้นหาเลขที่อยู่ IP จากวินซ็อก

กระบวนการที่ 2.3

ชื่อกระบวนการ LOOKUP PENDING

คำอธิบาย รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าได้ค้นหาเลขที่อยู่ IP ของสถานีบริการสำเร็จแล้ว เมื่อได้รับข้อความดังกล่าวแล้วจะเปลี่ยนไปดำเนินงานไปยังกระบวนการที่ 2.4

กระบวนการที่ 2.4

ชื่อกระบวนการ CONNECT TO HOST

คำอธิบาย สร้างการเชื่อมต่อไปยังสถานีบริการ FTP และเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.5 เพื่อรอผลลัพธ์ของการดำเนินการเชื่อมต่อจากวินซ็อก

กระบวนการที่ 2.5

ชื่อกระบวนการ CONNECTION PENDING

คำอธิบาย รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าได้เชื่อมต่อกับสถานีบริการสำเร็จแล้ว เมื่อได้รับข้อความดังกล่าวแล้วจะเปลี่ยนไปดำเนินงานไปยังกระบวนการที่ 2.6

- กระบวนการที่ 2.6  
ชื่อกระบวนการ COMMAND PENDING  
คำอธิบาย รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกถึงสิ่งที่เกิดขึ้นดังนี้
1. สถานีบริการได้ส่งผลลัพธ์ของคำสั่งมาให้
  2. สถานีบริการได้ยกเลิกการเชื่อมต่อด้วยกัน
- ถ้าหากได้รับข้อความที่ระบุไว้ในข้อแรกจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.9 แต่ถ้าหากเป็นข้อความที่ระบุไว้ในข้อที่สองจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.7
- กระบวนการที่ 2.7  
ชื่อกระบวนการ CLOSE CONNECTION  
คำอธิบาย ยุติการเชื่อมต่อกับสถานีบริการ และส่งข้อความที่บ่งบอกการทำงานไปให้แก่กระบวนการที่ 1.4 ดังแสดงในภาพประกอบ 3-12
- กระบวนการที่ 2.8  
ชื่อกระบวนการ PROCESS COMMAND  
คำอธิบาย นำคำสั่งที่จัดเตรียมไว้ส่งไปให้สถานีบริการ โดยถ้าหากคำสั่งที่จัดเตรียมไว้เป็นคำสั่งยุติการเชื่อมต่อกับสถานีบริการจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.7 แต่ถ้าหากเป็นคำสั่งอื่น ๆ จะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.6
- กระบวนการที่ 2.9  
ชื่อกระบวนการ PROCESS REPLIES  
คำอธิบาย นำผลลัพธ์ที่ได้จากสถานีบริการมาประมวลผล ซึ่งอาจต้องมีการสร้างคำสั่งขึ้นมาเพื่อส่งให้แก่สถานีบริการต่ออีกครั้ง
- กระบวนการที่ 2.10  
ชื่อกระบวนการ PROCESS DATA TRANSFER  
คำอธิบาย กระบวนการส่งถ่ายแฟ้มข้อมูลกับสถานีบริการ

กระบวนการที่ 2.11

ชื่อกระบวนการ PREPARE FTP COMMAND & CREATE TRANSFER DATA

คำอธิบาย จัดเตรียมคำสั่งและสร้างกระบวนการส่งถ่ายข้อมูล โดยมีการดำเนินการดังนี้

1. ถ้าเป็นคำสั่งที่ไม่เกี่ยวข้องกับกำรส่งถ่ายข้อมูลจะจัดเตรียมคำสั่งเก็บไว้ในคิวคำสั่ง หรือไว้ใน FTP-COMMAND-QUEUE
2. ถ้าใช้คำสั่งในข้อที่ 1 จะสร้างกระบวนการส่งถ่ายข้อมูลและดำเนินการจัดเตรียมคำสั่งดังนี้

2.1. กรณีเป็นคำสั่งที่ต้องการส่งถ่ายข้อมูลจากสถานีบริการจะมีลำดับคำสั่งในการจัดเตรียมดังนี้

PORT -> TYPE -> REST -> RETR

2.2. กรณีเป็นคำสั่งที่ต้องการส่งถ่ายข้อมูลไปยังสถานีบริการจะมีลำดับคำสั่งในการจัดเตรียมดังนี้

PORT -> TYPE -> REST -> APPE

จากการใช้คำสั่ง REST ในข้อที่ 2 จะช่วยให้การส่งถ่ายข้อมูลที่ได้ยุติการส่งถ่ายกระทันหันก่อนที่จะส่งถ่ายเสร็จสมบูรณ์สามารถกลับมาส่งถ่ายข้อมูลอีกครั้งต่อจากเดิมได้

กระบวนการที่ 2.12

ชื่อกระบวนการ DISPATCH EVENT MESSAGE

คำอธิบาย จัดส่งข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่ได้รับจากส่วนบริการเครือข่ายไปให้แก่กระบวนการต่าง ๆ ตามที่ได้ระบุไว้ในแผนภาพกระแสข้อมูล

จากภาพประกอบ 8-14

กระบวนการที่ 2.10.1

ชื่อกระบวนการ PREPARE CONNECTION

คำอธิบาย จัดเตรียมการเชื่อมต่อจากสถานีบริการ FTP ที่จะมีเข้ามา และเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.10.2

- กระบวนการที่ 2.10.2**  
**ชื่อกระบวนการ CONNECTION PENDING**  
**คำอธิบาย** รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าการเชื่อมต่อจากสถานีบริการเข้ามา เมื่อได้รับข้อความดังกล่าวแล้วจะเปลี่ยนไปดำเนินการไปยังกระบวนการที่ 2.10.3
- กระบวนการที่ 2.10.3**  
**ชื่อกระบวนการ ACCEPT CONNECTION & DETERMINE COMMAND**  
**คำอธิบาย** ต้อนรับการเชื่อมต่อและจัดเตรียมการส่งถ่ายข้อมูล โดยพิจารณาคำสั่งของการส่งถ่ายข้อมูลที่ได้รับจากกระบวนการที่ 2.11 ว่าถ้าหากเป็นคำสั่งที่ต้องการส่งข้อมูลไปยังสถานีบริการจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.10.4 แต่ถ้าหากเป็นคำสั่งที่ต้องการรับข้อมูลจากสถานีบริการจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.10.5
- กระบวนการที่ 2.10.4**  
**ชื่อกระบวนการ SEND TO HOST**  
**คำอธิบาย** ดำเนินการส่งข้อมูลไปยังสถานีบริการ และเมื่อดำเนินการส่งข้อมูลเสร็จสมบูรณ์แล้วจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.10.6
- กระบวนการที่ 2.10.5**  
**ชื่อกระบวนการ RECEIVE FROM HOST**  
**คำอธิบาย** ดำเนินการรับข้อมูลจากสถานีบริการ และเมื่อดำเนินการรับข้อมูลเสร็จสมบูรณ์แล้วจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 2.10.6
- กระบวนการที่ 2.10.6**  
**ชื่อกระบวนการ CLOSE CONNECTION**  
**คำอธิบาย** ยุติการเชื่อมต่อการส่งถ่ายข้อมูลกับสถานีบริการ และส่งข้อความที่บ่งบอกการทำงานไปให้แก่กระบวนการที่ 1.4 ดังแสดงในภาพประกอบ 3-12

กระบวนการที่ 2.10.7  
 ชื่อกระบวนการ DISPATCH EVENT MESSAGE  
 คำอธิบาย จัดส่งข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่ได้รับจากส่วนบริการเครือข่าย  
 ไปให้แก่กระบวนการต่าง ๆ ตามที่ได้ระบุไว้ในแผนภาพกระแสข้อมูล

จากภาพประกอบ 8-16

กระบวนการที่ 3.1  
 ชื่อกระบวนการ PREPARE REQUEST INFORMATION  
 คำอธิบาย จัดเตรียมข้อมูลที่จะทำการส่งถ่าย โดยส่งชื่อสถานีบริการไปให้แก่กระบวนการ  
 ที่ 3.2 และส่งข้อมูลที่ต้องการดำเนินการส่งถ่ายไปให้แก่กระบวนการที่ 3.6

กระบวนการที่ 3.2  
 ชื่อกระบวนการ LOOKUP HOST  
 คำอธิบาย ค้นหาเลขที่อยู่ IP ของสถานีบริการ HTTP และเปลี่ยนการดำเนินงานไปยัง  
 กระบวนการที่ 3.3 เพื่อรอผลลัพธ์ของการค้นหาเลขที่อยู่ IP จากวินซ็อก

กระบวนการที่ 3.3  
 ชื่อกระบวนการ LOOKUP PENDING  
 คำอธิบาย รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าได้ค้นหาเลขที่อยู่ IP  
 ของสถานีบริการสำเร็จแล้ว เมื่อได้รับข้อความดังกล่าวแล้วจะเปลี่ยนไปดำเนิน  
 งานไปยังกระบวนการที่ 3.4

กระบวนการที่ 3.4  
 ชื่อกระบวนการ CONNECT TO HOST  
 คำอธิบาย สร้างการเชื่อมต่อไปยังสถานีบริการ HTTP และเปลี่ยนการดำเนินงานไปยัง  
 กระบวนการที่ 3.5 เพื่อรอผลลัพธ์ของการดำเนินการเชื่อมต่อจากวินซ็อก



- กระบวนการที่ 3.5**  
**ชื่อกระบวนการ** CONNECTION PENDING  
**คำอธิบาย** รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าได้เชื่อมต่อกับสถานีบริการสำเร็จแล้ว เมื่อได้รับข้อความดังกล่าวแล้วจะเปลี่ยนไปดำเนินการไปยังกระบวนการที่ 3.6
- กระบวนการที่ 3.6**  
**ชื่อกระบวนการ** SEND REQUEST  
**คำอธิบาย** ส่งคำร้องขอไปยังสถานีบริการ ซึ่งในการจัดเตรียมคำร้องขอนั้นจากข้อกำหนดของโปรโตคอล HTTP ในส่วนของ Header ของคำร้องขอที่ชื่อว่า Range จะช่วยให้การส่งถ่ายข้อมูลที่ได้ยุติการส่งถ่ายกระทันหันก่อนที่จะส่งถ่ายเสร็จสมบูรณ์สามารถกลับมาส่งถ่ายข้อมูลอีกครั้งต่อจากเดิมได้ และเมื่อดำเนินการส่งคำร้องขอเสร็จสมบูรณ์แล้วจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 3.7
- กระบวนการที่ 3.7**  
**ชื่อกระบวนการ** RESPONSE PENDING  
**คำอธิบาย** รอรับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าสถานีบริการได้ส่งข้อมูลตอบกลับมาให้แล้ว เมื่อได้รับข้อความดังกล่าวจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 3.8
- กระบวนการที่ 3.8**  
**ชื่อกระบวนการ** RECEIVE RESPONSE  
**คำอธิบาย** ดำเนินการรับข้อมูลตอบกลับจากสถานีบริการ และเมื่อดำเนินการรับข้อมูลเสร็จสมบูรณ์แล้วจะเปลี่ยนการดำเนินงานไปยังกระบวนการที่ 3.9
- กระบวนการที่ 3.9**  
**ชื่อกระบวนการ** CLOSE CONNECTION  
**คำอธิบาย** ยุติการเชื่อมต่อกับสถานีบริการ และส่งข้อความที่บ่งบอกการทำงานไปให้แก่กระบวนการที่ 1.4 ดังแสดงในภาพประกอบ 3-12

- กระบวนการที่ 3.10  
ชื่อกระบวนการ DISPATCH EVENT MESSAGE  
คำอธิบาย จัดส่งข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่ได้รับจากส่วนบริการเครือข่าย ไปให้แก่กระบวนการต่าง ๆ ตามที่ได้ระบุไว้ในแผนภาพกระแสข้อมูล
- จากภาพประกอบ 8-16
- กระบวนการที่ 4.1  
ชื่อกระบวนการ DETERMINE & DISPATCH MESSAGE  
คำอธิบาย รับข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่เกิดขึ้นจากวินซ็อกและส่งต่อไปให้ งานบริการฐานข้อมูล ( กระบวนการที่ 4.3 ) และงานบริการซ็อกเก็ต ( กระบวนการที่ 4.4 )
- กระบวนการที่ 4.2  
ชื่อกระบวนการ SEARCH IP ADDRESS  
คำอธิบาย ค้นหาเลขที่อยู่ IP
- กระบวนการที่ 4.3  
ชื่อกระบวนการ DISPATCH DATABASE MESSAGE  
คำอธิบาย จัดส่งข้อความต่อไปยังกระบวนการที่ 2 หรือ 3
- กระบวนการที่ 4.4  
ชื่อกระบวนการ SEND DATA & DISPATCH MESSAGE  
คำอธิบาย จัดส่งข้อมูลที่ค้างอยู่ใน SENDDATA ไปให้สถานีปลายทาง และจัดส่งข้อความต่อไปยังกระบวนการที่ 2 หรือ 3
- กระบวนการที่ 4.5  
ชื่อกระบวนการ SEND DATA  
คำอธิบาย ส่งข้อมูลไปยังสถานีปลายทาง แต่ถ้าไม่สามารถส่งออกไปได้ทันทีก็จะจัดเก็บไว้ในคิวค้างส่ง SENDDATA

กระบวนการที่ 4.6

ชื่อกระบวนการ RECEIVE DATA

คำอธิบาย รับข้อมูลจากสถานีปลายทาง

กระบวนการที่ 4.7

ชื่อกระบวนการ MAKE CONNECTION

คำอธิบาย สร้างการเชื่อมต่อกับสถานีปลายทาง

- คำอธิบายตัวจัดเก็บข้อมูล

FTP-COMMAND-QUEUE = \*จัดเก็บคำสั่งที่พร้อมที่จะส่งไปยังสถานีบริการ FTP\*

{FTP-COMMAND + FTP-COMMAND-PARAMETER}

READY-JOB = \*จัดเก็บสารสนเทศของงานส่งถ่ายที่พร้อมที่จะดำเนินการส่งถ่ายข้อมูล

มีโครงสร้างการจัดเก็บแบบ Linked List แต่เข้าถึงข้อมูลแบบคิว\*

{JOBID}

RUNNING-JOB = \*จัดเก็บสารสนเทศของงานส่งถ่ายที่กำลังดำเนินการส่งถ่ายข้อมูลอยู่

มีโครงสร้างการจัดเก็บแบบ Double Linked List\*

{JOBID + [FTP-JOB|HTTP-JOB]}

SENDDATA = \*จัดเก็บข้อมูลที่ไม่สามารถส่งไปให้แก่สถานีปลายทางได้เนื่องจากระบบเครือข่าย

ติดขัด มีโครงสร้างการจัดเก็บแบบ Linked List แต่เข้าถึงข้อมูลแบบคิว\*

{DATASTREAM + STREAM-LENGTH}

TRANS-REQ = \*แฟ้มข้อมูลที่จัดเก็บสารสนเทศของการส่งถ่ายข้อมูล โดยมีการจัดเก็บแบบเรียง

ลำดับ (Sequential)\*

{JOBID + PROTOCOL + TRANS-METHOD + URL + USER-NAME + PASSWORD +

LOCAL-FILENAME + JOBSTATE + WAITING-TIME + BYTEDONE + FILE-SIZE +

STARTUP-TIME }

- คำอธิบายของกระแสข้อมูล

Database Message =

LOOKUPDONE

Download Cmd =

[RETR|LIST]

Finish Trans = \*ข้อความที่บ่งบอกว่าได้ดำเนินการส่งถ่ายข้อมูลเสร็จแล้ว โดยข้อความนี้ได้จัดส่งผ่านทางระบบจัดส่งข้อความของระบบปฏิบัติการวินโดวส์ไปยังกระบวนการที่รอรับอยู่\*

Formatted Command = \*คำสั่งที่มีรูปแบบของคำสั่งสมบูรณ์\*

Ftp Cmd = \*อุหวัข้อ 2.1.3.1\*

FTP Command = \*คำสั่งที่มีรูปแบบยังไม่สมบูรณ์\*

Ftp Req =

TRANS-METHOD + URL + USER-NAME + PASSWORD + LOCAL-FILENAME +  
BYTEDONE

Ftp Request Data =

TRANS-METHOD + URL + USER-NAME + PASSWORD + LOCAL-FILENAME +  
BYTEDONE

Ftp Rpy = \*อุหวัข้อ 2.1.3.1\*

Host Name = \*ชื่อของสถานีบริการ ต.ย. เช่น www.psu.ac.th\*

Http Req =

URL + USER-NAME + PASSWORD + LOCAL-FILENAME + BYTEDONE

Http Req Data = \*อุหวัข้อ 2.1.3.2\*

HTTP Request Information =

URL + USER-NAME + PASSWORD + LOCAL-FILENAME + BYTEDONE

Http Rsp Data = \*อุหวัข้อ 2.1.3.2\*

Incoming Data =

DATASTREAM

IP Address = \* 192.100.77.3 \*

Notify Msg =

[Database Message|Socket Message]

Outgoing Data =

DATASTREAM

Socket Message =

[FD\_CONNECT|FD\_WRITE|FD\_READ|FD\_CLOSE|FD\_ACCEPT]

Trans Data =

JOBID + PROTOCOL + TRANS-METHOD + URL + USER-NAME + PASSWORD +  
LOCAL-FILENAME + WAITING-TIME + BYTEDONE + FILE-SIZE +  
STARTUP-TIME

Transfer Commands = \*คู่มือข้อ 2.1.3.2\*

[Download Cmd|Upload Cmd]

Upload Cmd =

[STOR|APPE]

User Cmd = \*คำสั่งที่มีรูปแบบยังไม่สมบูรณ์\*

Winsock Msg =

Notify Msg

● คำอธิบายของกระแสเหตุการณ์

Connection Done = \*เหตุการณ์ที่แสดงว่าได้เชื่อมต่อกับสถานีบริการแล้ว\*

EOF = \*เหตุการณ์ที่แสดงว่าได้อ่านข้อมูลจากแฟ้มที่กำหนดไว้ถึงตอนท้ายของแฟ้มแล้ว\*

Host Close Connection = \*เหตุการณ์ที่แสดงว่าสถานีบริการได้ยุติการเชื่อมต่อด้วย\*

Host Request to Connection = \*เหตุการณ์ที่แสดงว่าสถานีบริการต่อการเชื่อมต่อด้วย\*

Lookup Done = \*เหตุการณ์ที่แสดงว่าวินซ็อกได้ค้นหาเลขที่อยู่ IP เสร็จแล้ว\*

Reply Arrived = \*เหตุการณ์ที่แสดงว่าสถานีบริการ FTP ได้ส่งผลลัพธ์ของคำสั่ง FTP มาให้แล้ว\*

Response Arrived = \*เหตุการณ์ที่แสดงว่าสถานีบริการ HTTP ได้ส่งคำตอบมาให้แล้ว\*

User Close Connection = \*เหตุการณ์ที่แสดงว่าผู้ใช้ได้ยุติการเชื่อมต่อกับสถานีบริการ\*

Wait Connection = \*เหตุการณ์ที่แสดงว่ากำลังรอผลลัพธ์ของการเชื่อมต่อกับสถานีบริการ\*

Wait Lookup = \*เหตุการณ์ที่แสดงว่ากำลังรอผลลัพธ์ของการค้นหาเลขที่อยู่ IP\*

Wait Reply = \*เหตุการณ์ที่แสดงว่ากำลังรอผลลัพธ์ของคำสั่ง FTP จากสถานีบริการ\*

Wait Response = \*เหตุการณ์ที่แสดงว่ากำลังรอคำตอบจากสถานีบริการ HTTP\*

• คำอธิบายหน่วยของข้อมูล

- APPE — คำอธิบายหน่วยของข้อมูล Upload Cmd
- BYTEDONE — จำนวนไบต์ข้อมูลที่ได้รับหรือส่งไปเรียบร้อยแล้ว
- DATASTREAM — ข้อมูลเป็นไบต์เรียงติดต่อกัน
- Download Cmd — คำสั่งของโปรโตคอล FTP ที่ใช้ในการส่งถ่ายข้อมูลมาจากสถานีบริการ ดูหัวข้อ 2.1.3.1
- Download-Dir-Name — ใส่ชื่อไดเรกทอรีที่จัดจะเก็บแฟ้มข้อมูลที่ส่งถ่ายมาจากสถานีบริการ
- E-mail — เลขที่อยู่อีเมลล์สำหรับเป็นรหัสผ่านสำหรับการส่งถ่ายข้อมูลโดยโปรโตคอล FTP
- Editor-Name — ชื่อโปรแกรมเอดิเตอร์ (Editor)
- FD\_ACCEPT — เหตุการณ์ทางเครือข่ายที่บ่งบอกว่ามีสถานีงานอื่นขอเชื่อมต่อด้วย
- FD\_CLOSE — เหตุการณ์ทางเครือข่ายที่บ่งบอกว่าสถานีปลายทางยุติการเชื่อมต่อด้วย
- FD\_CONNECT — เหตุการณ์ทางเครือข่ายที่บ่งบอกว่าสามารถเชื่อมต่อกับสถานีปลายทางได้แล้ว
- FD\_READ — เหตุการณ์ทางเครือข่ายที่บ่งบอกว่าสถานีปลายทางส่งข้อมูลมาให้
- FD\_WRITE — เหตุการณ์ทางเครือข่ายที่บ่งบอกว่าสามารถส่งข้อมูลไปยังสถานีปลายทางได้
- FILE-SIZE — ขนาดของแฟ้มข้อมูลที่ทำกรรับหรือส่ง
- FTP-COMMAND — ดูหัวข้อ 2.1.3.1
- FTP-COMMAND-PARAMETER — ดูหัวข้อ 2.1.3.1
- FTP-JOB — งานส่งถ่ายข้อมูลที่ใช้โปรโตคอล FTP
- HTTP-JOB — งานส่งถ่ายข้อมูลที่ใช้โปรโตคอล HTTP
- JOBID — หมายเลขของงานที่ส่งถ่ายข้อมูล
- JOBSTATE — สถานะของงาน หรือกระบวนการของการส่งถ่ายข้อมูลที่เป็นอยู่ในขณะใดขณะหนึ่งซึ่งมีอยู่ 3 สถานะด้วยกันคือ
- SUBMIT — ไม่มีการส่งถ่ายข้อมูล
  - READY — พร้อมที่จะทำการส่งถ่ายข้อมูล
  - RUNNING — กำลังส่งถ่ายข้อมูลอยู่
- LIST — คำอธิบายหน่วยของข้อมูล Download Cmd
- LOCAL-FILENAME — ชื่อแฟ้มข้อมูลที่ใช้ในการรับ หรือส่งข้อมูลซึ่งขึ้นอยู่กับ
- TRANS-METHOD
- LOOKUPDONE — เหตุการณ์ทางเครือข่ายที่บ่งบอกว่าค้นหาเลขที่อยู่ IP ได้แล้ว
- Max-Connection — จำนวนสูงสุดของงานส่งถ่ายข้อมูลที่สามารถเชื่อมต่อกับสถานีบริการได้

**PASSWORD** — รหัสผ่านของผู้ใช้

**PROTOCOL** — ชนิดของโปรโตคอลที่เลือกใช้ (Ftp / Http)

**Recv-Bytes** — จำนวนขนาดของบัฟเฟอร์ (Buffer) ที่ใช้เก็บข้อมูลที่ได้รับมาจากสถานีบริการ

**RETR** — คำอธิบายหน่วยของข้อมูล Download Cmd

**Scan-Time** — จำนวนความดีเป็นวินาทีในการตรวจสอบงานส่งถ่ายข้อมูลหลายงานที่กำลังเชื่อมต่ออยู่กับสถานีบริการ

**Send-Bytes** — จำนวนขนาดของบัฟเฟอร์ที่เก็บข้อมูลที่ใช้ส่งไปยังสถานีบริการ

**STARTUP-TIME** — เวลาที่ต้องการให้เริ่มทำการส่งถ่ายข้อมูล

**STOR** — คำอธิบายหน่วยของข้อมูล Upload Cmd

**STREAM-LENGTH** — ขนาดของ DATASTREAM

**TRANS-METHOD** — รูปแบบการดำเนินงานของการส่งถ่ายข้อมูลว่าจะเป็นการรับ (Download) หรือ ส่ง (Upload)

**Upload Cmd** — คำสั่งของโปรโตคอล FTP ที่ใช้ในการส่งถ่ายข้อมูลไปยังสถานีบริการ ดูหัวข้อ

2.1.3.1

**URL** — ค่า URL ที่ใช้งาน

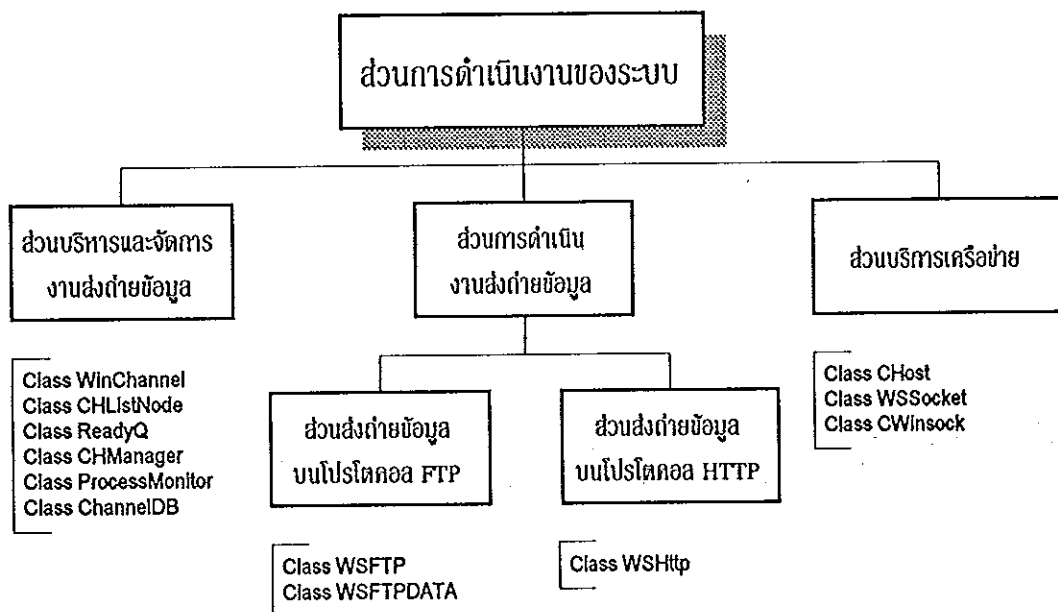
**USER-NAME** — ชื่อของผู้ใช้ที่จะเข้าใช้สถานีบริการ

**WAITING-TIME** — ช่วงเวลาที่กำหนดให้สามารถรอการรับหรือส่งแพ็คเกจข้อมูลอันถัดไป โดยเริ่มจับเวลาที่รอภายหลังที่ได้รับหรือส่งแพ็คเกจข้อมูลล่าสุดไปแล้ว ช่วงเวลาที่รอนี้จะอยู่ในช่วง 1-60 นาที

**WWW-Proxy-Name** — ชื่อของสถานีบริการหรือกซึ่ สำหรับการส่งถ่ายข้อมูลโดยโปรโตคอล HTTP

3.2.2.3 ส่วนของโปรแกรม

การพัฒนาในส่วนการดำเนินงานของระบบนี้ได้ดำเนินการสร้างเป็นคลาสขึ้นมาซึ่งสามารถแสดงในรูปของผังโครงสร้างดังแสดงในภาพประกอบ 3-17



ภาพประกอบ 3-17 ผังโครงสร้างคลาสของส่วนการดำเนินงานของระบบ

จากภาพประกอบ 3-17 สามารถอธิบายการพัฒนาและหน้าที่ของแต่ละคลาสของแต่ละส่วนการดำเนินงานของระบบแสดงดังตาราง 3-2 ดังนี้

ตาราง 3-2 คลาสของส่วนการดำเนินงานของระบบ

คลาส	คำอธิบาย
WinChannel	เป็นคลาสที่สร้างขึ้นโดยกำหนดให้มีคุณสมบัติเป็นวินโดว์ หน้าของคลาสคือ ทำหน้าที่ประหนึ่งเป็นงานส่งถ่ายข้อมูลงานหนึ่งที่กำลังดำเนินการส่งถ่ายข้อมูลอยู่ ซึ่งได้ถูกจัดเก็บไว้ในคลาส CHListNode และได้ใช้คุณสมบัติของการเป็นวินโดว์แสดงสารสนเทศของการส่งถ่ายที่กำลังดำเนินอยู่ ภายหลังจากที่ได้ส่งถ่ายข้อมูลเสร็จแล้วจะตรวจสอบผลลัพธ์ของการส่งถ่ายว่าเสร็จสมบูรณ์หรือไม่ ถ้าไม่สมบูรณ์จะดำเนินการส่งถ่ายข้อมูลใหม่อีกครั้ง
CHListNode	เป็นคลาสที่ทำหน้าที่จัดเก็บงานส่งถ่ายข้อมูลหลาย ๆ งานที่อยู่ในสถานะกำลังส่งถ่ายข้อมูลอยู่ โดยมีโครงสร้างการจัดเก็บแบบ Double Linked List



ตาราง 3-2 (ต่อ) คลาสของส่วนการดำเนินงานของระบบ

คลาส	คำอธิบาย
ReadyQ	เป็นคลาสที่ทำหน้าที่จัดเก็บงานส่งถ่ายข้อมูลหลาย ๆ งานที่อยู่ในสถานะพร้อมที่จะทำการส่งถ่ายข้อมูล โดยมีโครงสร้างการจัดเก็บแบบ Linked List และมีการเข้าถึงข้อมูลที่จัดเก็บแบบคิว
CHManager	เป็นคลาสที่ทำหน้าที่สร้างงานส่งถ่ายข้อมูล และบริหารและจัดการงานส่งถ่ายข้อมูลที่สร้างขึ้นตามที่ได้อธิบายการพัฒนาไว้ในหัวข้อ 3.2.2.1
ProcessMonitor	เป็นคลาสที่สร้างขึ้นโดยกำหนดให้มีคุณสมบัติเป็นวินโดว์และมีรายการเมนู หน้าทีของคลาสคือ <ol style="list-style-type: none"> <li>ใช้คุณสมบัติของการเป็นวินโดว์แสดงงานส่งถ่ายข้อมูลทั้งหมดที่เกิดขึ้นที่ถูกรับและจัดการโดยคลาส CHManager โดยข้อมูลของงานที่นำมาแสดงจะได้อาจมาจากคลาส ChannelDB</li> <li>ติดต่อกับผู้ใช้ผ่านรายการเมนูเพื่อให้ผู้ใช้ปรับปรุงข้อมูลการส่งถ่ายที่แสดงไว้ในวินโดว์</li> <li>ติดต่อกับผู้ใช้ผ่านรายการเมนูเพื่อให้ผู้นำข้อมูลการส่งถ่ายที่ได้ดำเนินงานค้างไว้ไปดำเนินการสร้างงานส่งถ่ายข้อมูลต่อจากเดิม</li> </ol>
ChannelDB	เป็นคลาสที่ทำหน้าที่เข้าใช้งานแฟ้มข้อมูล hftp.dbf ซึ่งได้จัดเก็บข้อมูลของงานส่งถ่ายข้อมูลทั้งหมดที่เกิดขึ้น
WSFTP	เป็นคลาสที่พัฒนาด้วยวิธีการกลไกสถานะที่ประกอบด้วยโมดูลของคลาสทำหน้าที่เป็นสถานะหนึ่ง ๆ ของกลไกสถานะ และแต่ละโมดูลนี้จะทำงานร่วมกันในการส่งถ่ายข้อมูล โดยมีการติดต่อกับเครือข่ายผ่านทางคลาสจากส่วนบริการเครือข่ายอีกต่อหนึ่ง จากหัวข้อที่ 3.2.2.1 คลาสนี้จะทำหน้าที่เป็นส่วนเชื่อมต่อควบคุม
WSFTPDATA	คลาสนี้ได้พัฒนาขึ้นด้วยวิธีการเดียวกันกับคลาส WSFTP หน้าทีของคลาสนี้จะทำหน้าที่เป็นส่วนเชื่อมต่อข้อมูล
WSHttp	คลาสนี้ได้พัฒนาขึ้นด้วยวิธีการเดียวกันกับสองคลาสก่อนหน้านี้
Chost	ประกอบด้วย โมดูลของคลาสที่มีการทำงานดังต่อไปนี้ <ol style="list-style-type: none"> <li>บริการค้นหาเลขที่อยู่ IP โดยเรียกใช้วินซ็อก API กลุ่มฐานข้อมูลช่วยในการค้นหา</li> <li>จัดส่งข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่บ่งบอกว่าได้ดำเนินการค้นหาเลขที่อยู่ IP เสร็จแล้ว โดยข้อความที่ได้รับจะได้อาจมาจากคลาส Cwinsock และจัดส่งต่อไปให้แก่คลาสที่ทำหน้าที่ส่งถ่ายข้อมูลต่อไป</li> </ol>

ตาราง 8-2 (ต่อ) คลาสของส่วนการดำเนินงานของระบบ

คลาส	คำอธิบาย
WSSocket	ประกอบด้วยโมดูลของคลาสที่เรียกใช้วินซ็อก API กลุ่มซ็อกเก็ตเพื่อช่วยในการทำงาน ซึ่งแต่ละโมดูลมีการทำงานดังต่อไปนี้ <ol style="list-style-type: none"> <li>1. บริการสร้างการเชื่อมต่อไปยังเครื่องคอมพิวเตอร์เครื่องอื่น</li> <li>2. บริการจัดส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ที่เชื่อมต่อด้วย</li> <li>3. บริการรับข้อมูลจากเครื่องคอมพิวเตอร์ที่เชื่อมต่อด้วย</li> <li>4. จัดส่งข้อความที่ระบุถึงเหตุการณ์ทางเครือข่ายที่สัมพันธ์กับการใช้บริการของคลาสจากหัวข้อที่ 1-4 โดยคลาสที่ส่งข้อความมาให้และคลาสที่จะจัดส่งต่อไปให้เป็นเช่นเดียวกันกับโมดูลของคลาส Chost ที่ระบุไว้ในข้อที่ 2</li> </ol>
CWinsock	รับข้อความของเหตุการณ์ทางเครือข่ายจากวินซ็อก และส่งต่อไปให้แก่คลาส Chost และ WSSocket

### 3.3 การทดสอบระบบ

การทดสอบระบบงานได้ดำเนินการแบ่งออกเป็นสองส่วนคือ การทดสอบทั่วไป และการทดสอบประสิทธิภาพ

#### 3.3.1 การทดสอบทั่วไป

เป็นการทดสอบการทำงานของโปรแกรมทุกส่วนการใช้งานแต่ส่วนที่มีการเน้นทดสอบมากที่สุดมีดังนี้

- ทดสอบการรับส่งข้อมูลภายใต้กฎเกณฑ์ต่อไปนี้
  1. ทดสอบในขณะที่มีความหนาแน่นของเครือข่ายต่าง ๆ กัน
  2. ทดสอบโดยกำหนดระยะห่างระหว่างเครื่องคอมพิวเตอร์เป็นเกณฑ์
  3. ทดสอบการรับส่งเพิ่มข้อมูลที่มีขนาดต่าง ๆ กัน
- ทดสอบความคงทนของโปรแกรม วิธีการทดสอบจะป้อนข้อมูลประเภทที่จะส่งผลกระทบต่อการทำงานของโปรแกรมให้ทำงานผิดพลาด เช่น กำหนดชื่อของสถานีปลายทางที่ไม่ตัวตนจริงให้แก่โปรแกรม

ผลการทดสอบทั่วไป สรุปแล้วเป็นที่น่าพอใจ แต่ก็ไม่สามารถคาดเดาได้ว่าโปรแกรมจะทำงานได้ถูกต้องในสภาวะอื่น ๆ ที่อยู่นอกเหนือจากกฎเกณฑ์การทดสอบที่กำหนดไว้

#### 3.3.2 การทดสอบประสิทธิภาพ

- จุดประสงค์ของการทดสอบ

โดยทำการทดสอบในเรื่องความสัมพันธ์ระหว่างขนาดของบัฟเฟอร์ของโปรแกรมที่หักข้อมูลก่อนที่จะส่งหรือหักข้อมูลที่รับมาต่อความเร็วของการส่งถ่ายข้อมูล และระหว่างขนาดของบัฟเฟอร์ของระบบเครือข่ายต่อความเร็วของการส่งถ่ายข้อมูล ขนาดบัฟเฟอร์ของโปรแกรมสามารถควบคุมผ่านวินโดว์กำหนดสภาวะแวดล้อมของโปรแกรมของระบบดังแสดงในภาคผนวก ง ในภาพประกอบ ง-2 ส่วนขนาดของบัฟเฟอร์ของระบบเครือข่ายจะควบคุมขนาดผ่านทางโปรแกรม Trumpet Winsock

● แผนการทดสอบ

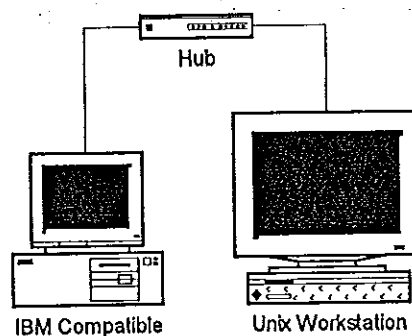
แบ่งการทดสอบอัตราเร็วในการรับส่งข้อมูลออกเป็น 16 ส่วน ซึ่งสามารถแสดงแผนการทดสอบในรูปของตารางดังแสดงในตาราง 3-3

ตาราง 3-3 แสดงแผนในการทดสอบโปรแกรม

		Program Buffer Size (bytes)			
		1460	2920	4380	5840
Network Buffer Size (bytes)	1460	1	5	9	13
	2920	2	6	10	14
	4380	3	7	11	15
	5840	4	8	12	16

จากตาราง 3-3 การทดสอบในแต่ละครั้งแบ่งออกเป็นการทดสอบรับข้อมูลและส่งข้อมูล โดยที่การทดสอบแต่ละอย่างจะทำการทดสอบ 3 ครั้งแล้วหาค่าเฉลี่ยออกมา สำหรับปัจจัยที่มีผลกระทบต่อกรทดสอบมีดังนี้

- Maximum Transmission Unit (MTU) กำหนดให้มีค่าเท่ากับ 576 ไบต์
- TCP Maximum Segment Size (MSS) กำหนดให้มีค่าเท่ากับ 512 ไบต์
- ขนาดของแฟ้มข้อมูลที่ทำกรทดสอบกำหนดให้มีขนาดเท่ากับ 519763 ไบต์
- Network Traffic = เป็นของการทดสอบนี้เท่านั้น
- เส้นทางการทดสอบ



ภาพประกอบ 3-18 แสดงเส้นทางของการทดสอบโปรแกรม

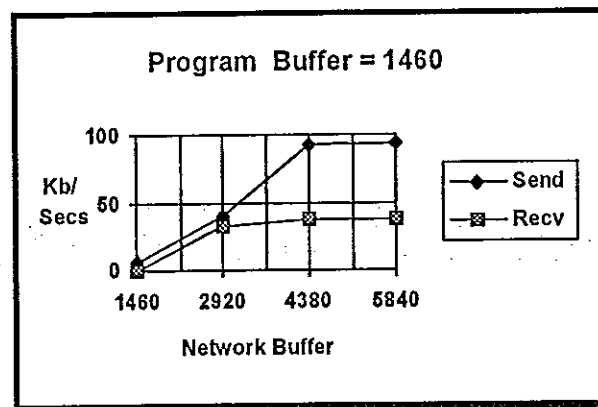
- ผลการทดสอบ

สามารถสรุปผลการทดสอบในรูปของตารางดังแสดงในตาราง 3-4

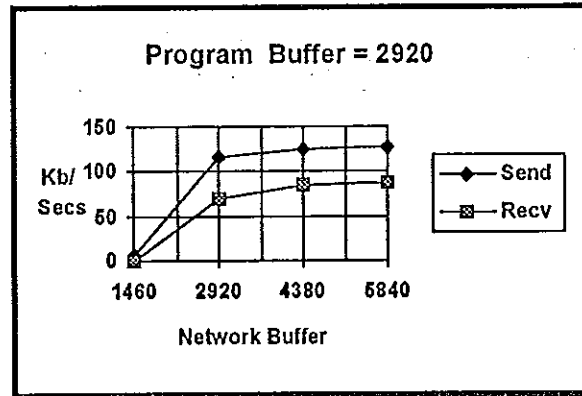
ตาราง 3-4 แสดงอัตราเร็วในการรับส่งข้อมูล (Kb/Secs).

		Program Buffer Size (bytes)			
		1460	2920	4380	5840
Network Buffer Size (bytes)	1460 Send	6.53	6.53	6.53	6.53
	Receive	0.10	0.10	0.10	0.10
	2920 Send	40.72	115.56	130.55	135.76
	Receive	32.23	69.99	91.58	91.69
	4380 Send	92.46	124.71	141.61	151.70
	Receive	37.82	84.32	93.14	118.13
	5840 Send	93.44	127.36	145.19	158.20
	Receive	37.98	88.17	103.93	122.25

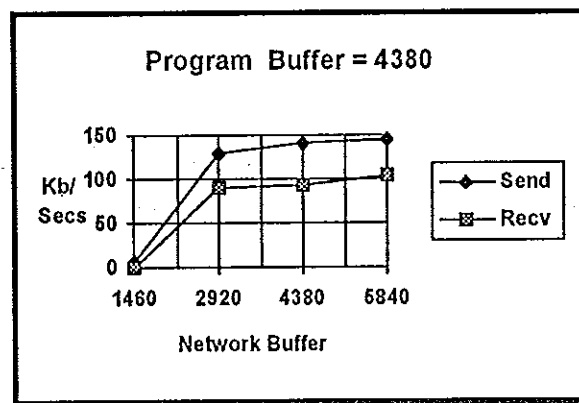
จากข้อมูลในตารางสามารถสรุปเป็นกราฟแสดงความสัมพันธ์ ระหว่างอัตราเร็วในการรับส่งข้อมูลกับขนาดของเน็ตเวิร์กบัฟเฟอร์โดยที่มีขนาดของโปรแกรมบัฟเฟอร์คงที่ ดังแสดงในภาพประกอบ 3-19 ถึงภาพประกอบ 3-22



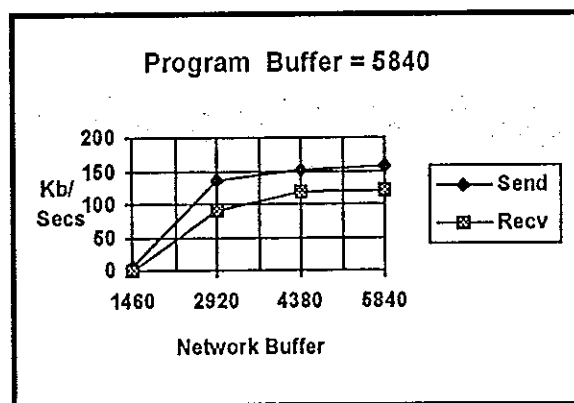
ภาพประกอบ 3-19 กราฟสรุปผลการทดสอบครั้งที่ 1-4



ภาพประกอบ 8-20 กราฟสรุปผลการทดสอบครั้งที่ 5-8



ภาพประกอบ 8-21 กราฟสรุปผลการทดสอบครั้งที่ 9-12



ภาพประกอบ 8-22 กราฟสรุปผลการทดสอบครั้งที่ 13-16

● สรุปผลการทดสอบ

จากการทดสอบสรุปผลได้สองประการด้วยกันคือ ประการแรกการเพิ่มขนาดของโปรแกรมบีทไฟเฟอร์ให้ใหญ่กว่า 4380 ไบต์ จะมีผลต่อการเพิ่มขึ้นของอัตราความเร็วของการส่งถ่ายข้อมูลน้อยมาก ประการที่สองคือ ขนาดของเน็ตเวิร์กบีทไฟเฟอร์และขนาดของโปรแกรมบีทไฟเฟอร์จะแปรผันโดยตรงกับอัตราความเร็วในการรับส่งข้อมูล นอกจากนี้แล้วอัตราความเร็วในการรับข้อมูลจะช้ากว่าการส่งข้อมูลในขณะที่อยู่ในสถานะแวดล้อมเดียวกัน จากการสังเกตระหว่างการทดสอบสามารถสรุปผลกระทบทที่ทำให้อัตราความเร็วในการรับช้ากว่าได้ดังนี้

— เกิดจากการสูญเสียเวลาในการจัดเก็บข้อมูลจากโปรแกรมบีทไฟเฟอร์ลงสู่แฟ้มข้อมูล

— เกิดจากขนาดของแพ็คเกจที่สถานีบริการ FTP ใช้ในการส่งข้อมูลมีขนาดเล็กทำให้ต้องเสียเวลาในการส่งข้อมูลหลาย ๆ ครั้ง

— เกิดจากจำนวนของข้อความที่วินซ็อกส่งมาให้ระบบงานให้ทำการอ่านข้อมูลที่เก็บอยู่ในเน็ตเวิร์กบีทไฟเฟอร์ มีจำนวนข้อความมากกว่าข้อความที่วินซ็อกส่งมาให้ระบบงานให้ทำการส่งข้อมูลไปยังเน็ตเวิร์กบีทไฟเฟอร์ การนับจำนวนข้อความทำได้โดยใช้ดีบั๊กเกอร์ (Debugger) ของโปรแกรมแปลภาษา Borland C++ 4.5

## บทที่ 4

### สรุป ปัญหา และข้อเสนอแนะ

ในบทนี้ กล่าวถึงข้อสรุปและผลที่ได้จากดำเนินการวิจัย และกล่าวถึงปัญหาตลอดจนอุปสรรคที่เกิดขึ้นในขณะทำการวิจัย ในหัวข้อสุดท้ายเป็นการให้ข้อเสนอแนะแก่ผู้ที่สนใจที่จะนำงานวิจัยไปพัฒนาต่อไป

#### 4.1 สรุปผลการวิจัย

งานวิทยานิพนธ์นี้ได้ศึกษาและพัฒนาระบบส่งถ่ายข้อมูลบนโปรโตคอล FTP และ HTTP บนเครือข่ายอินเทอร์เน็ต โดยมีลำดับขั้นการดำเนินงานโดยย่อดังนี้ ขั้นแรกเริ่มจากการศึกษาถึงโครงข่ายอินเทอร์เน็ตโดยทั่ว ๆ ไป ขั้นที่สองศึกษาถึงการพัฒนาโปรแกรมบนเครือข่ายโดยใช้วินโดวส์ซึ่งเกิดเป็นเครื่องมือช่วยในการเชื่อมต่อกับเครือข่าย ขั้นที่สามเป็นการออกแบบพัฒนาโปรแกรมของงานวิทยานิพนธ์นี้ โดยใช้วิธีการโปรแกรมแบบเชิงวัตถุเป็นเครื่องมือ ซึ่งก่อให้เกิดความสะดวกในการพัฒนาและง่ายต่อการแก้ไขปรับปรุง ขั้นตอนสุดท้ายคือ ตรวจสอบประสิทธิภาพและความถูกต้องของโปรแกรม โดยสรุปงานที่ได้จากการทำวิทยานิพนธ์มีดังนี้

1. ได้โปรแกรมส่งถ่ายข้อมูลที่ทำงานได้บนโปรโตคอลถึงสองชนิด ที่มีความน่าเชื่อถือและมีประสิทธิภาพในการส่งถ่ายข้อมูลค่อนข้างดี
2. ได้โปรแกรมที่สะดวกต่อการนำไปใช้พัฒนาโปรแกรมอื่น ๆ ต่อไป
3. สามารถนำโปรแกรมที่ได้ไปใช้ประกอบการเรียนการสอนในเนื้อหาบางส่วนของวิชาที่เกี่ยวข้องกับการสื่อสารข้อมูลบนเครือข่ายได้

#### 4.2 ปัญหาและอุปสรรค

1. เวลาที่ใช้ในการทดสอบโปรแกรมบนเครือข่าย เมื่อนำมารวม ๆ กันแล้วจะสิ้นเปลืองเวลามาก
2. เติชญ์กับปัญหา 2 อย่างพร้อม ๆ กัน คือ การพัฒนาส่วนของโปรแกรมที่เกี่ยวข้องกับระบบปฏิบัติการวินโดวส์ และ การพัฒนาส่วนของโปรแกรมที่เกี่ยวข้องกับเครือข่าย
3. โปรแกรมอรรถประโยชน์ที่ช่วยในการพัฒนามีข้อผิดพลาดในที่นี้คือ โปรแกรม Trumpet Winsock ความผิดพลาดที่เกิดขึ้นของเวอร์ชัน 1.0 จะมีมากกว่าเวอร์ชัน 3.0
4. การทำงานของโปรแกรมอรรถประโยชน์ที่ช่วยในการพัฒนาในที่นี้คือ โปรแกรม TracePlus/Winsock จะใช้เวลาส่วนหนึ่งในการทำงานซึ่งจะมีผลกระทบต่อเวลาที่ใช้ในการพัฒนา



5. การพัฒนาโปรแกรมบนเครือข่ายเป็นสิ่งที่ยากอยู่ในระดับหนึ่งคือ จะต้องสร้างโปรแกรมที่ทำหน้าที่เป็นสถานีบริการและโปรแกรมที่ทำหน้าที่เป็นสถานีลูกข่ายไปพร้อม ๆ กัน

#### 4.8 ข้อเสนอแนะ

1. การพัฒนาโปรแกรมบนเครือข่ายมีสิ่งที่เกี่ยวข้องกับการพัฒนาอยู่มาก เช่น โปรแกรมอรรถประโยชน์ที่ใช้ตรวจสอบการทำงานทางเครือข่าย และโปรแกรมอรรถประโยชน์ที่ช่วยในการพัฒนา และโปรแกรมอรรถประโยชน์อื่น ๆ ในบางครั้งเมื่อทำการทดสอบการทำงานของโปรแกรมที่พัฒนาและเกิดความผิดพลาดขึ้นจะต้องพิจารณาถึงสาเหตุที่เกิดขึ้นอย่างรอบครอบ เพราะมีสิ่งที่เกี่ยวข้องที่เชื่อมโยงกันอยู่มาก จึงมีความจำเป็นที่จะต้องตรวจสอบการทำงานของโปรแกรมอรรถประโยชน์ที่ใช้งาน เพื่อช่วยลดภาระในการตรวจสอบหาสาเหตุของความผิดพลาดให้น้อยลง

2. โปรแกรมดีบั๊กเกอร์ของโปรแกรมแปลภาษา Borland C++ 4.5 มีความสามารถไม่เพียงพอต่อการตรวจสอบส่วนของโปรแกรมที่เกี่ยวข้องกับเครือข่าย ดังนั้นผู้ที่จะนำระบบงานนี้ไปพัฒนาอาจใช้โปรแกรมแปลภาษาโปรแกรมอื่นที่มีความสามารถในการตรวจสอบส่วนของโปรแกรมจากที่กล่าวมาได้

3. ก่อนที่จะพัฒนาระบบงานบนเครือข่ายนั้นจะต้องออกแบบ หรือจัดหาโปรโตคอลมารองรับการพัฒนากระบวนการก่อนเสมอ

## บรรณานุกรม

- เกษมสันต์ พานิชกร. 2538. C++ และหลักการของ OOP ฉบับเริ่มต้น. กรุงเทพฯ : บริษัทซีเอ็ด  
ยูเคชั่น จำกัด.
- ขรรข เต็งอำนาจ. 2521. ระบบปฏิบัติการ (OPERATING SYSTEMS). กรุงเทพฯ : บริษัทซีเอ็ด  
ยูเคชั่น จำกัด.
- Berners-Lee, T.; Fielding, R. and Frystyk, H. 1996. HyperText Transfer Protocol (HTTP.1.0)  
Specification (RFC 1945).  
<http://www.w3.org/hypertext/WWW/Protocols/HTTP1.0/draft-ietf-http-spec.html>.
- Comer, Douglas. 1991. Internetworking with TCP/IP. Volumn I. Singapore : Simon &  
Schuster Asia Pte.
- Dilascia, Paul. 1992. Windows++ Writeing Reusable Windows Code in C++. 2d ed. New  
York : Addison Wesley.
- Hall, Martin. 1993. A Guide to Windows Sockets. s.l. : JSB Corporation.
- Morphet, John. 1995. Windows On The Internet. New York : McGraw-Hill.
- Roberts, Dave. 1995. Developing For The Internet With Winsock. Arizona : CORIOLIS  
GROUP BOOKS.
- Towfig, Mark., et al. 1994. FAQ About Windows Sockets Version 1.1.  
<ftp://SunSite.UNC.EDU/pub/micro/pc-stuff/ms-windows/winsoc/FAQ>.
- Postel, J. B. and Reynolds, J. K. 1985. RFC 959, File Transfer Protocol (FTP).
- Hall, Martin. et al. 1992. An Open Interface for Network Programming under Microsoft  
Windows (WINDOWS SOCKETS) Version 1.1.

## ภาคผนวก ก

### วินโดวส์ซ็อกเก็ต API

วินโดวส์ซ็อกเก็ต API ประกอบด้วย 4 ฟังก์ชันด้วยกัน แต่มีเพียงส่วนน้อยเท่านั้นที่ใช้บ่อย ฟังก์ชันเหล่านี้สามารถแบ่งออกให้เป็น 4 กลุ่มด้วยกันคือ กลุ่มเปลี่ยนแปลงค่า (Conversion), กลุ่มฐานข้อมูล (Database), กลุ่มซ็อกเก็ต (Socket) และ กลุ่มวินซ็อกเก็ตเพิ่มเติม (Winsock extensions) กลุ่มเปลี่ยนแปลงค่าจะช่วยเปลี่ยนแปลงค่าของลำดับไบนารีของเลขจำนวนเต็มระหว่างโฮสต์กับเน็ตเวิร์ก และช่วยเปลี่ยนแปลงค่าเลขที่อยู่ IP ระหว่างรูปแบบที่เป็นสายอักขระ (String) กับรูปแบบเชิงตัวเลข (Numeric) กลุ่มฐานข้อมูลจะช่วยโปรแกรมในการค้นหาสารสนเทศเกี่ยวกับโฮสต์, บริการของเน็ตเวิร์กและโปรโตคอลจากเน็ตเวิร์กคาด้าเบส (Database) ฟังก์ชันฐานข้อมูลที่ใช้กันบ่อยคือ ฟังก์ชันที่ช่วยโปรแกรมในการค้นหาเลขที่อยู่ IP ของสถานีบริการที่กำหนดจากผู้ใช้ กลุ่มซ็อกเก็ตใช้ในการสร้างการเชื่อมต่อ, รับและส่งข้อมูล และยุติการเชื่อมต่อเมื่อไม่มีความจำเป็นในการใช้งานอีกต่อไป กลุ่มสุดท้าย จะมีความแตกต่างจากกลุ่มอื่น ๆ กลุ่มนี้ได้ออกแบบให้โปรแกรมวินซ็อกเก็ตทำงานประสานเข้ากับการทำงานของระบบปฏิบัติการแบบหลายงาน (Multitasking) ของวินโดวส์ ฟังก์ชันกลุ่มนี้จะใช้การส่งข้อความในการบ่งบอกถึงความสำเร็จในการดำเนินงานของฟังก์ชัน เพื่อเป็นการหลีกเลี่ยงไม่ให้โปรแกรมถูกขัด (Blocking) รายละเอียดการทำงานของฟังก์ชันแต่ละกลุ่มมีดังนี้

ตาราง ก-1 แสดงกลุ่มฟังก์ชันเปลี่ยนแปลงค่า

Function	Description
htonl	Converts a long (32-bit) quantity form host to network byte order
htons	Converts a short (16-bit) quantity form host to network byte order
inet_addr	Converts an address string in dotted-decimal format to a 32-bit IP address
inet_ntoa	Converts a 32-bit IP address to a dotted-decimal string
ntohl	Converts a long (32-bit) quantity form network to host byte order
ntohs	Converts a long (16-bit) quantity form network to host byte order

ตาราง ก-2 แสดงกลุ่มฟังก์ชันฐานข้อมูล

Function	Description
gethostbyaddr	Retrieves host information according to the host IP address
gethostbyname	Retrieves host information according to the host name
gethostname	Retrieves the name of the local host
getprotobyname	Retrieves protocol information according to a familiar protocol name
getprotobynumber	Retrieves protocol information according to the specified protocol number
getservbyname	Retrieves service information according to a familiar service name
getservbyport	Retrieves service information according to the service port number

ตาราง ก-3 แสดงกลุ่มฟังก์ชันซ็อกเก็ต

Function	Description
accept	Accepts a connection on a listening socket and returns a newly created, connected socket
bind	Binds a local interface address and port number to a socket
closesocket	Closes a socket and releases the socket descriptor
connect	Establishes a connection with a remote host
getpeername	Gets the address and port information of the remote host to which a socket is connected
getsockname	Gets the local interface address and port information for a socket
getsockopt	Gets the value of a local socket option
ioctlsocket	Gets or sets the operation parameters of a socket
listen	Indicates a socket that should listen for incoming connections
recv	Receives data from a socket
recvfrom	Receives data from a socket and returns the remote host addressing information
select	Determines the readability, writability, and error status for one or more sockets
send	Sends data using a connected socket
sendto	Sends data to a specific remote host address and port number
setsockopt	Sets the value of a local socket option
shutdown	Disables the sending or receiving of data on a socket
socket	Creates a socket

ตาราง ก-4 แสดงกลุ่มฟังก์ชันวินซ็อกขยายเพิ่มเติม

Function	Description
WSAAsyncGetHostByAddr	Asynchronously retrieves host information using the host address as a key
WSAAsyncGetHostByName	Asynchronously retrieves host information using the host name as a key
WSAAsyncGetProtoByName	Asynchronously retrieves protocol information using the name of the protocol as a key
WSAAsyncGetProtoByNumber	Asynchronously retrieves protocol information using the number of the protocol as a key
WSAAsyncGetServByName	Asynchronously retrieves service information using the name of the service as a key
WSAAsyncGetServByPort	Asynchronously retrieves service information using the port number of the service as a key
WSAAsyncSelect	Requests notification of events pertaining to a given socket
WSACancelAsyncRequest	Cancels an in-progress asynchronous database operation
WSACancelBlockingCall	Cancels an in-progress function that is blocked in a blocking hook loop
WSACleanup	Terminates the use of Windows Sockets
WSAGetLastError	Gets the specific error code of the last Winsock error generated by this thread
WSAIsBlocking	Returns TRUE if a blocking operation is in progress
WSASetBlockingHook	Installs an application-specific blocking hook
WSASetLastError	Sets the error code that will be retrieved by WSAGetLastError
WSAStartup	Initiates startup of Windows Sockets
WSAUnhookBlockingHook	Restores the blocking hook to the Windows Sockets default blocking hook

ภาคผนวกนี้ แสดงเฉพาะหน้าที่การทำงานของฟังก์ชันวินซ็อก API เท่านั้น การใช้งานจริง ๆ ของแต่ละฟังก์ชันสามารถเปิดอ่านได้จากข้อกำหนดของวินโดวส์ซ็อกเก็ต หรือ จากแฟ้ม winsock.hlp

## ภาคผนวก ข

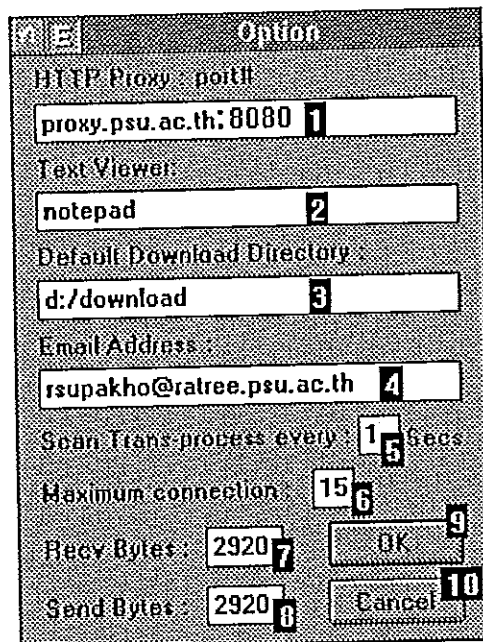
### คู่มือการติดตั้งและการใช้งาน

- ขั้นตอนการติดตั้ง

1. โปรแกรมนี้ใช้งานได้กับระบบปฏิบัติการวินโดวส์ 3.1 ,วินโดวส์ 95 และวินโดวส์ NT
2. ติดตั้งฮาร์ดแวร์และซอฟต์แวร์ที่จำเป็นต้องใช้ในการเชื่อมต่อกับอินเทอร์เน็ต
3. ติดตั้งโปรแกรม Trumpet Winsock ให้ทำตามคู่มือที่มาพร้อมกับซอฟต์แวร์
4. ถัดลอกแฟ้ม HFTP.EXE และ BC450RTL.DLL มายังไดเรกทอรีที่ต้องการติดตั้ง

- การกำหนดสถานะแวดล้อมของโปรแกรมก่อนการใช้งาน

การกำหนดสถานะแวดล้อมนี้ โปรแกรมจะกำหนดให้ผู้ใช้กระทำทันทีหลังจากติดตั้งโปรแกรมครั้งแรก หรือผู้ใช้อาจจะกำหนดสถานะแวดล้อมใหม่ในภายหลังก็ได้ ซึ่งสามารถกระทำได้โดยเลือกปุ่มหมายเลข 26 จากภาพประกอบ ข-2 จะได้วินโดว์ดังแสดงในภาพประกอบ ข-1



ภาพประกอบ ข-1 วินโดว์กำหนดสถานะแวดล้อมของโปรแกรม

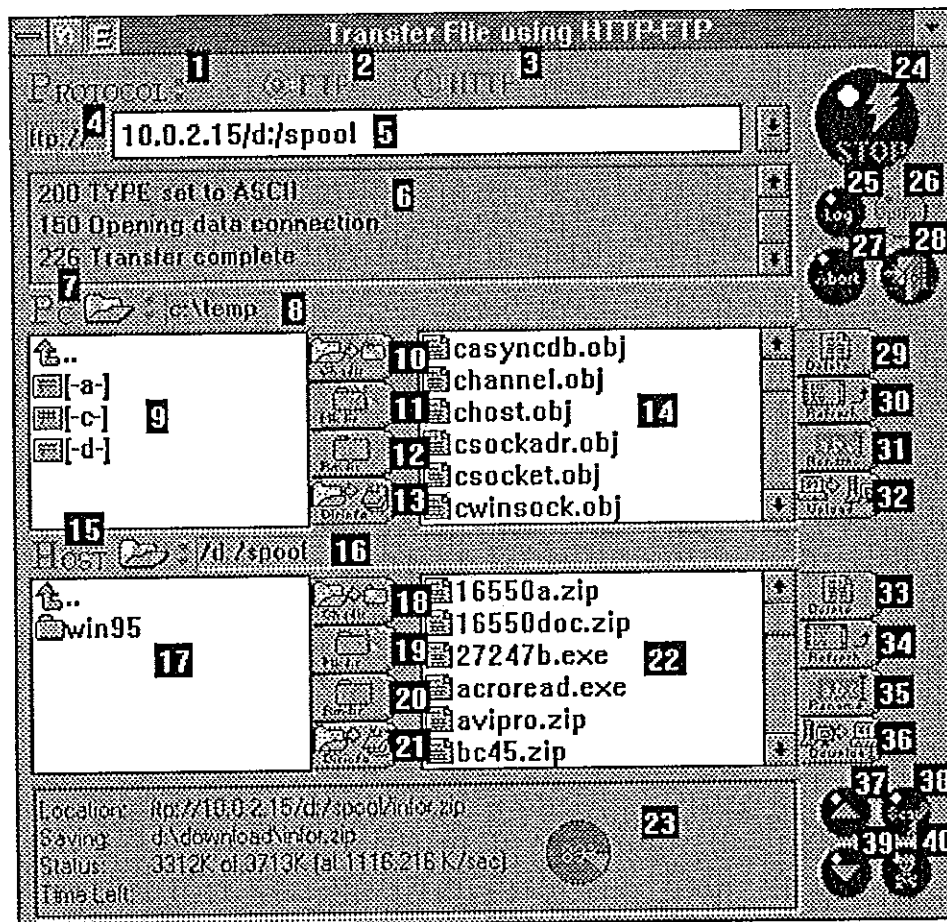
จากภาพประกอบ ข-1 สามารถอธิบายการใช้งานตามหมายเลขได้ดังนี้

1. ใช้เมื่อเครือข่ายที่กำลังใช้งานอยู่นั้นกำหนดให้การส่งถ่ายข้อมูลโดยใช้โปรโตคอล HTTP จะต้องกระทำผ่านสถานีบริการหรือกซ์ ซึ่งวิธีกำหนดข้อมูลดังนี้  
[ ชื่อสถานีบริการหรือกซ์ : หมายเลขพอร์ต ]
2. ให้ใส่ชื่อของโปรแกรมบรรณาธิการ (Edit Program) ไว้ใช้สำหรับดูแลผลลัพธ์ของการส่งถ่ายข้อมูลที่เกิดขึ้น
3. ถ้าต้องการให้เพิ่มข้อมูลที่รับมา (Download) เก็บไว้ในไดเรกทอรีที่แน่นอนให้ใส่ชื่อไดเรกทอรีไว้
4. กำหนดให้ใช้อิมัลเนชันเป็นรหัสผ่านในการเข้าใช้สถานีบริการ FTP ในกรณีที่ได้กำหนดชื่อผู้ใช้เท่ากับ anonymous
5. ความถี่ในการตรวจสอบงานส่งถ่ายข้อมูลที่กำลังดำเนินอยู่ว่ามีงานใดที่เกิดความล่าช้าในการส่งถ่าย
6. จำนวนสูงสุดของงานส่งถ่ายข้อมูลที่สามารถเชื่อมต่อกับสถานีบริการได้
7. ขนาดของบัฟเฟอร์ที่ใช้ในการรับข้อมูล (1460-5840 ไบต์)
8. ขนาดของบัฟเฟอร์ที่ใช้ในการส่งข้อมูล (1460-5840 ไบต์)
9. กดปุ่มนี้เพื่อนำข้อมูลตั้งแต่หมายเลข 1 - 8 ไปเก็บไว้ในแฟ้ม win.ini
10. กดปุ่มนี้เมื่อต้องการยกเลิกการดำเนินงานที่ผ่านมาทั้งหมด

• การส่งถ่ายข้อมูลผ่านวินโดว์หลัก

1. เรียกใช้โปรแกรม HFTP.EXE ขึ้นมาใช้งานจะปรากฏวินโดว์หลักดังแสดงในภาพประกอบ

ข-2



ภาพประกอบ ข-2 วินโดว์หลัก

2. เมื่อต้องการส่งถ่ายข้อมูลด้วยโปรโตคอล FTP ให้ปฏิบัติดังนี้

2.1. เลือกปุ่มหมายเลข 2

2.2. เลือกช่องใส่ข้อความหมายเลข 5 และใส่ค่า URL ซึ่งมีรูปแบบดังนี้

host [ ":" port ][ abs\_path ]

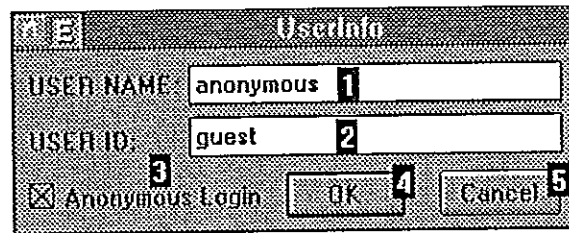
host = ชื่อของสถานีบริการ FTP

port = หมายเลขพอร์ตที่ต้องการติดต่อด้วยไม่จำเป็นต้องใส่ถ้าใช้พอร์ตหมายเลข 21

abs\_path = ไคเรกทอรีที่ต้องการเข้าใช้งาน



2.3. เลือกปุ่มหมายเลข 24 จะปรากฏวินโดว์ให้ใส่ชื่อและรหัสผ่านของผู้ใช้ดังแสดงในภาพประกอบ ข-3 ให้ใส่ชื่อและรหัสผ่าน ถ้าต้องการใช้ชื่อผู้ใช้เท่ากับ anonymous ให้เลือกหมายเลข 3 เมื่อบันทึกข้อมูลครบแล้วให้เลือกปุ่มหมายเลข 4 เพื่อเข้าใช้งานสถานีบริการ หรือเลือกปุ่มหมายเลข 5 เพื่อยกเลิกข้อมูลที่กำหนดไว้ทั้งหมด

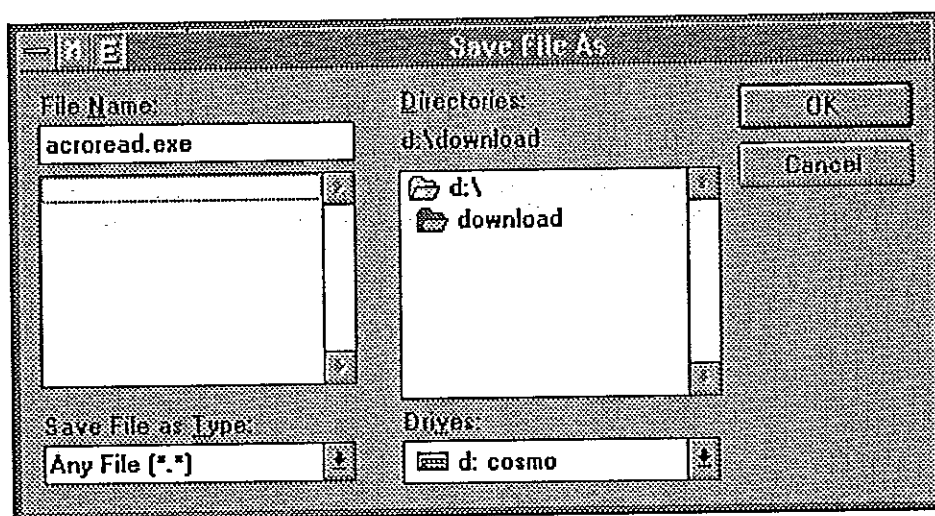


ภาพประกอบ ข-3 วินโดว์ให้ใส่ชื่อและรหัสผ่าน

2.4. เมื่อสามารถเข้าใช้งานได้แล้ว วินโดว์หมายเลข 6 จะแสดงข้อความที่มีการติดต่อระหว่างกัน และที่หมายเลข 16, 17 และ 22 จะแสดงระบบแก้ไขข้อมูลปัจจุบันของสถานีบริการ ส่วนระบบแก้ไขข้อมูลปัจจุบันของสถานีงานที่กำลังใช้งานอยู่จะแสดงที่หมายเลข 8, 9 และ 14

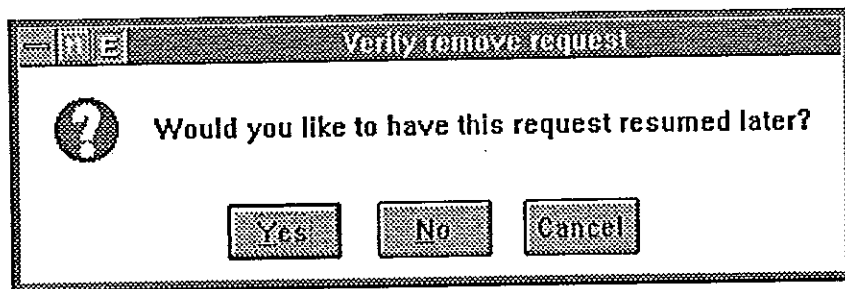
2.5. เมื่อต้องการส่งแก้ไขข้อมูล (Upload) ให้เลือกชื่อแก้ไขที่ต้องการส่งจากหมายเลข 14 และเลือกปุ่มหมายเลข 32 เพื่อทำการส่ง

2.6. เมื่อต้องการรับแก้ไขข้อมูล (Download) ให้เลือกชื่อแก้ไขที่ต้องการรับจากหมายเลข 22 และเลือกปุ่มหมายเลข 36 เพื่อทำการรับ โปรแกรมจะแสดงวินโดว์ให้ใส่ชื่อแก้ไขที่ต้องการจัดเก็บไว้ ดังแสดงในภาพประกอบ ข-4 จะสังเกตเห็นได้ว่าตำแหน่งของไดเรกทอรีคือ "download" จะเท่ากับที่ได้กำหนดไว้ในวินโดว์ดังแสดงในภาพประกอบ ข-1 หมายเลข 3



ภาพประกอบ ข-4 วินโดว์ใส่ชื่อแก้ไขที่ต้องการจัดเก็บ

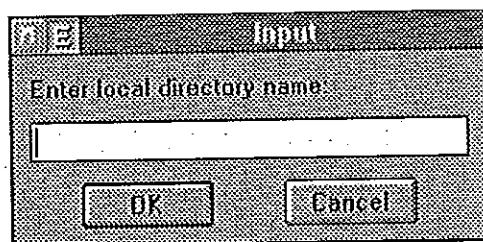
2.7. การแสดงการทำงานของกระบวนการส่งถ่ายข้อมูลที่ได้เชื่อมต่อกับสถานีปลายทางจะแสดงไว้ในวินโดว์หมายเลข 23 ซึ่งจะแสดงซ้อนทับกันอยู่ให้เลือกรุ่นหมายเลข 37 เพื่อดูกระบวนการก่อนหน้านี้ หรือเลือกรุ่นหมายเลข 39 สำหรับกระบวนการถัดไป แต่ถ้าต้องการไล่ดูแบบอัตโนมัติให้เลือกรุ่นหมายเลข 38 ถ้าต้องการยกเลิกกระบวนการส่งถ่ายข้อมูลที่กำลังแสดงให้เลือกรุ่นหมายเลข 40 โปรแกรมจะแสดงกรอบข้อความ (Message box) สอบถามยืนยันการยกเลิกจากผู้ใช้ ดังแสดงในภาพประกอบ ข-5 ถ้าต้องการยกเลิกและให้สามารถกลับมาส่งถ่ายข้อมูลใหม่ได้อีกครั้งให้เลือกรุ่น "Yes" ถ้าเลือกรุ่น "No" เป็นการยกเลิกการส่งถ่ายอย่างสมบูรณ์ และเลือกรุ่น "Cancel" ถ้าไม่ต้องการยกเลิก



ภาพประกอบ ข-5 กรอบข้อความสอบถามยืนยันการยกเลิกการส่งถ่ายข้อมูล

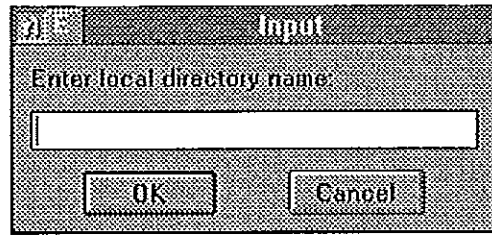
2.8. เมื่อต้องการใช้คำสั่งอื่น ๆ กับระบบแก้ไขข้อมูลของเครื่องคอมพิวเตอร์ที่ใช้งานอยู่ให้ดำเนินการดังนี้

2.8.1. เมื่อต้องการเปลี่ยนไปใช้งานไครเรททอรีอื่น ให้คลิก (Click) เมาส์ (Mouse) 2 ครั้งที่ชื่อไครเรททอรีจากหมายเลข 9 หรือเปลี่ยนการใช้งานโดยเลือกรุ่นหมายเลข 10 จะปรากฏวินโดว์ให้ใส่ชื่อไครเรททอรี ดังแสดงในภาพประกอบ ข-6



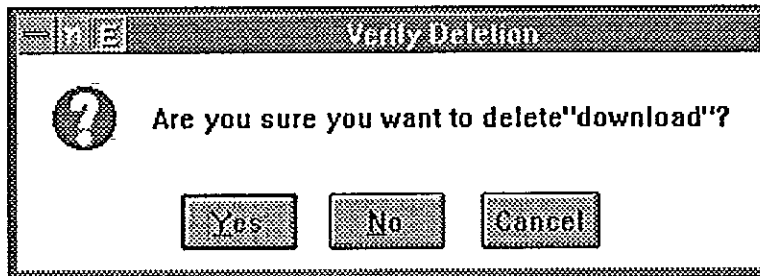
ภาพประกอบ ข-6 วินโดว์ให้ใส่ชื่อไครเรททอรีที่ต้องการย้ายไปทำงาน

2.8.2. เมื่อต้องการสร้างไครเรททอรีให้เลือกรุ่นหมายเลข 11 จะปรากฏวินโดว์ให้ใส่ชื่อไครเรททอรี ดังแสดงในภาพประกอบ ข-7



ภาพประกอบ ข-7 วินโดว์ให้ใส่ชื่อไดเรกทอรีที่ต้องการสร้างขึ้นใหม่

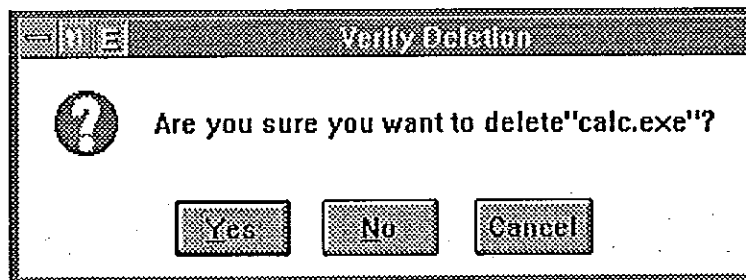
2.8.3. เมื่อต้องการลบ ไดเรกทอรีออกให้เลือกชื่อ ไดเรกทอรีที่ต้องการจากหมายเลข 9 และเลือกปุ่มหมายเลข 12 จะปรากฏกรอบข้อความยืนยันก่อนที่จะทำการลบจริง ดังแสดงในภาพประกอบ ข-8



ภาพประกอบ ข-8 กรอบข้อความยืนยันก่อนที่จะลบไดเรกทอรี

2.8.4. เมื่อต้องการทราบชื่อแฟ้มที่เก็บอยู่ในไดเรกทอรีปัจจุบันให้เลือกปุ่มหมายเลข 13

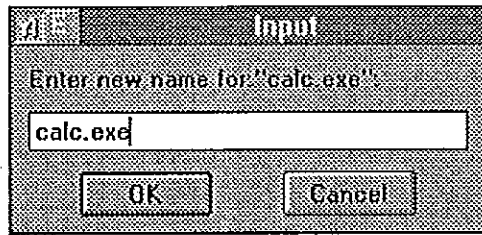
2.8.5. เมื่อต้องการลบแฟ้มข้อมูล ให้เลือกชื่อแฟ้มจากหมายเลข 14 และเลือกปุ่มหมายเลข 29 จะปรากฏกรอบข้อความยืนยันก่อนที่จะทำการลบจริง ดังแสดงในภาพประกอบ ข-9



ภาพประกอบ ข-9 กรอบข้อความยืนยันก่อนที่จะลบแฟ้มข้อมูล

2.8.6. เมื่อต้องการให้ปรับปรุงการแสดงชื่อ ไดเรกทอรีและชื่อแฟ้มข้อมูลใหม่อีกครั้งหนึ่ง ให้เลือกปุ่มหมายเลข 30

2.8.7. เมื่อต้องการเปลี่ยนชื่อแฟ้มข้อมูลให้เลือกชื่อแฟ้มที่ต้องการเปลี่ยนจากหมายเลข 14 และเลือกปุ่มหมายเลข 31 จะปรากฏ วินโดว์ให้ป้อนชื่อใหม่ดังแสดงในภาพประกอบ ข-10



### ภาพประกอบ ข-10 วินโดว์ให้ใส่ชื่อแฟ้มอันใหม่

2.9. เมื่อต้องการใช้คำสั่งที่เกี่ยวข้องกับระบบแฟ้มข้อมูลของสถานีบริการให้ใช้หลักเกณฑ์เช่นเดียวกับที่กล่าวไว้ในหัวข้อ 2.8

2.10. ในช่วงขณะที่มีการรอผลลัพท์การทำงานของคำสั่ง (FTP Reply) จากสถานีบริการ ถ้าต้องการยกเลิกคำสั่งให้เลือกปุ่มหมายเลข 27

2.11. ถ้าต้องการทราบชื่อไคลเอนท์ปัจจุบันของสถานีบริการที่ใช้งานอยู่ให้ดูที่หมายเลข 8 สำหรับชื่อไคลเอนท์ของสถานีบริการให้ดูที่หมายเลข 16

3. เมื่อต้องการส่งถ่ายข้อมูลด้วยโปรโตคอล HTTP ให้ทำดังนี้

3.1. เลือกปุ่มหมายเลข 3

3.2. เลือกช่องใส่ข้อความหมายเลข 5 และใส่ค่า URL ซึ่งมีรูปแบบดังนี้

host [ ":" port ][ abs\_path ]

host = ชื่อของสถานีบริการ

port = หมายเลขพอร์ตที่ต้องการติดต่อด้วยไม่จำเป็นต้องใส่ถ้าใช้พอร์ตหมายเลข 80

abs\_path = เส้นทางและชื่อแฟ้มข้อมูลที่ต้องการรับมา

3.3. เลือกปุ่มหมายเลข 24 เพื่อเริ่มการส่งถ่ายข้อมูล โปรแกรมจะแสดงวินโดว์ให้ใส่ชื่อแฟ้มข้อมูลที่ต้องการจัดเก็บไว้ ดังแสดงในภาพประกอบ ข-4

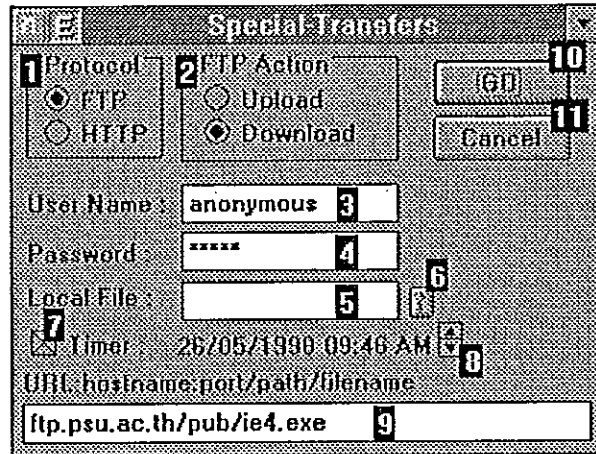
3.4. การแสดงการทำงานของกระบวนการส่งถ่ายข้อมูลเป็นเช่นเดียวกันกับหัวข้อ 1.7

3.5. เมื่อต้องการใช้คำสั่งอื่น ๆ กับระบบแฟ้มข้อมูลของสถานีงานที่ใช้งานอยู่ให้ดำเนินการเช่นเดียวกับหัวข้อ 1.8

3.6. การใช้งานระบบแฟ้มข้อมูลของสถานีบริการทางไม่สามารถทำได้

#### ● การส่งถ่ายข้อมูลแบบพิเศษ

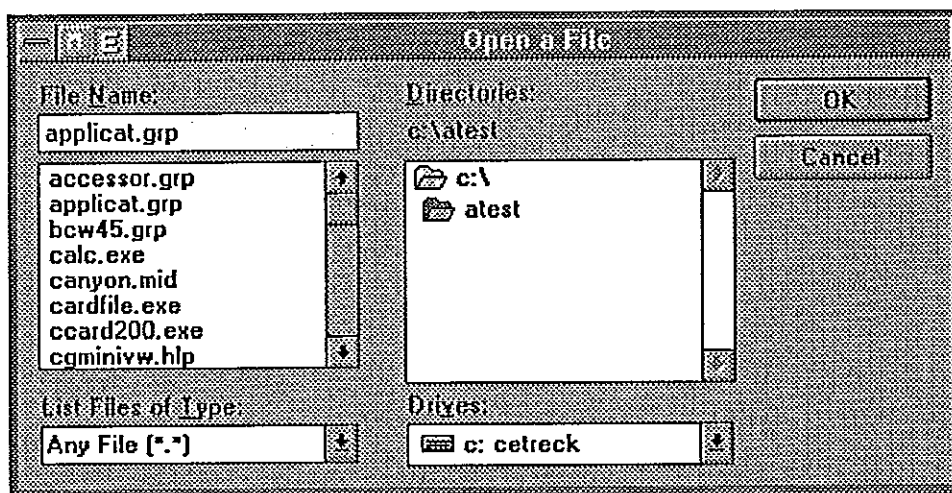
วิธีการส่งถ่ายข้อมูลแบบนี้จะเป็นการกำหนดข้อมูลที่จำเป็นต่อการส่งถ่ายข้อมูลอย่างละเอียดผ่านทางวินโดว์ ดังแสดงในภาพประกอบ ข-11



ภาพประกอบ ข-11 วินโดว์รับงานส่งถ่ายข้อมูล

จากภาพประกอบ ข-11 วิธีการเปิดวินโดว์ขึ้นมาใช้ทำได้โดยเรียกเมนูระบบจากวินโดว์หลัก จากนั้นเลือกรายการเมนู "Special Transfer..." หรืออาจกดปุ่ม Alt+t ก็ได้ ส่วนวิธีการกำหนดข้อมูลในวินโดว์มีดังนี้

1. ให้เลือกโปรโตคอลที่ต้องการใช้งานจากปุ่มหมายเลข 1
2. ถ้าใช้โปรโตคอล FTP ให้เลือกปุ่มหมายเลข 2 เพื่อกำหนดว่าจะทำการรับหรือส่งข้อมูล
3. กำหนดชื่อและรหัสผ่านของผู้ใช้ กรณีที่ใช้โปรโตคอล FTP ถ้าไม่มีการกำหนดชื่อและรหัสผ่านโปรแกรมจะกำหนดให้ชื่อผู้ใช้เท่ากับ "anonymous" และให้รหัสผ่านเท่ากับอีเมลแอดเดรสที่ได้กำหนดไว้ในวินโดว์ดังแสดงในภาพประกอบ ข-1 หมายเลข 4 แต่ถ้าใช้โปรโตคอล HTTP ชื่อผู้ใช้และรหัสผ่านจะมีการใช้งานถ้ามีการกำหนดขึ้นเท่านั้น
4. ถ้าใช้โปรโตคอล FTP ส่งแฟ้มข้อมูลจะต้องกำหนดชื่อแฟ้มในช่องใส่ข้อมูลหมายเลข 5 หรือให้เลือกปุ่มหมายเลข 6 จะปรากฏวินโดว์ชื่อแฟ้มให้เลือกแสดงดังแสดงในภาพประกอบ ข-12



ภาพประกอบ ข-12 วินโดว์ให้เลือกชื่อแฟ้มที่ต้องการส่ง

5. ถ้าต้องการตั้งเวลาให้เริ่มทำการส่งถ่ายข้อมูลให้เลือกปุ่มหมายเลข 7 และเลือกปุ่มหมายเลข 8 สำหรับตั้งเวลา

6. ให้ใส่ค่า URL ไว้ในช่องใส่ข้อมูลหมายเลข 9

7. เลือกปุ่มหมายเลข 10 เพื่อทำการส่งถ่ายข้อมูล หรือเลือกปุ่มหมายเลข 11 เพื่อยกเลิกการใช้งานวินโดว์

- การเฝ้าคุมการทำงานของงานส่งถ่ายข้อมูล

การเฝ้าคุมจะต้องใช้วินโดว์ "Process Monitor" ดังแสดงในภาพประกอบ ข-13

Local File	State	%Done	URL
ie4.exe	Submit	10%	ftp://ftp.psu.ac.th/pub/ie4.exe
ie4.exe	Ready	50%	ftp://ftp.microsoft.com/pub/i
index.htm	Running	80%	http://ftp.nectec.or.th

ภาพประกอบ ข-13 แสดงวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล

จากภาพประกอบ ข-13 วิธีการเปิดวินโดว์ขึ้นมาใช้ทำได้โดยเรียกเมนูระบบจากวินโดว์หลักจากนั้นเลือกรายการเมนู "Process Monitor..." หรืออาจกดปุ่ม Alt+m ก็ได้ ข้อมูลในแต่ละแถวคือ งานของการส่งถ่ายข้อมูลที่เกิดขึ้น ส่วนข้อมูลที่แสดงในแต่ละคอลัมน์สามารถอธิบายได้ดังนี้

- Local File แสดงชื่อแฟ้มที่ทำการรับหรือส่ง
- State แสดงสถานะของงานส่งถ่ายข้อมูลในแต่ละแถว ซึ่งมี 3 สถานะคือ สถานะแรก Submit งานที่อยู่ในสถานะนี้จะไม่มีการส่งถ่ายข้อมูล สถานะที่สอง Ready งานที่อยู่ในสถานะนี้พร้อมที่จะทำการส่งถ่ายข้อมูลแล้ว สถานะสุดท้ายคือ Running แสดงว่างานกำลังทำการส่งถ่ายข้อมูลอยู่
- %Done แสดงจำนวนเปอร์เซ็นต์ของข้อมูลที่ทำการส่งถ่ายข้อมูลเรียบร้อยแล้ว
- URL ค่า url ของงานในแต่ละแถว

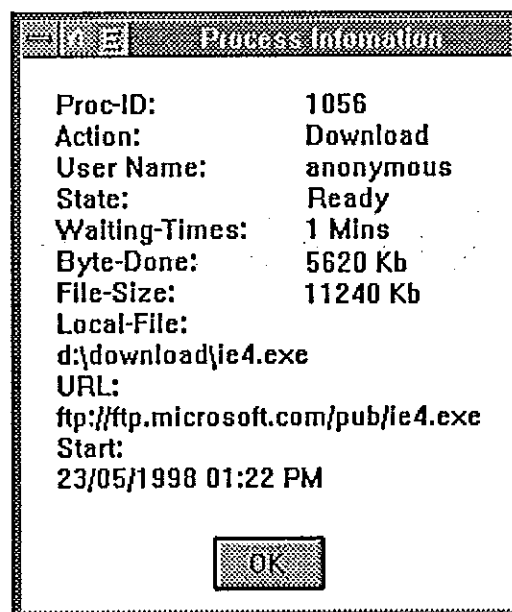
นอกเหนือจากการเฝ้าคุมงานส่งถ่ายข้อมูลแล้ว วินโดว์นี้ยังเพิ่มขีดความสามารถในการทำงานอื่น ๆ ซึ่งความสามารถดังกล่าวนี้สามารถใช้งานได้จากรายการเมนูของวินโดว์ และก่อนที่ จะทำการใช้งานจะต้องเลือกงานส่งถ่ายข้อมูลที่ต้องการก่อน การใช้งานเมนูอธิบายไว้ในตาราง ข-1 และ ข-2

ตาราง ข-1 แสดงการใช้งานรายการเมนู Edit ของวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล

รายการในเมนู Edit	การใช้งาน
Change State	ใช้ในการเปลี่ยนสถานะของงานส่งถ่ายข้อมูลจากที่เป็นอยู่ไปสู่สถานะใหม่คือ Submit หรือ Ready
Change Waiting-times	ใช้เปลี่ยนแปลงช่วงเวลาในการรอคอยของการส่งถ่ายข้อมูล
Change URL	ใช้เปลี่ยนแปลงค่าของ URL
Delete Select	ยกเลิกงานส่งถ่ายข้อมูลที่ได้เลือกไว้จากจากระบบ
Exit	ปิดการใช้งานวินโดว์

ตาราง ข-2 แสดงการใช้งานรายการเมนู View ของวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล

รายการในเมนู View	การใช้งาน
Refresh	ใช้เพื่อปรับปรุงการแสดงผลข้อมูลในวินโดว์ให้ทำการวาดใหม่อีกครั้งหนึ่ง
Process	ใช้เพื่อดูข้อมูลของงานการส่งถ่ายข้อมูลที่ได้เลือกไว้ ค.ย. ดังแสดงในภาพประกอบ ข-14



ภาพประกอบ ข-14 แสดงข้อมูลของงานส่งถ่ายข้อมูล

● การรายงานผลของการส่งถ่ายข้อมูล

โปรแกรมได้สร้างรายงานของการส่งถ่ายข้อมูลขึ้นมา 2 ประเภทคือ ประเภทแรกจะเป็นรายงานของการส่งถ่ายข้อมูลที่กำลังดำเนินอยู่ และที่กำลังแสดงการทำงานอยู่ในวินโดว์หมายเลข 23 จากภาพประกอบ ข-2 การดูรายงานประเภทนี้กระทำได้โดยดับเบิลคลิก (Double Click ) บริเวณใดก็ได้ในวินโดว์หมายเลข 23 โปรแกรมจะดำเนินการสร้างรายงานเก็บไว้ในแฟ้มข้อมูลชั่วคราว และจะเรียกใช้โปรแกรมบรรณาธิการซึ่งเป็นโปรแกรมที่ได้กำหนดเอาไว้ในวินโดว์ดังกล่าวแสดงในภาพประกอบ ข-1 ขึ้นมาแสดงแฟ้มข้อมูลดังกล่าว

รายงานประเภทที่สองจะเป็นรายงานของการส่งถ่ายข้อมูลที่เกิดขึ้นทั้งหมดและได้ดำเนินการเสร็จสิ้นแล้ว การดูรายงานประเภทนี้จะต้องเลือกปุ่มหมายเลข 25 จากภาพประกอบ ข-2 โปรแกรมจะดำเนินการสร้างรายงานเก็บไว้ในแฟ้ม http.log และจะเรียกใช้โปรแกรมบรรณาธิการขึ้นมาแสดงแฟ้มข้อมูลดังกล่าว รูปแบบของรายงานที่แสดงมีดังนี้

```

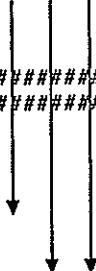
Connected 10.0.2.15
USER anonymous
220-Serv-U FTP-Server v2.1 for WinSock ready...
220 ftp.neoteo.or.th FTP server (Version wu-2.4(22) Thu Feb 13 18:02:26 GMT+0700 1997) ready.
CWD /c:/dos
230 User logged in - proceed
PWD
250 Directory changed to /c:/dos
PORT 10,0,2,15,4,172
257 "/c:/dos"TYPE A
200 PORT Command OK
LIST
Receiving directory
200 TYPE set to ASCII
150 Opening data connection
226 Transfer complete
#####
Completed Transfers:
ftp://10.0.2.15/c:/dos/ansi.sys -> d:\download\ansi.sys
Completed Transfers:
ftp://10.0.2.15/c:/dos/append.exe -> d:\download\append.exe
Completed Transfers:
ftp://10.0.2.15/c:/dos/attrib.exe -> d:\download\attrib.exe
##incompleted Transfers:
ftp://10.0.2.15/index.htm -> d:\download\index.htm
Because 550 File /c:/index.htm not found
#####
#### ftp://10.0.2.15/c:/dos/ansi.sys -> d:\download\ansi.sys
####
Connected 10.0.2.15
USER anonymous
220-Serv-U FTP-Server v2.1 for WinSock ready...
...
Completed Transfers:
ftp://10.0.2.15/c:/dos/ansi.sys -> d:\download\ansi.sys
#####
#####
#### ftp://10.0.2.15/c:/dos/append.exe -> d:\download\append.exe
####
Connected 10.0.2.15
USER anonymous
220-Serv-U FTP-Server v2.1 for WinSock ready...
...

```

ส่วนที่หนึ่ง

ส่วนที่สอง

ส่วนที่สาม





```

Completed Transfers:
ftp://10.0.2.15/c:/dos/append.exe -> d:\download\append.exe
#####
#####
#### ftp://10.0.2.15/c:/dos/attrib.exe -> d:\download\attrib.exe
####
Connected 10.0.2.15
USER anonymous
220-Serv-U FTP-Server v2.1 for WinSock ready. ...
...
Completed Transfers:
ftp://10.0.2.15/c:/dos/attrib.exe -> d:\download\attrib.exe
#####
#####
#### ftp://10.0.2.15/index.htm -> d:\download\index.htm
####
Connected 10.0.2.15
USER anonymous
220-Serv-U FTP-Server v2.1 for WinSock ready...
...
##Incompleted Transfers:
ftp://10.0.2.15/index.htm -> d:\download\index.htm
Because 550 File /c:/index.htm not found
550 File /c:/index.htm not found
#####

```

จากรูปแบบของรายงาน ในส่วนที่หนึ่งจะแสดงข้อความที่มีการรับส่งกับสถานีบริการ FTP ที่เชื่อมต่อผ่านทางวินโดวส์หลัก ในส่วนที่สองจะแสดงรายการสรุปของการส่งถ่ายข้อมูลเท่านั้น ส่วนรายงานของการส่งถ่ายข้อมูลอย่างละเอียดของแต่ละรายการสรุปจะแสดงไว้ในส่วนที่สาม

## ภาคผนวก ค

### การพัฒนาโปรแกรมด้วยวิธีการแบบเชิงวัตถุ

การพัฒนาโปรแกรมด้วยวิธีการนี้คือ แนวความคิดใหม่ในการเขียนโปรแกรมโดยมีจุดประสงค์หลักคือ เพื่อให้โปรแกรมประยุกต์ที่เขียนขึ้นจำลองหรือเลียนแบบการทำงานจริง ๆ ของมนุษย์ให้ใกล้เคียงกับสถานการณ์ที่เกิดขึ้นกับวัตถุต่าง ๆ บนโลกให้มากที่สุด แต่ถ้ามองในแง่มุมมองของผู้เขียนโปรแกรมแล้ว วิธีการนี้เป็นเครื่องมือที่จะสามารถช่วยผู้เขียนโปรแกรมจัดการกับความยุ่งยากซับซ้อนของโปรแกรมให้ลดลง

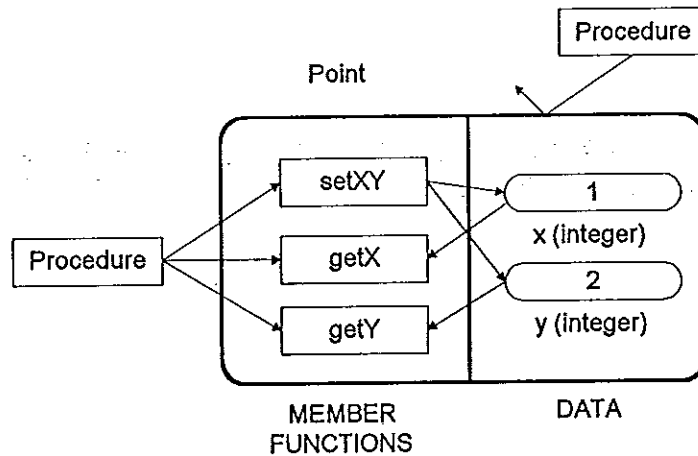
ในงานวิทยานิพนธ์นี้ได้เลือกใช้ภาษา C++ ในการเขียนโปรแกรม ภาษานี้ได้ออกแบบขึ้นมาเพื่อลดความยุ่งยากและซับซ้อนของภาษา C ลง การทำงานบางอย่างที่จะต้องใช้คำสั่งภาษา C ให้เรียกใช้ฟังก์ชันของระบบปฏิบัติการโดยตรงไปตรงมาเหมือนภาษาระดับต่ำ จะทำให้การเรียกใช้ง่ายขึ้น นอกจากนั้น C++ ยังมีความสามารถในการสร้างชนิดข้อมูลในรูปแบบที่ยืดหยุ่นกว่า ทำให้การปรับเปลี่ยนและการควบคุมทำได้ง่าย และสามารถปรับให้เข้ากับจุดประสงค์ของโปรแกรมประยุกต์

ลักษณะที่สำคัญของภาษาในแบบเชิงวัตถุมีด้วยกัน 3 ประการคือ Encapsulation, Inheritance และ Polymorphism

#### • Encapsulation

ถ้าพิจารณาจากศัพท์แล้วแปลได้ว่า การนำมารวมกันโดยมีเปลือกห่อหุ้ม ซึ่งความหมายที่แท้จริงของการพัฒนาโปรแกรมวิธีการแบบเชิงวัตถุคือ การรวมกันของโครงสร้างข้อมูลกับฟังก์ชันที่เกี่ยวข้องหรือเรียกใช้ข้อมูลนั้น (เรียกว่า Methods หรือ Action) เกิดเป็นวัตถุตัวใหม่หรือออบเจกต์ (Object) ที่สามารถทำการซ่อนข้อมูลจากภายนอกได้ โดย C++ จะเรียกวัตถุตัวนี้ว่า คลาส (Class)

เมื่อต้องการเข้าถึงข้อมูลในวัตถุ ก็จำเป็นต้องเรียกใช้ฟังก์ชันสมาชิก (Member Function) ของออบเจกต์ ซึ่งฟังก์ชันเหล่านี้จะเป็นผู้เข้าไปจัดการข้อมูลด้วยตัวเองแล้วส่งค่ากลับมาให้ตามที่ต้องการ ผู้เขียนโปรแกรมไม่สามารถเข้าไปจัดการข้อมูลได้ด้วยตนเองโดยตรงเหมือนแต่ก่อน นี่เป็นการป้องกันการเข้าถึงข้อมูลของฟังก์ชันอื่น ๆ ที่ไม่เกี่ยวข้อง และป้องกันการเปลี่ยนแปลงข้อมูลโดยอุบัติเหตุเรียกว่า การปิดบังข้อมูล (Data Hiding) ส่งผลให้ข้อมูลมีความปลอดภัยเพิ่มขึ้น ดังแสดงในภาพประกอบ ค-1

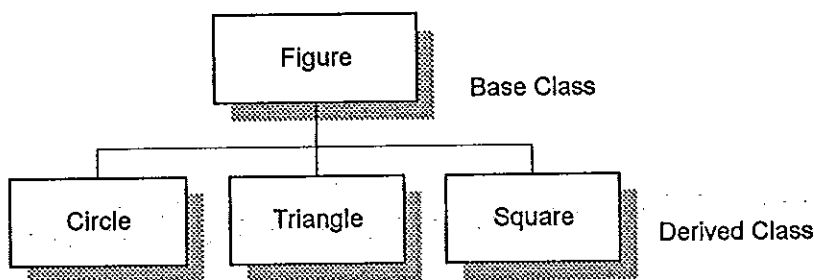


ภาพประกอบ ค-1 แสดงคุณสมบัติของ Encapsulation

จากภาพประกอบ ค-1 โครงสร้างของวัตถุที่ชื่อ Point แสดงให้เห็นถึงการปิดบังข้อมูล ส่วนที่เป็นข้อมูลไม่สามารถมองเห็นได้โดยจากโปรแกรม ยกเว้นจะมองผ่าน “ช่อง” หรือฟังก์ชันที่ จัดเตรียมไว้

● Inheritance

Inheritance คือการสืบทอดคุณสมบัติเป็นการสร้างคลาสใหม่ขึ้นมาโดยมีพื้นฐานมาจากคลาสเดิม แต่จะมี data หรือ member function ที่พิเศษเป็นของตัวเองเพิ่มขึ้นมาจากคลาสพื้นฐานเดิม ซึ่งก่อให้เกิดการแตกสายพันธุ์ของวัตถุเป็นวัตถุใหม่ที่มีคุณสมบัติของวัตถุเดิมซ่อนอยู่ไม่ มากก็น้อยนั่นเอง ซึ่งสามารถแสดงตัวอย่างการรับทอดคุณสมบัติดังแสดงในภาพประกอบ ค-2



```

class Figure{
.
.
.
};
class Circle:Figure{
.
.
.
};

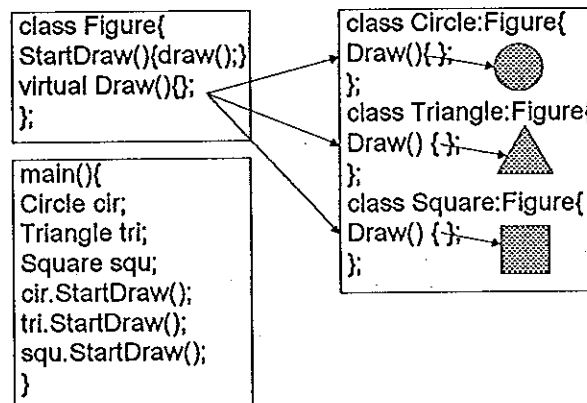
class Triangle:Figure{
.
.
.
};
class Square:Figure{
.
.
.
};
    
```

ภาพประกอบ ค-2 แสดงการสืบทอดคุณสมบัติจากคลาสพื้นฐานไปยังคลาสนุพันธ์

จากภาพประกอบ ก-2 รูปทรงเรขาคณิตได้แบ่งออกเป็นชั้น ๆ ตามลำดับชั้นของคลาส (Class Hierarchy) ที่ยอดบนสุดของลำดับชั้นจะเป็นคลาสพื้นฐาน (Base Class) ซึ่งในที่นี้คือ รูปทรงทั่วไป ในระดับชั้นรองลงมาจะเป็นคลาสอนุพันธ์ (Derived Class) ที่มีการสืบทอดสมบัติบางส่วนมาจากคลาสพื้นฐานเดิม แต่ก็มีคุณสมบัติเฉพาะที่มีในคลาสอนุพันธ์เพิ่มขึ้น

### ● Polymorphism

เป็นคำที่มาจากภาษากรีกแปลว่า “Having many shapes” หมายถึงการที่คลาสหนึ่ง ๆ สามารถแปรเปลี่ยนไปได้หลายรูปร่างขึ้นอยู่กับสภาพแวดล้อมหรือสถานการณ์ในขณะนั้น ซึ่งสามารถแสดงตัวอย่างดังแสดงในภาพประกอบ ก-3



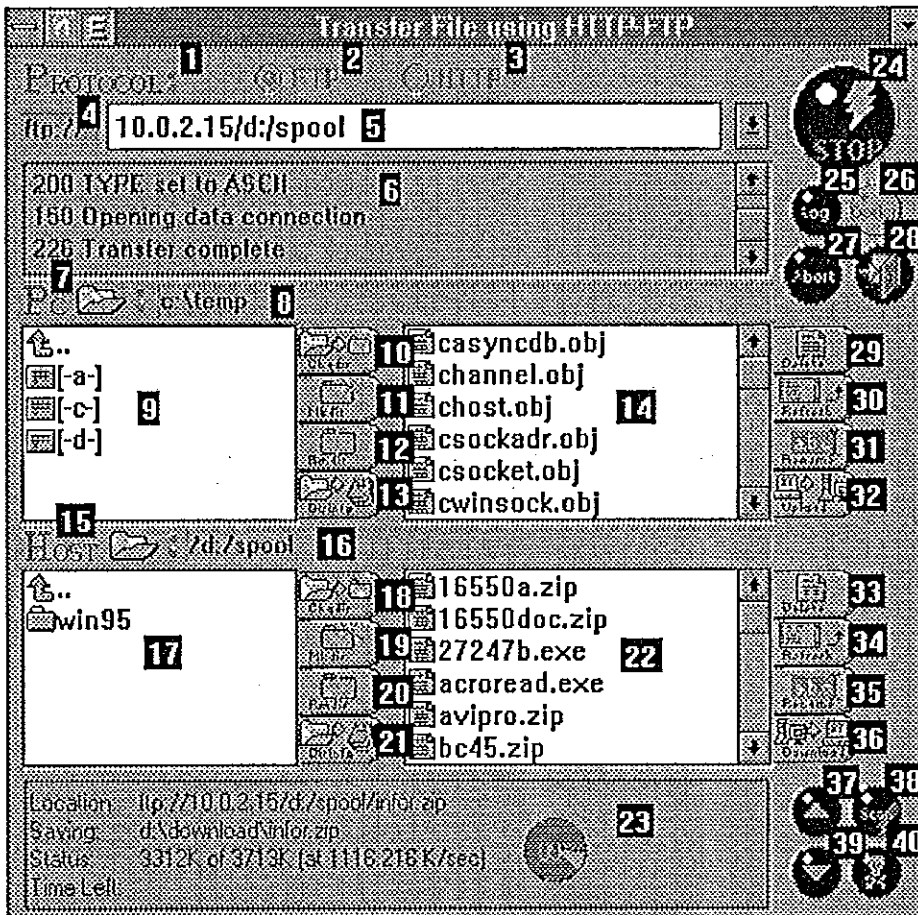
ภาพประกอบ ก-3 แสดงการทำงานของ Polymorphism

จากภาพประกอบ ก-3 จะเห็นว่ามียุทธศาสตร์เรขาคณิต 3 แบบที่แตกต่างกัน แต่ทุกอันก็ยังคงอยู่ในคลาสพื้นฐานเหมือนกันหมด ซึ่งสำหรับ C++ แล้วจะมอง Polymorphism ในส่วนที่เรียกว่าฟังก์ชันเสมือน หรือ Virtual Function จะเป็นการยอมให้ผู้เขียน โปรแกรมใช้ฟังก์ชันชื่อเดียวกัน แต่ทำงานไม่เหมือนกันจำแนกไปตามชนิดของข้อมูลที่ฟังก์ชันได้รับขณะทำงาน เมื่อฟังก์ชันได้รับข้อมูลที่ต้องการแล้วก็จะไปเลือกฟังก์ชันที่เหมาะสมกับข้อมูลนั้น ๆ ทำงานต่อไป

ภาคผนวก ง

ภาพประกอบของส่วนติดต่อกับผู้ใช้

- วินโดว์หลัก



ภาพประกอบ ง-1 วินโดว์หลัก

ตาราง ง-1 แสดงองค์ประกอบของวินโดว์หลัก

ลำดับที่	ชื่อองค์ประกอบ	หน้าที่
1	Bitmap	แสดงคำว่าโปรโตคอลให้ผู้ใช้เลือกองค์ประกอบที่ 2 และ 3
2	Radio button	เลือกเพื่อส่งถ่ายข้อมูลโดยโปรโตคอล FTP
3	Radio button	เลือกเพื่อส่งถ่ายข้อมูลโดยโปรโตคอล HTTP
4	Static text	แสดงชนิดของ URL ที่จะใส่ไว้ในองค์ประกอบที่ 5
5	Combo box	ใส่ค่า URL
6	Edit text	แสดงสถานะการทำงานของการทำงานส่งถ่ายข้อมูลของโปรแกรม
(7)	Bitmap	แสดงให้เห็นว่าองค์ประกอบที่ 8 เป็นชื่อไคลเอนต์ปัจจุบัน
(8)	Static text	แสดงชื่อไคลเอนต์ปัจจุบันที่กำลังใช้งานอยู่
(9)	List box	แสดงรายชื่อไคลเอนต์
(10)	Push button	เลือกเพื่อย้ายการทำงานไปยังไคลเอนต์อื่น
(11)	Push button	เลือกเพื่อสร้างไคลเอนต์ใหม่
(12)	Push button	เลือกเพื่อลบไคลเอนต์หรือออก
(13)	Push button	เลือกเพื่อแสดงรายละเอียดทั้งหมดของไคลเอนต์ที่ใช้งานอยู่
(14)	List box	แสดงรายชื่อเพิ่มในไคลเอนต์ปัจจุบัน
[15]	Bitmap	แสดงให้เห็นว่าองค์ประกอบที่ 16 เป็นชื่อไคลเอนต์ปัจจุบัน
[16]	Static text	แสดงชื่อไคลเอนต์ปัจจุบันที่กำลังใช้งานอยู่
[17]	List box	แสดงรายชื่อไคลเอนต์
[18]	Push button	เลือกเพื่อย้ายการทำงานไปยังไคลเอนต์อื่น
[19]	Push button	เลือกเพื่อสร้างไคลเอนต์ใหม่
[20]	Push button	เลือกเพื่อลบไคลเอนต์หรือออก
[21]	Push button	เลือกเพื่อแสดงรายละเอียดทั้งหมดของไคลเอนต์ที่ใช้งานอยู่
[22]	List box	แสดงรายชื่อเพิ่มในไคลเอนต์ปัจจุบัน
23	child window	แสดงสารสนเทศของงานส่งถ่ายข้อมูลงานหนึ่งจากหลาย ๆ งานที่กำลังดำเนินอยู่
24	Push button	เลือกเพื่อเริ่มทำการส่งถ่ายข้อมูล
25	Push button	เลือกเพื่อแสดงรายงานสรุปของการส่งถ่ายข้อมูล
26	Push button	เลือกเพื่อกำหนดสถานะการทำงานของโปรแกรม
27	Push button	เลือกเพื่อยุติการใช้งานคำสั่งของโปรโตคอล FTP
28	Push button	เลือกเพื่อยุติการทำงานของโปรแกรม
(29)	Push button	เลือกเพื่อลบเพิ่มข้อมูลออกจากไคลเอนต์ปัจจุบัน

ตาราง ง-1 (ต่อ) แสดงส่วนประกอบของวินโดว์หลัก

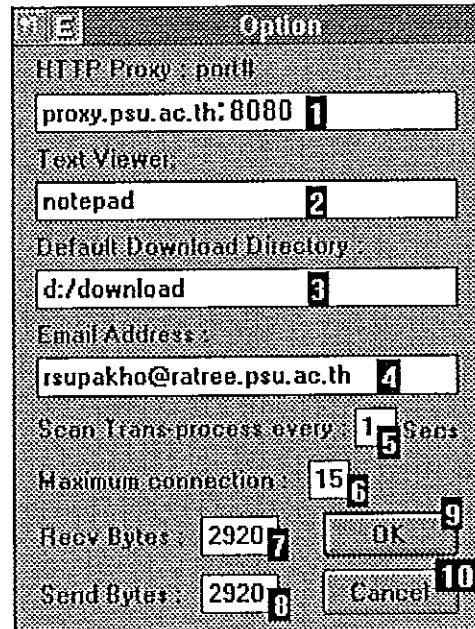
ลำดับที่	ชื่อองค์ประกอบ	หน้าที่
(30)	Push button	เลือกเพื่อแสดงรายชื่อไคลเอนต์และชื่อเพิ่มในองค์ประกอบที่ 9 และ 14 ใหม่
(31)	Push button	เลือกเพื่อเปลี่ยนชื่อเพิ่มใหม่
(32)	Push button	เลือกเพื่อส่งเพิ่มที่ได้เลือกไว้จากองค์ประกอบที่ 14 ไปยังสถานีบริการ FTP ที่กำลังเชื่อมต่ออยู่
[33]	Push button	เลือกเพื่อลบเพิ่มข้อมูลออกจากไคลเอนต์ปัจจุบัน
[34]	Push button	เลือกเพื่อแสดงรายชื่อไคลเอนต์และชื่อเพิ่มในองค์ประกอบที่ 17 และ 22 ใหม่
[35]	Push button	เลือกเพื่อเปลี่ยนชื่อเพิ่มใหม่
[36]	Push button	เลือกเพื่อรับเพิ่มที่ได้เลือกไว้จากองค์ประกอบที่ 22 จากสถานีบริการ FTP ที่กำลังเชื่อมต่ออยู่
37	Push button	เลือกเพื่อแสดงงานส่งถ่ายข้อมูลที่อยู่ก่อนงานที่กำลังแสดงอยู่ในองค์ประกอบที่ 23
38	Push button	เลือกเพื่อแสดงงานส่งถ่ายข้อมูลที่อยู่ถัดจากงานที่กำลังแสดงอยู่ในองค์ประกอบที่ 23 อย่างอัตโนมัติ
39	Push button	เลือกเพื่อแสดงงานส่งถ่ายข้อมูลที่อยู่ถัดจากงานที่กำลังแสดงอยู่ในองค์ประกอบที่ 23
40	Push button	เลือกเพื่อยกเลิกงานส่งถ่ายข้อมูลที่กำลังแสดงอยู่ในองค์ประกอบที่ 23

หมายเหตุ:

( ) ระบบเพิ่มของสถานีลูกข่าย

[ ] ระบบเพิ่มของสถานีบริการ FTP

- วินโดว์กำหนดสภาวะแวดล้อมของโปรแกรม



ภาพประกอบ ง-2 วินโดว์กำหนดสภาวะแวดล้อมของโปรแกรม

ตาราง ง-2 แสดงองค์ประกอบของวินโดว์กำหนดสภาวะแวดล้อมของโปรแกรม

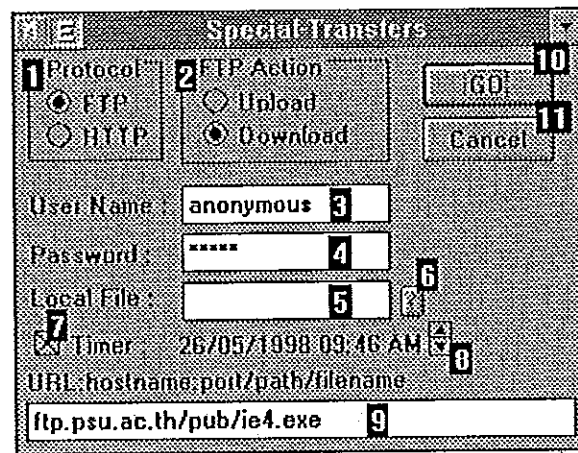
ลำดับที่	ชื่อองค์ประกอบ	หน้าที่
1	Edit text	ใส่ชื่อของสถานีบริการหรือชื่อ สำหรับการส่งถ่ายข้อมูลโดย โพรโตคอล HTTP
2	Edit text	ใส่ชื่อโปรแกรมบรรณาธิการ (Edit Program) สำหรับแสดงรายละเอียดของไดเรกทอรี จากองค์ประกอบที่ 13 และ 21 ของ วินโดว์หลัก
3	Edit text	ใส่ชื่อไดเรกทอรีที่จัดเก็บแฟ้มข้อมูลที่ได้รับมาจากสถานีบริการ
4	Edit text	ใส่เลขที่อยู่อีเมลล์สำหรับเป็นรหัสผ่านสำหรับการส่งถ่ายข้อมูลโดย โพรโตคอล FTP
5	Edit text	ใส่จำนวนความถี่เป็นวินาทีในการตรวจสอบงานส่งถ่ายข้อมูล หลายงานที่กำลังเชื่อมต่อกับสถานีบริการ
6	Edit text	ใส่จำนวนสูงสุดของงานส่งถ่ายข้อมูลที่สามารถเชื่อมต่อกับสถานี บริการได้
7	Edit text	ใส่จำนวนไบต์ของ Buffer ที่ใช้เก็บข้อมูลที่รับมาจากสถานี บริการ



ตาราง ง-2 (ต่อ) แสดงองค์ประกอบของวินโดว์กำหนดสถานะแวดล้อมของโปรแกรม

ลำดับที่	ชื่อองค์ประกอบ	หน้าที่
8	Edit text	ใส่จำนวนไบต์ของ Buffer ที่เก็บข้อมูลที่จะส่งไปยังสถานีบริการ
9	Push button	เลือกเพื่อจัดเก็บค่าที่ใส่ไว้ในองค์ประกอบที่ 1-8
10	Push button	ยกเลิกการใช้งานวินโดว์

• วินโดว์รับงานส่งถ่ายข้อมูล



ภาพประกอบ ง-3 วินโดว์รับงานส่งถ่ายข้อมูล

ตาราง ง-3 แสดงองค์ประกอบของวินโดว์รับงานส่งถ่ายข้อมูล

ลำดับที่	ชื่อองค์ประกอบ	หน้าที่
1	Radio button	เลือกชนิดของโปรโตคอลที่จะทำการส่งถ่ายข้อมูล
2	Radio button	เลือกทิศทางของการส่งถ่ายข้อมูลที่ใช้โปรโตคอล FTP
3	Edit text	ใส่ชื่อผู้ใช้
4	Edit text	ใส่รหัสผ่านของผู้ใช้
5	Edit text	ใส่ชื่อแฟ้มที่จะส่งไปให้แก่สถานีบริการ FTP
6	Push button	เช่นเดียวกับองค์ประกอบที่ 5 แต่ชื่อแฟ้มที่ใส่ได้มาจากวินโดว์รายชื่อแฟ้มที่ระบบปฏิบัติการเตรียมไว้ให้
7	check box	เลือกให้มีการตั้งเวลาให้เริ่มงานส่งถ่ายข้อมูล
8	Time	ปรับเปลี่ยนเวลา
9	Edit text	ใส่ค่า URL
10	Push button	เลือกเพื่อสร้างงานส่งถ่ายข้อมูลขึ้นมา
11	Push button	ยกเลิกการใช้งานวินโดว์

• วินโดว์เฝ้าคุมการส่งถ่ายข้อมูล

Local File	State	%Done	URL
	Submit	10%	ftp://ftp.psu.ac.th/pub/ie4.exe
ie4.exe	Ready	50%	ftp://ftp.microsoft.com/pub/i
index.htm	Running	80%	http://ftp.nectec.or.th

ภาพประกอบ ง-4 แสดงวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล

- รายละเอียดของสารสนเทศที่แสดง
  - Local File ชื่อแฟ้มบนสถานีลูกข่ายที่กำลังดำเนินการส่งถ่ายข้อมูลอยู่
  - State สถานะของงานส่งถ่ายข้อมูล
  - %Done จำนวนเปอร์เซ็นต์ของข้อมูลที่ส่งถ่ายเสร็จแล้ว
  - URL

ตาราง ง-4 แสดงรายการเมนู Edit ของวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล

รายการในเมนู Edit	การใช้งาน
Change State	ใช้ในการเปลี่ยนสภาวะของงานส่งถ่ายข้อมูลจากที่เป็นอยู่ไปสู่สภาวะใหม่
Change Waiting-times	ใช้เปลี่ยนแปลงช่วงเวลาในการรอคอยของการส่งถ่ายข้อมูล
Change URL	ใช้เปลี่ยนแปลงค่าของ URL
Delete Select	ยกเลิกงานส่งถ่ายข้อมูลที่ได้เลือกไว้จากระบบ
Exit	ปิดการใช้งานวินโดว์

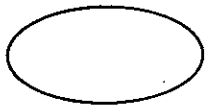
ตาราง ง-5 แสดงรายการเมนู View ของวินโดว์เฝ้าคุมงานส่งถ่ายข้อมูล

รายการในเมนู View	การใช้งาน
Refresh	ใช้เพื่อปรับปรุงการแสดงผลข้อมูลในกรอบวินโดว์ให้ทำการวาดใหม่อีกครั้งหนึ่ง
Process	ใช้เพื่อดูข้อมูลของงานการส่งถ่ายข้อมูลที่ได้เลือกไว้

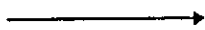
## ภาคผนวก จ

### วิธีกลไกสถานะ

วิธีกลไกสถานะ (State Machine) เป็นแผนภาพที่แสดงการเปลี่ยนสถานะ (State) ที่ดำเนินอยู่ในระบบ ซึ่งจะมีจำนวนของสถานะที่แน่นอน ซึ่งเป็นวิธีหนึ่งที่ใช้ในการออกแบบโปรโตคอล หรืออาจนำไปประยุกต์ใช้กับระบบงานอื่น ๆ ก็ได้ สัญลักษณ์แสดงความหมายที่ปรากฏในแผนภาพกลไกสถานะมีดังนี้



แทนสถานะ ซึ่งจะมีชื่อของสถานะปรากฏภายในกรอบ

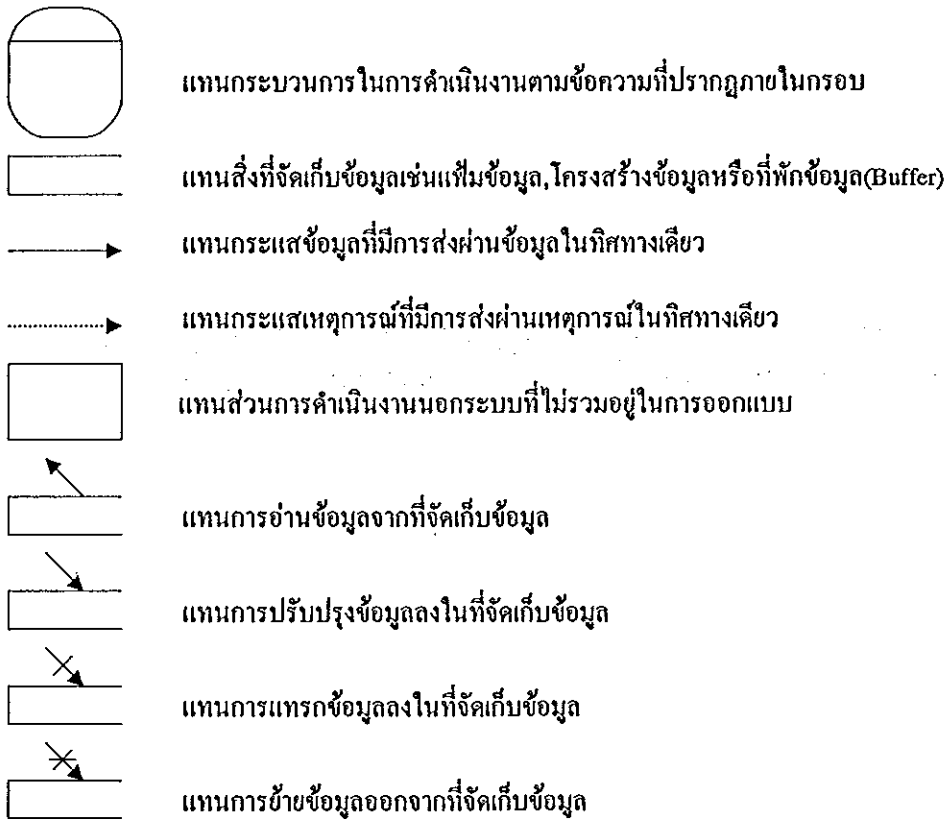


แทนการเปลี่ยนสถานะจากสถานะเดิม ไปยังอีกสถานะถัดไปตามทิศทางของหัวลูกศร ซึ่งจะมีเหตุการณ์ที่ทำให้ต้องเปลี่ยนสถานะกำกับไว้

ภาคผนวก ก

แผนภาพกระแสข้อมูล

แผนภาพกระแสข้อมูล (DFD:Data Flow Diagram) เป็นแผนภาพที่ใช้แสดงการส่งถ่ายข้อมูลที่เกิดขึ้นระหว่างกระบวนการในระบบ ทำให้ทราบถึงขั้นตอนของกระบวนการต่าง ๆ ว่ามีการส่งถ่ายข้อมูลอย่างไร ซึ่งเป็นการสะดวกในการทำความเข้าใจการดำเนินงานของระบบหนึ่ง ๆ ก่อนที่จะลงรายละเอียดต่าง ๆ สัญลักษณ์ที่แสดงความหมายที่ปรากฏในแผนภาพกระแสข้อมูลมีดังนี้



การอธิบายสิ่งที่จัดเก็บไว้ในที่จัดเก็บข้อมูลมีรูปแบบที่แสดงโดยใช้ตัวอย่างได้ดังนี้  
ชื่อของที่จัดเก็บข้อมูล = \*สิ่งที่เขียนภายในเครื่องหมายดอกจันคือหมายเหตุ\*

หน่วยข้อมูลที่ 1 + หน่วยข้อมูลที่ 2 + หน่วยข้อมูลที่ 3

ค.ย. การจัดเก็บข้อมูลซ้ำ

PASSENGER LIST = {passenger name}

ค.ย. การจัดเก็บข้อมูลที่เลือกได้

CUSTOMER ORDER = [vacuum cleaner order | jet engine order]

## ประวัติผู้เขียน

ชื่อ นาย สุภกร รัตมีมณฑล

วัน เดือน ปีเกิด วันเสาร์ที่ 27 เดือน เมษายน พ.ศ. 2511

วุฒิการศึกษา

วุฒิ

ชื่อสถาบัน

ปีที่สำเร็จการศึกษา

วิทยาศาสตร์บัณฑิต

มหาวิทยาลัยรามคำแหง

2536

(วิทยาการคอมพิวเตอร์)

ทุนการศึกษา (ที่ได้รับระหว่างการศึกษา)

งานวิจัยนี้ได้รับทุนสนับสนุนจากโครงการทุนบัณฑิตศึกษาในประเทศ สำนักงาน  
พัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ