

Java Network Programming

การเขียนโปรแกรมโครงข่ายด้วยภาษาจาวา

เนื้อหาประกอบด้วย

- ❖ แนวคิดการเขียนโปรแกรมผ่านระบบเครือข่าย
- ❖ จาวาซ็อกเก็ต
- ❖ จาวาเทรตและจาวาคอนเคอร์เรนซ์เซิร์ฟเวอร์
- ❖ การใช้งานมิดเดิลแวร์ด้วยจาวาวอร์ดา
- ❖ จาวาอาร์เอ็มไอ
- ❖ การเขียนโปรแกรมติดต่อฐานข้อมูลด้วยเจดีบีซี
- ❖ จาวาเซิร์ฟเล็ต
- ❖ จาวาเซิร์ฟเวอร์เพจ
- ❖ จาวาเซิร์ฟเวอร์เพจ
- ❖ จาวาเว็บเซอร์วิส

คำนำ

ภายใต้แรงขับเคลื่อนและความกดดันของสถานการณ์ของโลกไร้พรมแดน ทำให้การพัฒนาโปรแกรมคอมพิวเตอร์ยุคใหม่จำเป็นต้องสร้างโปรแกรมที่มีขีดความสามารถในการจัดการข้อมูลแบบได้แบบไร้พรมแดน ฉะนั้นการพัฒนาโปรแกรมแบบเดิมที่รันบนเครื่องคอมพิวเตอร์เพียงเครื่องเดียวนั้น ไม่เพียงพอต่อความต้องการของผู้ใช้ในยุคปัจจุบัน ผู้เขียนมีโอกาสพัฒนาโปรแกรมให้กับองค์กร ซึ่งเป็นโปรแกรมที่ควบคุมระบบต่าง ๆ ผ่านระบบเครือข่าย ซึ่งโปรแกรมดังกล่าวเป็นเพียงหนึ่งรูปแบบของการพัฒนาโปรแกรมผ่านระบบเครือข่ายเท่านั้น

ตำราเล่มนี้แนะนำการพัฒนาโปรแกรมผ่านระบบเครือข่าย โดยเริ่มตั้งแต่การพัฒนาด้วยการจัดการข้อมูลผ่านพอร์ตโดยตรง ซึ่งผู้เขียนโปรแกรมต้องออกแบบข้อมูลการรับและส่ง อีกทั้งควบคุมจังหวะการส่งข้อมูลกันเอง จนกระทั่งถึงโมเดลของการพัฒนาโปรแกรมแบบกระจายที่อาศัยมิดเดิลแวร์คอร์บายและจาวาอาร์เอ็มไอช่วยในการจัดการข้อมูล ครึ่งหลังของหนังสือเล่มนี้ยังได้นำเสนอโมเดลของการพัฒนาโปรแกรมด้วยรูปแบบเว็บแอปพลิเคชันในรูปแบบต่างๆ อีกด้วย

ผู้เขียนหวังเป็นอย่างยิ่งว่าตำราเล่มนี้จะเป็นคู่มือที่ดีในการพัฒนาโปรแกรมผ่านระบบเครือข่าย ทั้งโปรแกรมเมอร์ ครู อาจารย์ นิสิต นักศึกษา เพื่อที่จะได้นำเอาความรู้ที่มีไปต่อยอดสำหรับการบูรณาการให้เป็นผู้มีความเชี่ยวชาญและใช้งานได้จริง

เชาวลิต ชันคำ

25 ธ.ค. 54

สารบัญ

หน้า

คำนำ.....	(1)
สารบัญ	(3)
สารบัญภาพ	(7)
สารบัญตาราง	(19)
บทที่ 1 แนวคิดการเขียนโปรแกรมผ่านระบบเครือข่าย.....	1
เครือข่ายคอมพิวเตอร์	1
โมเดลสถาปัตยกรรมเครือข่าย	8
ไคลเอนต์/เซิร์ฟเวอร์	12
สรุป	18
แบบฝึกหัดท้ายบท	19
บทที่ 2 ซ็อกเก็ต	21
ความหมายและความสำคัญ	21
การจัดการอินเทอร์เน็ตแอดเดรส	24
การจัดการข้อมูลผ่านซูอาร์แอล	29
การเขียน โปรแกรมผ่านซ็อกเก็ตด้วยพีซีพี	33
สรุป	50
แบบฝึกหัดท้ายบท	51
บทที่ 3 เทตและคอนเคอร์เรนซ์เซิร์ฟเวอร์.....	53
โพรเซส	53
เทต	55
การใช้งานเทตด้วยภาษาจาวา	57
คอนเคอร์เรนซ์เซิร์ฟเวอร์ด้วยมัลติเทต	63
สรุป	79
แบบฝึกหัดท้ายบท	80
บทที่ 4 การใช้งานมิดเดิลแวร์ด้วยจาวาคอร์บา	81
มิดเดิลแวร์	81
ออบเจกต์โอเรียลเตดมิดเดิลแวร์	85

	หน้า
คอร์ดบา	90
การเขียนโปรแกรมผ่านระบบเครือข่ายด้วยจาวาคอร์ดบา.....	95
สรุป	117
แบบฝึกหัดท้ายบท	118
บทที่ 5 จาวอาร์เอ็มไอ	119
ความหมายและหลักการของจาวอาร์เอ็มไอ	119
หลักการการทำงานของอาร์เอ็มไอ.....	122
การพัฒนาแอปพลิเคชันด้วยจาวอาร์เอ็มไอ.....	125
สรุป	138
แบบฝึกหัดท้ายบท	139
บทที่ 6 การเขียนโปรแกรมติดต่อฐานข้อมูลด้วยเจดบีซี	141
การเขียนติดต่อฐานข้อมูลด้วยภาษาจาวา.....	141
ระบบจัดการฐานข้อมูล.....	144
มายเอสคิวแอลแอคเคมินิสเตรเตอร์	149
การดาวน์โหลดเจดบีซีไดรเวอร์และการติดตั้งไดรเวอร์	155
การเขียนโปรแกรมจัดการแฟ้มข้อมูล.....	159
สรุป	164
แบบฝึกหัดท้ายบท	165
บทที่ 7 จาวาเชิร์ฟเล็ต.....	167
สถาปัตยกรรมและรูปแบบทั่วไปของเชิร์ฟเล็ต	167
กลาสฟิชเว็บเชิร์ฟเวอร์	172
โปรแกรมเน็ตบิน	177
การใช้งานเชิร์ฟเล็ตด้วยโปรแกรมเน็ตบิน	181
การใช้งานฐานข้อมูลด้วยจาวาเชิร์ฟเล็ต	194
สรุป	201
แบบฝึกหัดท้ายบท	202
บทที่ 8 จาวาเชิร์ฟเวอร์เพจ.....	203
หลักการของเว็บแอปพลิเคชัน	203
รูปแบบและการใช้งานเจเอสพี	206

	หน้า
การจัดการฐานข้อมูลด้วยภาษาเจเอสพี	227
การใช้งานฟอร์มร่วมกับเจเอสพี	217
สรุป	237
แบบฝึกหัดท้ายบท	238
บทที่ 9 จาวาเซิร์ฟเวอร์เฟส	239
พื้นฐานจาวาเซิร์ฟเวอร์เฟส	239
รูปแบบของจาวาเซิร์ฟเวอร์เฟส	243
การเขียนจาวาเซิร์ฟเวอร์เฟสด้วยเน็ตบิน	249
สรุป	292
แบบฝึกหัดท้ายบท	293
บทที่ 10 จาวาเว็บเซอร์วิส	295
ความหมายและความสำคัญของเว็บเซอร์วิส	295
องค์ประกอบของเว็บเซอร์วิส	296
การพัฒนาเว็บเซอร์วิส	301
การพัฒนาจาวาเว็บเซอร์วิสด้วยโปรแกรมเน็ตบิน	303
สรุป	343
แบบฝึกหัดท้ายบท	344
บรรณานุกรม	345

บทที่ 1

แนวคิดการเขียนโปรแกรมผ่านระบบเครือข่าย

วัตถุประสงค์ของการพัฒนาโปรแกรมผ่านระบบเครือข่ายนั้นเพื่อให้ใช้งานทรัพยากรของคอมพิวเตอร์ร่วมกันได้ไม่ว่าจะเป็นอุปกรณ์ต่าง ๆ เช่น ซีดีรอม เครื่องพิมพ์ ฮาร์ดดิสก์ หรือ อื่น ๆ ทั้งโปรแกรมหรืออ็อบเจกต์ของโปรแกรมร่วมกันกล่าวคือหากมีผู้ใดในเครือข่ายพัฒนาโปรแกรมจะเป็นฟังก์ชัน (Function) โปรซีเจอร์ (Procedure) หรืออ็อบเจกต์ (Object) ก็ตาม หากได้พัฒนาขึ้นมาแล้วให้ผู้อื่น สามารถเข้าถึงและใช้งานส่วนที่เป็นประโยชน์ได้ก็ถือว่าเป็นแนวคิดที่จะเอื้อต่อการใช้งานร่วมกัน และด้วยวัตถุประสงค์เดียวกันแต่ต่างองค์กรต่างหน่วยงานกันทำให้เกิดมาตรฐานต่าง ๆ รวมถึงมีโมเดลการส่งผ่านข้อมูลโดยอาศัยระบบเครือข่ายที่แตกต่างกันออกไปมากมาย ซึ่งในหนังสือเล่มนี้จะยกมากล่าวพอเข้าใจเป็นหลักการเบื้องต้นเพียง TCP Model และ OSI Model ที่ถือว่าเป็นหลักสำคัญเท่านั้น

เนื้อหาของบทนี้จะกล่าวถึงแนวคิดการสร้างเครือข่ายเบื้องต้น การออกแบบระบบต่าง ๆ การสร้างของการเขียนโปรแกรมบนระบบเครือข่าย และมาตรฐานเบื้องต้นที่จำเป็นต้องมีความรู้สำหรับการใช้งานในบทต่อไป

เครือข่ายคอมพิวเตอร์

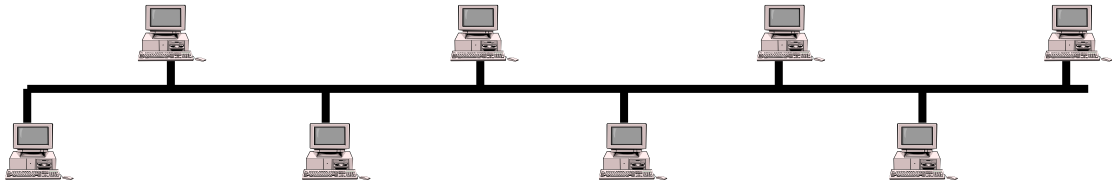
เครือข่ายคอมพิวเตอร์ (Computer Networking) คือการนำเอาคอมพิวเตอร์ตั้งแต่สองเครื่องขึ้นไปมาเชื่อมต่อกันเพื่อแบ่งปันทรัพยากร ทั้งอุปกรณ์ โปรแกรมและข้อมูลต่าง ๆ ที่ในระบบคอมพิวเตอร์ เพื่อให้เกิดการใช้งานได้อย่างมีประสิทธิภาพสูงสุด ทั้งในแง่ของการใช้ทรัพยากรและการติดต่อสื่อสารกัน

แนวคิดของการแบ่งปันกันนี้จะมีทั้งระดับภายในพื้นที่ใกล้ ๆ กัน และแตกต่างกันไปทั่วโลก นอกจากนั้นแล้วการสื่อสารกันในระดับที่เป็นเครือข่ายที่ใหญ่ขึ้นตามลำดับ ทำให้เกิดแนวคิด และสิ่งที่ต้องคำนึงถึงในการพัฒนาส่วนของการติดต่อกันทั้งทางด้านกายภาพ และด้านการพัฒนาโปรแกรม และมาตรฐานการส่งข้อมูลต่าง ๆ บนเครือข่ายเพื่อให้ปลอดภัยและเข้าตรงไปหาผู้ที่ต้องการใช้ให้มากที่สุด จึงเกิดรูปแบบของการเชื่อมต่อขึ้นมาในรูปแบบที่ต่างกันไปเรียกว่าโทโปโลยี (Topology)

โทโปโลยี

โทโปโลยี คือ การออกแบบการเชื่อมต่อคอมพิวเตอร์ในระบบ ซึ่งแนวคิดดังกล่าวนี้จะพิจารณาทางด้านกายภาพ ซึ่งจะต้องมีอุปกรณ์ของการส่งข้อมูล ทั้งที่เป็นสาย และไร้สายต่าง โดยรูปแบบของการเชื่อมต่อแบ่งได้ดังนี้

1) บัส (Bus)

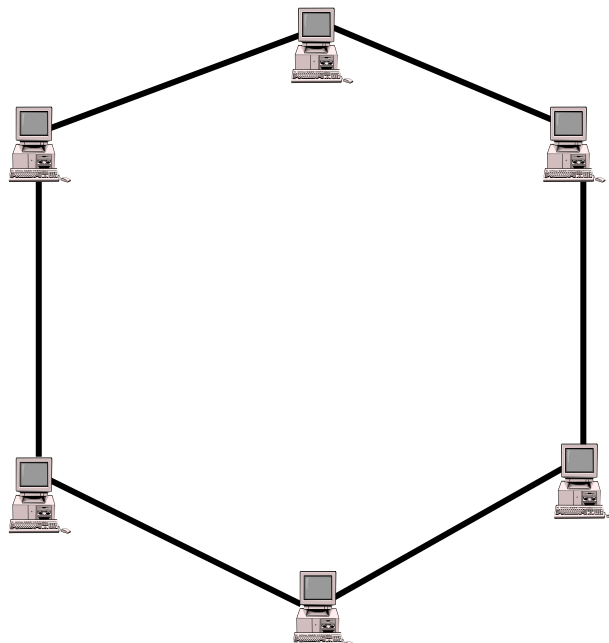


ภาพที่ 1.1 แสดงการสถาปัตยกรรมแบบบัส

(ที่มา : Computer Networks, 2009)

การเชื่อมต่อแบบนี้จะใช้สื่อในการส่งข้อมูล เช่นสายแบคโบน (Backbone) เดินไว้เป็นเส้นทางเดียว แล้วเครื่องที่จะเข้าเชื่อมต่อจะนำสายหรืออุปกรณ์สำหรับจับสัญญาณเข้ามาต่อดังรูป ซึ่งแบบนี้มีข้อดีคือสามารถขยายขนาดของเครือข่ายได้โดยไม่ต้องใช้ต้นทุนมาก แต่ข้อเสียคือ หากสายส่งสัญญาณชำรุดใช้การไม่ได้จะทำให้ระบบเครือข่ายทั้งระบบล่มได้

2) แบบวงแหวน (Ring)

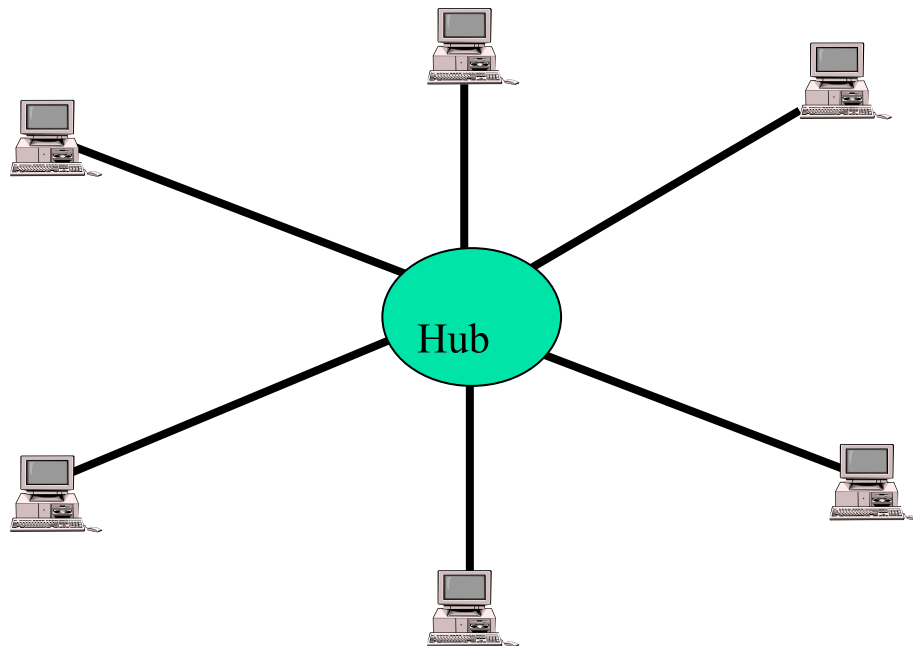


ภาพที่ 1.2 แสดงการสถาปัตยกรรมแบบวงแหวน

(ที่มา : Computer Networks, 2009)

เครือข่ายที่เชื่อมต่อแบบนี้จะมีลักษณะคล้ายกับแบบบัส โดยนำหัวท้ายของสายสัญญาณมาต่อกันเข้าเป็นวงกลมช่วยแก้ปัญหาด้านการส่งข้อมูลหากสายขาดไป

3) แบบดาว (Star)

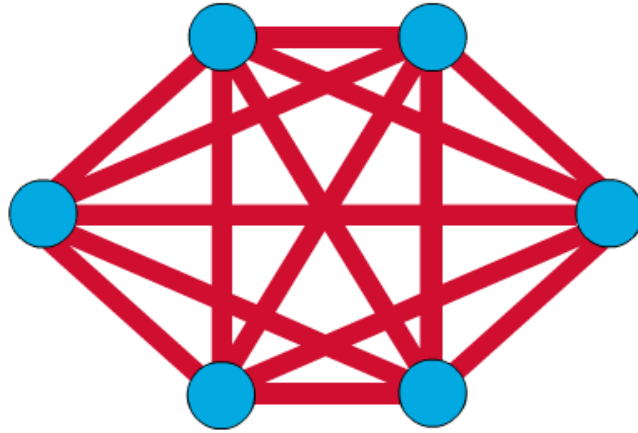


ภาพที่ 1.3 แสดงการสถาปัตยกรรมแบบดาว

(ที่มา : Computer Networks, 2009)

การเชื่อมต่อแบบนี้ จะมีตัวกลางเป็นส่วนกระจายสัญญาณและทำหน้าที่เป็นตัวเชื่อมต่อหลัก ซึ่งสามารถจัดปัญหาของแบบวงแหวนและแบบบัสที่มีปัญหาที่อาจเกิดกับสายสัญญาณทำให้ระบบมีปัญหาได้ แต่อย่างไรก็ตามแบบนี้เองเป็นแบบที่นิยมใช้งานอยู่หากตัวเชื่อมต่อเกิดปัญหาขึ้นระบบก็จะเกิดปัญหาในการส่งข้อมูลได้เช่นกัน

4) การเชื่อมต่อแบบเมช (Mesh)



ภาพที่ 1.4 แสดงการสถาปัตยกรรมแบบเมช

(ที่มา : Computer Network topology, 2009)

การเชื่อมต่อแบบนี้ จะนำคอมพิวเตอร์เพื่อเชื่อมต่อกันทั้งหมดทุกเครื่องซึ่งการเชื่อมต่อโดยวิธีนี้สามารถรับประกันได้ว่าระบบเครือข่ายไม่มีโอกาสที่จะเสียได้ แต่ก็ต้องแลกด้วยการลงทุนที่สูงกว่าแบบอื่น ๆ

อุปกรณ์สำหรับการเชื่อมต่อ

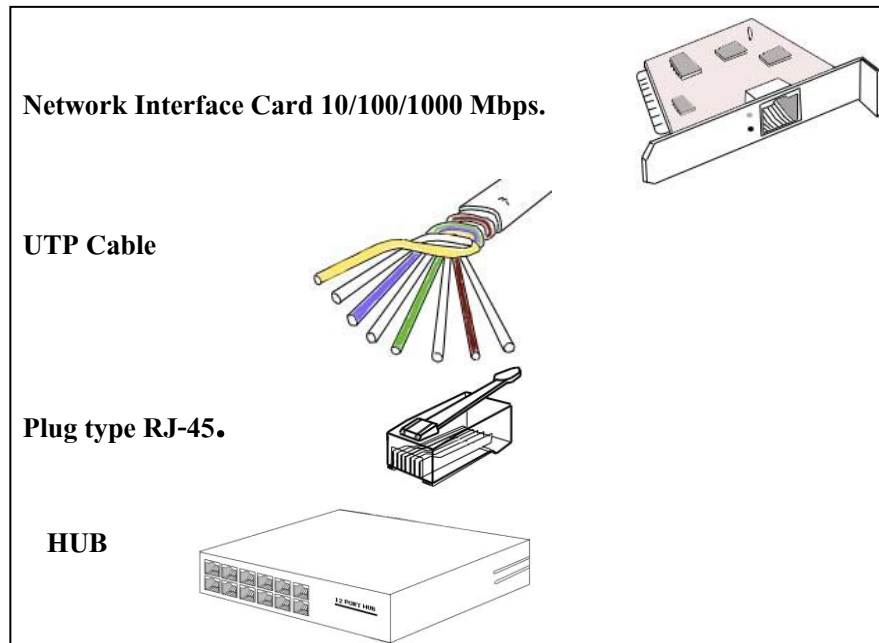
สิ่งที่ทำให้การเชื่อมต่อแบบต่าง ๆ เกิดขึ้นได้คืออุปกรณ์เครือข่าย (Network device) ที่มีอยู่หลากหลายชนิด ซึ่ง WeeSan Lee (2009) ได้แบ่งประเภทของอุปกรณ์เชื่อมต่อ ดังนี้

- LAN
- WLAN
- Dial-up Services
- Broadband Services
- WAN

โดยที่การเชื่อมต่อต้องใช้ส่วนการของการ์ด LAN ซึ่งแบ่งออกเป็นประเภทหลัก ๆ ดังนี้

- 10BaseT เป็นอุปกรณ์แบบเดิมที่มีความเร็วในการส่งข้อมูลต่ำ ใช้งาน 10Mbps
- 10Mbps (Mega bits per second)
- 100BaseT
- 100Mbps
- 1000BaseT
- 1000Mbps or 1Gbps

นอกจากนั้นแล้วยังใช้สายที่มีอยู่ทั่วไป 2 ประเภทคือสายโคแอกเซียล (Coaxial cable) และสายแบบบิดเกลียวคู่ (Twist pair) เป็นต้น
ตัวอย่างของสายประเภทต่าง ๆ แสดงดังภาพ



ภาพที่ 1.5 แสดงตัวอย่างของอุปกรณ์เครือข่ายแบบใช้สาย

นอกจากนั้นแล้วยังมีอุปกรณ์ต่าง ๆ ที่ใช้กับระบบไร้สาย ในปัจจุบันมีหลากหลายประเภทที่ทำให้การเชื่อมต่อและการใช้งานเครือข่ายทำได้สะดวกมากขึ้น ตัวอย่างเช่น Wi-Fi (Wireless Fidelity), Access Point (AP), ESSID (Extended Service Set ID) หรือ Hotspot เป็นต้น อุปกรณ์ที่ทำงานกับส่วนการเชื่อมต่อต้องมีมาตรฐานเช่น 802.11b - 11Mbps, 802.11g - 54Mbps, 802.11a - 54Mbps เป็นต้น

สถาปัตยกรรมของเครือข่าย

สถาปัตยกรรมของเครือข่ายจะพิจารณาตามความสำคัญของเครื่องที่เกิดจากการเชื่อมต่อกันเป็นเครือข่าย และขึ้นอยู่กับการให้บริการทรัพยากรต่าง ๆ รวมถึงส่วนของโปรแกรมที่มีการเรียกใช้ผ่านเครือข่าย โดยที่สถาปัตยกรรมในแนวคิดนี้จะเกี่ยวข้องกับการพัฒนาโปรแกรมของหนังสือเล่มนี้ โดยทั่วไปอาจแบ่งได้ดังนี้

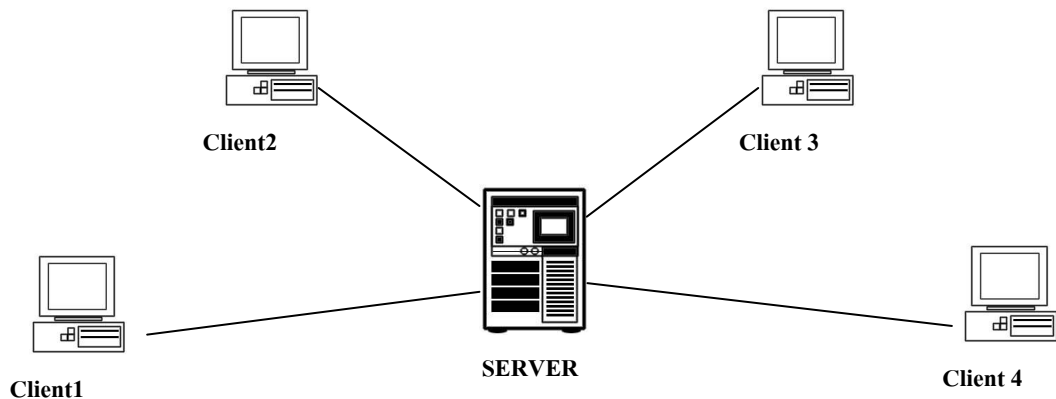
1. แบบเพียร์-ทู-เพียร์ (Peer-to-Peer) ซึ่งทุกเครื่องในกลุ่มมีความสำคัญเท่ากัน สามารถทำให้เครื่องตนเองเป็นได้ทั้งผู้ให้บริการและผู้รับบริการ แสดงดังภาพ



ภาพที่ 1.6 แสดงการสถาปัตยกรรมแบบเพียร์-ทู-เพียร์

ระบบการเชื่อมต่อแบบนี้เครื่องทุกเครื่องสามารถให้บริการทั้งเพิ่ม อุปกรณ์ฟ่วงต่าง ๆ โดยไม่มีเครื่องใดเป็นแม่ข่ายควบคุม แต่ต้องมีระบบปฏิบัติการที่สนับสนุนการเชื่อมต่อแบบนี้ด้วยจึงจะสามารถแบ่งปันทรัพยากรต่าง ๆ ระหว่างกันได้ ปัจจุบันนี้ระบบปฏิบัติการทุกชนิดสามารถทำให้เกิดการทำงานแบบนี้ได้

2. แบบไคลเอ็นต์/เซิร์ฟเวอร์ (Client/Server) แบบนี้จะมีเครื่องที่สำคัญที่สุดในกลุ่มทำหน้าที่เป็นผู้ให้บริการ และเครื่องที่เหลือเป็นลูกข่ายทำหน้าที่เป็นผู้รับบริการ แสดงดังรูป



ภาพที่ 1.7 แสดงการสถาปัตยกรรมไคลเอนต์/เซิร์ฟเวอร์

การเชื่อมต่อแบบนี้จะมีหลากหลายรูปแบบโมเดล ซึ่งจะอธิบายในตอนท้ายของบทนี้ และนอกจากนั้นยังใช้แนวคิดนี้สำหรับการเขียนโปรแกรมในบทต่อ ๆ ไป อีกด้วย

โปรโตคอล

โปรโตคอล หมายถึง กฎเกณฑ์ที่เป็นมาตรฐานในการติดต่อสื่อสารระหว่างกันของคอมพิวเตอร์ ซึ่งกำหนดขึ้นมาเป็นมาตรฐานสำหรับส่งข้อมูลจากต้นทางไปยังปลายทางได้อย่างถูกต้อง โดยผู้รับและผู้ส่งจะต้องเรียนรู้ร่วมกันและกำหนดให้ตรงกันเสมอ

ตัวอย่างโปรโตคอลเช่น TCP/IP ซึ่งพัฒนาโดย ARPANET จนได้รับความนิยมและมีการใช้อย่างกว้างขวางจนเป็นมาตรฐานการส่งข้อมูลในเครือข่ายอินเทอร์เน็ตในปัจจุบัน นอกจากนั้นยังมีหน่วยงานที่เกี่ยวข้องกับการสร้างมาตรฐานดังกล่าวนี้ อีก เช่น

ISO (International Standards Organization)

IEEE (American Institution of Electrical and Electronics Engineers)

ITU-T (International Telecommunication Union- Telecommunications)

หมายเลขของเครื่องหรืออุปกรณ์เครือข่าย

การกำหนดเป้าหมายของการติดต่อกันของเครื่องคอมพิวเตอร์ในระบบเครือข่ายจำเป็นต้องกำหนดหมายเลขประจำแต่ละตำแหน่งซึ่งแบ่งออกเป็นหมายเลขของฮาร์ดแวร์ (Max Address) และหมายเลขไอพี

ไอพี หมายถึงหมายเลขที่กำหนดขึ้นมาเพื่ออ้างอิงสำหรับเครื่องคอมพิวเตอร์หรืออุปกรณ์ของคอมพิวเตอร์ในระบบเครือข่าย ซึ่งจะไม่ซ้ำกัน และหมายเลขดังกล่าวนี้สามารถที่จะเปลี่ยนแปลงได้ ปัจจุบันนี้มีการใช้งานอยู่สองเวอร์ชันคือ IPV4 และ IPv6 เป็นต้น

หมายเลขฮาร์ดแวร์ได้แก่ แมคแอดเดรส ซึ่งจะเป็หมายเลขที่ผูกติดกับฮาร์ดแวร์นั้น ๆ ทั้งอุปกรณ์เครือข่าย และฮาร์ดแวร์ที่ใช้สำหรับการติดต่อ

การทำงานของโปรโตคอล IP

การทำงานของโปรโตคอลนี้ต้องมีการกำหนด IP Address (IPV4) มีขนาด 32 bits แบ่งเป็นกลุ่มคลาสดังนี้

ตารางที่ 1.1 แสดงการแบ่งไอพีคลาส

Class	Address Range	Supports
Class A	1.0.0.1 to 126.255.255.254	สนับสนุน 16 ล้านโฮส บน 127 เครือข่าย
Class B	128.1.0.1 to 191.255.255.254	สนับสนุน 65,000 โฮส บน 16,000 เครือข่าย
Class C	192.0.1.1 to 223.255.254.254	สนับสนุน 254 โฮส บน 2 ล้านเครือข่าย
Class D	224.0.0.0 to 239.255.255.255	สงวนสำหรับกลุ่มของมัลติเคส
Class E	240.0.0.0 to 254.255.255.254	สงวนไว้สำหรับอนาคตหรือการทำวิจัยและพัฒนา

(ที่มา: IP Class, 2009)

สำหรับไอพีที่หมายเลข 127.x.x.x จะเป็นไอพีสงวนไว้สำหรับใช้เป็นลูบแบบสำหรับทดสอบการทำงานของอุปกรณ์เครือข่าย เช่น ของเครื่องตนเองจะมีหมายเลข 127.0.0.1 เป็นต้นหรือ

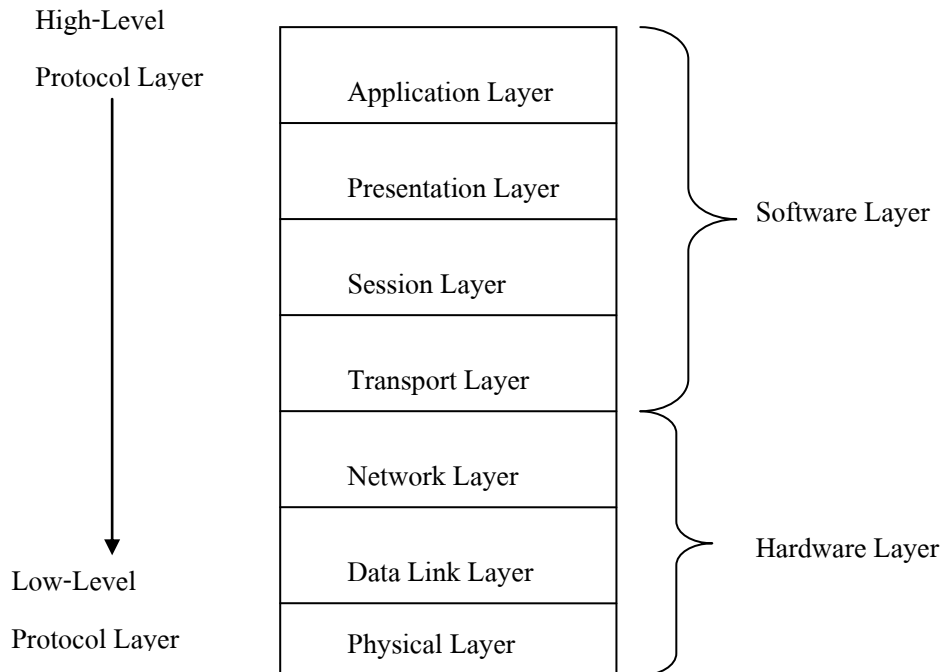
หมายเลข 255.255.255.255 สงวนเอาไว้เป็นส่วนของบรอดแคสต์(broadcast) ไปยังทุก ๆ โฮสบนเครือข่ายเดียวกันเป็นต้น

แต่เนื่องจากในปัจจุบันไอพีเวอร์ชัน 4 ที่ใช้งานกันอยู่ไม่สามารถรองรับการใช้งานที่จะเกิดขึ้นในอนาคตได้ อีกทั้งจำนวนของอุปกรณ์และผู้ใช้เครื่องคอมพิวเตอร์มีจำนวนมากขึ้นดังนี้จึงเกิดการเต็มของไอพีเวอร์ชันนี้ขึ้น ดังนั้นไอพีจึงถูกพัฒนาเป็นเวอร์ชัน 6 (IPv6) โดยการขยายขนาดของแอดเดรสออกเป็น 128 บิต เพื่อให้สามารถกำหนดแอดเดรสได้มากกว่าเดิม 296 เท่า สนับสนุนการส่งข้อมูลแบบเรียลไทม์ (Realtime) และรองรับอุปกรณ์อิเล็กทรอนิกส์ในอนาคตอาจจำเป็นต้องมีไอพีทุก ๆ ชิ้น

โมเดลสถาปัตยกรรมเครือข่าย

โมเดลสถาปัตยกรรมเครือข่าย (Network Architecture Model) ที่จะกล่าวถึงต่อไปนี้จะอธิบายถึงลำดับชั้นของฮาร์ดแวร์และซอฟต์แวร์ที่จำเป็นต่อการส่งผ่านข้อมูลไปยังจุดสองจุด ซึ่งโมเดลที่ใช้กันมากและเป็นมาตรฐานของการส่งผ่านข้อมูลคือ OSI Model และ Internet Model จะแบ่งลำดับชั้นของ Model ออกเป็น 7 ชั้นหลักแสดงดังภาพ 1.1 และมีรายละเอียด ดังต่อไปนี้คือ

- Application Layer เป็นชั้นที่เป็นของโปรแกรมประยุกต์ทั่วไปที่ผู้ใช้ได้ติดต่อและใช้งานหรือการมองก็มองโดยผ่านโปรแกรมประยุกต์ต่าง ๆ ที่เราใช้งาน เช่น โปรแกรม ล็อกอินระยะไกล (Remote Login) ,FTP tools,e-mail webpage ต่าง ๆ เป็นต้น
- Presentation Layer เป็นชั้นที่แปลงข้อมูลระหว่างชั้น บนสุดคือ Application ให้อยู่ในรูปแบบที่เหมาะสมที่จะส่งและรับข้อมูล
- Session Layer เป็นลำดับชั้นที่ทำการสร้างส่วนของการติดต่อระหว่างผู้ใช้
- Transport Layer เป็นลำดับชั้นที่จัดเตรียม end-to-end สำหรับการติดต่อและให้แน่ใจว่าข้อมูลไปถึงจุดหมายปลายทางอย่างแท้จริงเมื่อออกจากต้นฉบับ (Source) ไปแล้วถ้าหากมองอีกอย่างหนึ่งก็คือการ Synchronize นั่นเอง



ภาพที่ 1.8 แสดงลำดับชั้นของ OSI Model

- Network Layer เป็นลำดับชั้นของการสร้างและตอบรับ โดยจัดการและสร้างการติดต่อโดยถ่ายข้อมูลแบบโหนดต่อโหนดภายใต้เครือข่าย
- Data Link Layer ลำดับนี้ตอบรับสำหรับการรับข้อมูลและโอนย้ายข้อมูลลงไปยัง frame ด้วย Header ที่เหมาะสมและจัดการเรื่องของ address รวมถึงการตรวจหาข้อผิดพลาดด้วย
- Physical Layer ลำดับชั้นนี้จะจัดการในเรื่องของฮาร์ดแวร์และสัญญาณไฟฟ้าที่จะใช้ในการส่งผ่านข้อมูล โดยจัดการในเรื่องแรงดันไฟฟ้า จังหวะของการรับและส่งในลักษณะของสัญญาณไฟฟ้า เป็นต้น

Internet Model หรือ TCP/IP Model

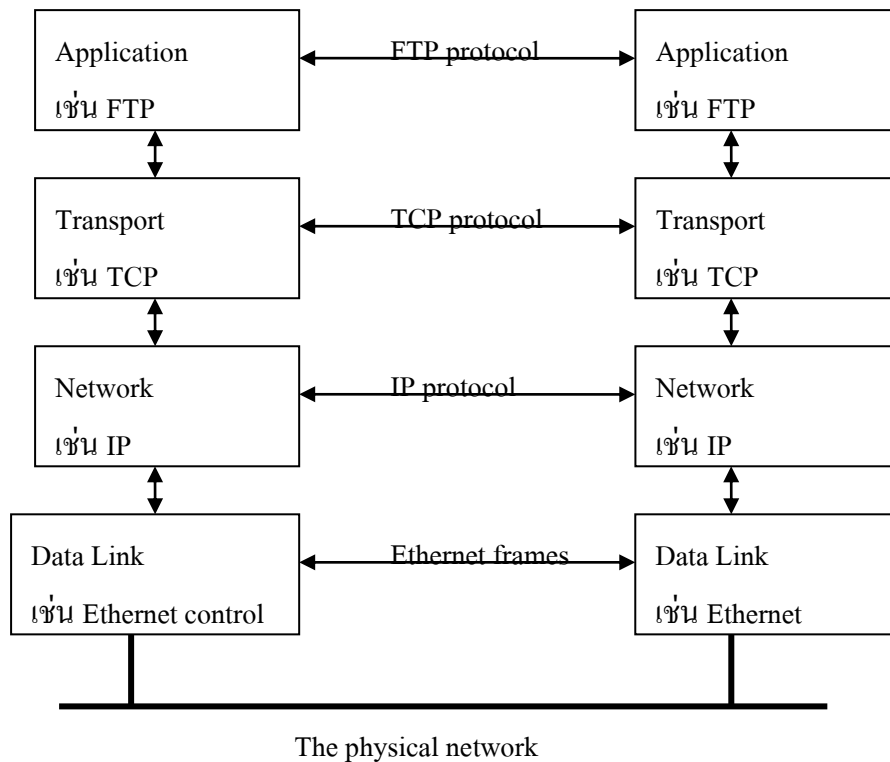
โมเดลนี้ใช้งานกันอย่างแพร่หลายในปัจจุบันเพราะเป็นส่วนหลักที่ใช้ในการติดต่อกันผ่านเครือข่ายของอินเทอร์เน็ต ลำดับชั้นของอินเทอร์เน็ตโมเดลจะแบ่งออกเป็น 5 ลำดับชั้นของการทำงาน ซึ่งจะน้อยกว่า OSI Model แต่จะครอบคลุมชั้นของ OSI เทียบเท่ากันถึงแม้จะเหลื่อมกันในบางลำดับชั้นก็ตาม แสดงลำดับชั้นเปรียบเทียบกับ OSI Model ได้ ดังภาพที่ 1.9

OSI Model	TPC/IP Model
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	Network Layer
Physical Layer	Physical Layer

ภาพที่ 1.9 แสดงการเปรียบเทียบระหว่าง OSI และ TCP/IP Model

การติดต่อระหว่าง 2 ระบบ

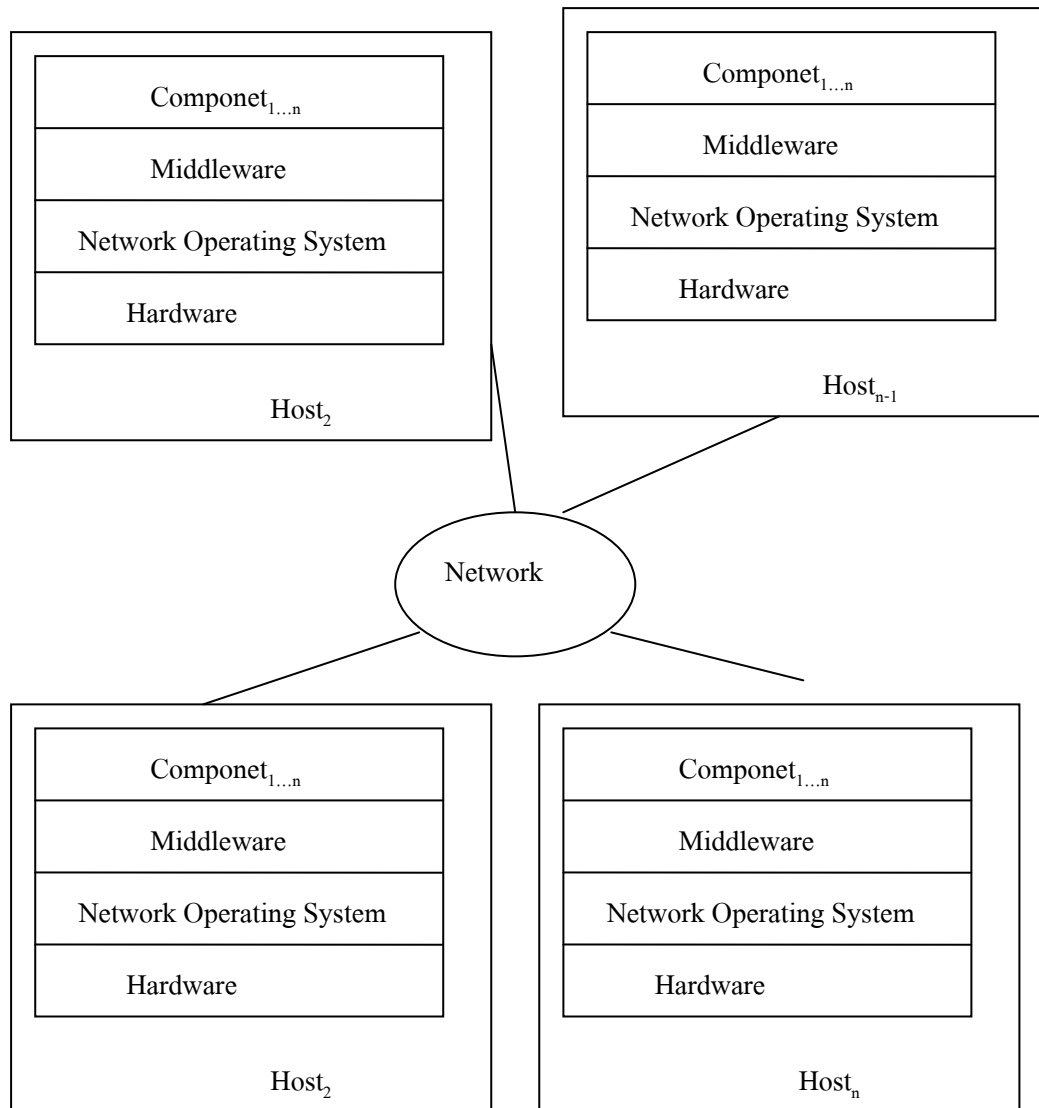
การติดต่อกันระหว่าง 2 ระบบนั้นในรายละเอียดส่วนนี้จะอธิบายให้เห็นในรูปแบบของการจัดการระหว่าง TCP/IP กับ TCP/IP ในรายละเอียดของการติดต่อในรูปแบบของคอมพิวเตอร์ นั้นผู้ส่งข้อมูลจะต้องทำการแปลงข้อมูลและจัดการตามลำดับชั้นที่สูงสุดลงมาหาต่ำสุดแล้วจึงส่งข้อมูลผ่านลำดับชั้น Physical Layer ไปตามสื่อของระบบเครือข่าย ผู้รับจะทำตรงกันข้ามคือรับข้อมูลจากลำดับชั้นต่ำสุดแล้วแปลงขึ้นไปยังชั้นสูงสุด ในขณะที่ผู้ใช้เองจะมีความรู้สึได้ว่าได้ติดต่อกับลำดับชั้นที่เท่ากันเท่านั้นเช่นหากใช้งาน โปรแกรมเว็บเบราว์เซอร์อยู่ก็จะรู้สึว่าติดต่อกับระดับของเว็บเบราว์เซอร์ ภาพแสดงดังต่อไปนี้



ภาพที่ 1.10 แสดงตัวอย่างการสร้างการติดต่อผ่านโอเอสไอโมเดล

ระบบกระจาย (Distributed system)

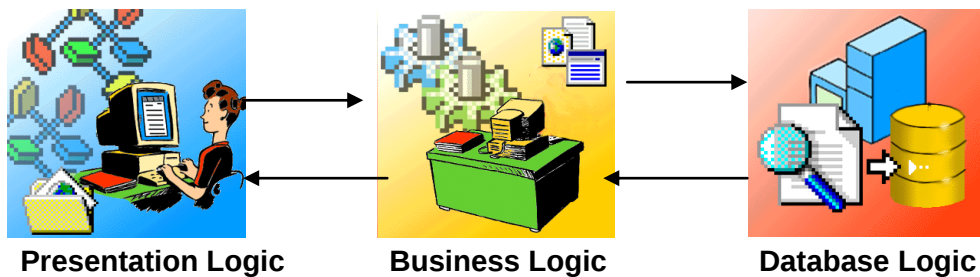
ในระบบนี้จะมีการติดต่อผ่านกันเครือข่าย โดยที่การติดต่อนั้นอาจต่างระบบปฏิบัติการกัน มีฮาร์ดแวร์ที่แตกต่างกันแต่สามารถใช้งานคอมพิวเตอร์ โดยที่ผู้เรียกใช้งานไม่จำเป็นต้องรู้ว่าคอมพิวเตอร์เหล่านี้อยู่ที่เครื่องใด ลักษณะการทำงานเช่นนี้จะมีตัวกลางสำหรับการดำเนินการที่เรียกว่ามิดเดิลแวร์ (Middleware) ระบบประมวลผลแบบกระจาย คือ กลุ่มของโฮสต์ (Host) ที่ทำงานผ่านระบบเครือข่ายโดยแต่ละโฮสต์จะรันมิดเดิลแวร์และเพื่อให้บริการคอมพิวเตอร์ที่ทำงานบนโฮสต์นั้น ๆ กับผู้เรียกใช้ผ่านเครือข่ายเช่นกัน โดยที่คอมพิวเตอร์นั้น ๆ จะเป็นรูปแบบเดี่ยวหรืออยู่ในรูปแบบกลุ่มหรืออื่น ๆ ภาพที่ 1.11 แสดงแนวทางการทำงานของระบบแบบกระจาย



ภาพที่ 1.11 แสดงแนวคิดการประมวลผลแบบกระจาย

ไคลเอนต์/เซิร์ฟเวอร์

โดยพื้นฐานของไคลเอนต์/เซิร์ฟเวอร์แล้วจะกล่าวถึงพื้นฐานของการใช้งานแบบนี้รวมถึงสรุปเกี่ยวกับหลักการของไคลเอนต์และเซิร์ฟเวอร์ การเตรียมทรัพยากรหรือส่วนประกอบต่าง ๆ ทั้งทางด้านโปรแกรมและฮาร์ดแวร์ที่จะใช้จัดการเป็นบางส่วน โดยปกติแล้วระบบนี้มี 2 ส่วน คือ ไคลเอนต์เป็นตัวถูกเป็นผู้ที่จะใช้บริการที่มีอยู่บนฝั่งของแม่ (Server) โดยส่วนที่เป็นตัวเชื่อมระยะกลางคือเครือข่ายคอมพิวเตอร์ซึ่งอาจเป็น LAN WAN Internet Intranet หรืออื่น ๆ ก็ได้



ภาพที่ 1.12 แสดงแนวคิดการให้บริการเชิงแอปพลิเคชัน

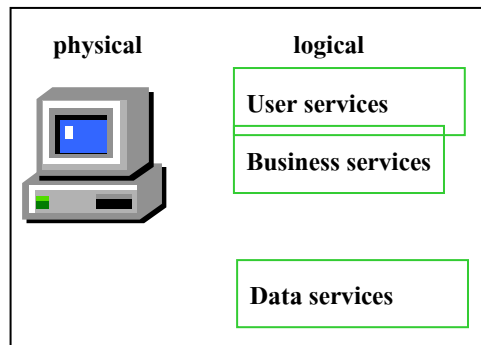
ในการทำงานกับเครื่อง ๆ เดียวผู้ใช้อาจมองไม่ออกว่ามีการแบ่งการใช้งานแบ่งออกเป็น ส่วน ๆ ในความรู้สึกรู้สึกของผู้จะเหมือนกับการเข้าถึงฐานข้อมูลโดยตรง แต่ความจริงแล้วจะถูกแบ่ง ออกเป็นชั้น ๆ แนวคิดของ 2-tier จะแยกเป็นส่วนของ Database Logic ออกจากส่วนอื่น ๆ โดยจะ แยก Database Logic ไว้ที่เซิร์ฟเวอร์แต่ส่วนของ Presentation, Business, Database logic ส่วนของ ไคลเอนต์ไว้ที่ไคลเอนต์โดย database จะอยู่บนเซิร์ฟเวอร์โดยที่บางส่วนจะอยู่บนไคลเอนต์ โดยที่ ส่วนต่าง ๆ อธิบายได้ดังนี้

- Presentation Logic เป็นส่วนของการติดต่อกับผู้ใช้ซึ่งจะต้องมีความสวยงาม ใช้งานง่ายและสะดวกต่อการใช้งาน
- Business Logic ส่วนของการออกแบบซอฟต์แวร์ทำหน้าที่เป็นตัวกลาง สำหรับผู้ใช้กับส่วนของการจัดการฐานข้อมูล เช่นการนำเอาข้อมูลที่ผู้ใช้ป้อนเข้ามาบันทึก หรือนำเอาความต้องการจากส่วนของ Presentation Logic มาประมวลผลแล้วดึงข้อมูลจากส่วนของ ฐานข้อมูล Database Logic เป็นต้น
- Database Logic ส่วนของฐานข้อมูลซึ่งในปัจจุบันคือ กลุ่มของ SQL เซิร์ฟเวอร์ต่าง ๆ นั่นเองที่มีขีดความสามารถในการจัดการข้อมูลให้ค้นหาได้รวดเร็ว การปกป้อง การสร้างควมมีเสถียรให้กับข้อมูล เป็นต้น

ถ้าเป็นการเขียนโปรแกรมแบบใช้งานเดี่ยว (Stand Alone) การเข้าถึงข้อมูลในฐานข้อมูลใน แต่ละระดับชั้นจะอยู่บนเครื่องเดียว ซึ่งมีข้อแตกต่างกับการใช้งานแบบไคลเอนต์/เซิร์ฟเวอร์โมเดล ที่ต้องใช้งานคอมพิวเตอร์ที่ซับซ้อน โดยอาจมีแนวคิดการจัดแยกส่วนต่าง ๆ ดังนี้

- Presentation Logic จะอยู่บนไคลเอนต์
- DBMS จะอยู่บนเซิร์ฟเวอร์
- Business Logic อาจอยู่บน DBMS Logic หรืออยู่บนไคลเอนต์หรือเอาไว้ที่ เซิร์ฟเวอร์ก็ได้

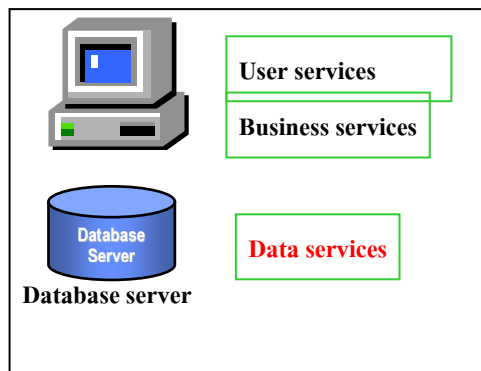
- เทียเดี่ยว



ภาพที่ 1.13 แสดงเทียเดี่ยว

ทูเทียร์โมเดล (2-tier model)

เทียร์แบบนี้จะแยกส่วนสำหรับการ User Service และ Business Service เอาไว้ด้วยกัน และ ส่วนของ Data Service จะแยกไว้ต่างหาก ซึ่งหากพิจารณาการติดตั้งโปรแกรมที่พัฒนาขึ้นอาจกล่าวได้ว่า ส่วนของฐานข้อมูลนั้นจะอยู่บนดาต้าเบสเซิร์ฟเวอร์ และส่วนของโปรแกรมจะติดตั้งไว้บนเครื่องของผู้ใช้ เช่น การพัฒนาโปรแกรมติดต่อฐานข้อมูลโดยตรง ซึ่งผู้พัฒนาโปรแกรมจะต้องเขียนส่วนโปรแกรมติดต่อ การจัดการคอมโพเนนต์ต่าง ๆ ไว้ที่เครื่องผู้ใช้ และการทำงานของโปรแกรมจะใช้เครือข่ายช่วยส่งข้อมูลเพื่อบันทึก แก้ไข หรือคำนวณกับฐานข้อมูลโดยตรง โดยที่ผู้พัฒนาโปรแกรมจะต้องสร้างส่วนการจัดการการเข้าใช้งานของผู้ใช้ด้วย แนวคิดแสดงดังภาพ 1.14



ภาพที่ 1.14 แสดงทูเทียร์

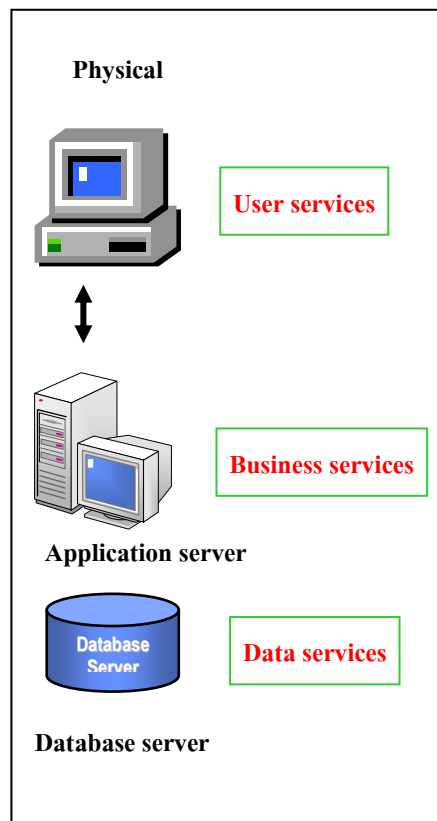
การหาตำแหน่งของ Business Logic (Locating the Business Logic) มีวิธีการ 3 ทางที่จะแยกจัดการ distribute business logic คือ ติดตั้งไว้ที่ ไคลเอนต์ (Fat client) หากติดตั้งไว้ที่เซิร์ฟเวอร์ (Fat Server) โดยแยกเป็นส่วน ๆ ทั้งข้าง ไคลเอนต์และเซิร์ฟเวอร์

ข้อดีของ Fat Sever คือ ง่ายต่อการปรับปรุงในส่วนของแอปพลิเคชัน โดยที่ไม่เกี่ยวกับไคลเอนต์ ข้อมูลที่สำคัญถูกซ่อนจากไคลเอนต์ ง่ายต่อการปรับแก้ไขโค้ดแบบส่วนกลาง ลดแบนวิดท์ ถ้าประมวลผลบนเซิร์ฟเวอร์งานสำคัญจะใช้ทำและมีความเสถียรดีกว่า ไคลเอนต์จะได้ทำงานง่ายขึ้นยิ่งหากไคลเอนต์มีความแตกต่างกันมาก

ข้อดีของ Fat Client ถ้าเปลี่ยนแปลงที่ไคลเอนต์จะไม่กระทบเซิร์ฟเวอร์ทำให้เซิร์ฟเวอร์มีความเสถียร ง่ายต่อโปรแกรม เพราะ ใช้เครือข่ายน้อย และมีรูปแบบการเข้าถึงได้หลายรูปแบบเช่น GUI แตกต่างกัน

ทรีเทียร์โมเดล(The 3-tier Model)

เทียร์แบบนี้จะแยกส่วน User Service, Business Service และ Data Service เป็นอิสระจากกัน ซึ่งสามารถกำหนดเป็นอิสระทั้งทางกายภาพและการจัดการซอฟต์แวร์ นอกจากนั้นยังอาศัยความสามารถของการจัดการ Business Service ด้วยมิดเดิลแวร์ที่มีประสิทธิภาพด้วย อีกทั้งสนับสนุนการพัฒนาโปรแกรมแบบกระจายได้เป็นอย่างดีอีกด้วย แนวคิดแสดงดังภาพ



ภาพที่ 1.15 แสดงทรีเทียร์

ส่วนสำคัญคือ ถ้าเป็น mainframe host จะแยกและเป็น database ที่ใหญ่มากบางที่เรียกว่า back-end server ซึ่ง เซิร์ฟเวอร์โดยปกติแล้วจะทำการแบ่งเป็นแอปพลิเคชันลอจิกบางที่เรียกว่ามิดเดิลแวร์เซิร์ฟเวอร์ ข้อดีของทรีเทียร์จะมากกว่าทูเทียร์ เพราะส่วนที่เป็น แอปพลิเคชันลอจิกจะอิสระจากไคลเอนต์และ Back-End อีกทั้งสามารถทำงานได้ง่ายด้วยข้อมูลจากหลากหลายทรัพยากร สามารถจัดการได้หลาย ๆ Back-End Server อีกด้วย แต่อย่างไรก็ตามปัญหาคือ ความซับซ้อนสูงที่การจัดการเครือข่าย การจัดการข้อมูล การดูแลรักษาและการพัฒนา ระบบการทำงานแบบนี้ เช่น

1. ระบบ เอทีเอ็ม

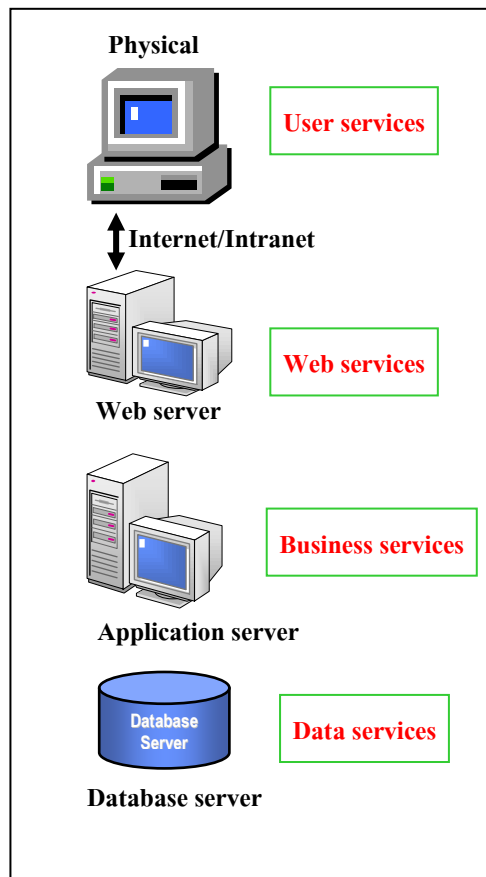
- เครื่อง ATM คือไคลเอนต์
- Middle-tier คือ Server เตรียมจัดการเรื่องของการตรวจสอบยอดเงิน การโอน การส่งคิวรี ไปยัง back-end
- Back-End Server
 - จัดการเรื่องบัญชี
 - จัดการเรื่องการใช้งานพร้อมกัน การประมวลผลโปร่งใสเหมือนการใช้งานอยู่คนเดียว

2. เว็บแอปพลิเคชันต่าง

- Web browser คือไคลเอนต์
- คอมพิวเตอร์ฝั่งที่อยู่เว็บเซิร์ฟเวอร์ที่มาจากมิดเดิลเทียร์
- Back-End Server จะบรรจุค่าดาต้าเบสอยู่

มัลติเทียร์โมเดล(Multi-tier Model หรือ N-Tier)

เทียร์แบบนี้ทำงานสนับสนุนการประมวลแบบกระจายที่มีความแตกต่างกันทั้งระบบปฏิบัติการรูปแบบการติดต่อ รูปแบบของทรัพยากร ซึ่งอาจจะมีเทียร์ต่าง ๆ ที่กล่าวมาแล้วเป็นส่วนประกอบก็ได้ ยกตัวอย่างเช่น ระบบที่ออกแบบทรีเทียร์จำเป็นต้องติดต่อกับเว็บเซิร์ฟวิส ต้องติดต่อกับระบบของทูเทียร์ด้วย เป็นต้น แสดงแนวคิดดังภาพ



ภาพที่ 1.16 แสดงมัลติเทียร์

การประมวลของเทียร์ประเภทนี้จะซับซ้อนและต้องมีเทคนิคต่าง ๆ ที่ใช้สำหรับการจัดการ ตัวอย่างเช่น มัลติเทียร์เว็บแอปพลิเคชัน (**Multi-tier web application**) โดยที่

- web browser คือ client software เป็น 1-tier
- web page คอมโพเนนต์คือเทียร์ที่สอง (Middle-tier)
- DB-tier คือ database front-end และเทียร์ที่สี่คือ Database ที่มีการใช้เทคนิคของ data mining ก็ได้

ข้อดีของการพัฒนาระบบแบบไคลเอนต์/เซิร์ฟเวอร์ คือ ทำให้ระบบที่สร้างขึ้นสามารถใช้งานได้อย่างกว้างขวางด้วยราคาประหยัด การประมวลผลข้อมูลที่ตำแหน่งบนเซิร์ฟเวอร์จะทำให้ลดปัญหาเรื่องความคับคั่งของเครือข่าย ความเร็วของการตอบรับและลดความต้องการแบนวิดท์ที่มากขึ้น สามารถทำงานแบบระบบเปิดได้ โดยกำหนดมาตรฐานกลางเอาไว้ แต่ก็มีข้อเสียคือ อาจเกิดปัญหาคอขวดของเซิร์ฟเวอร์ การใช้การกระจายงานซับซ้อน รวมถึงต้องการการพัฒนา บำรุงรักษา หรือปรับปรุงเวอร์ชัน เป็นต้น

สรุป

ระบบเครือข่ายคอมพิวเตอร์คือการนำเอาคอมพิวเตอร์ตั้งแต่สองเครื่องขึ้นไปมาเชื่อมต่อกัน โดยวิธีการใดวิธีการหนึ่งทั้งที่มีสายหรือไม่ก็ได้ เพื่อให้สามารถแบ่งปันทรัพยากร ข้อมูล และ โปรแกรมกันได้ องค์ประกอบของการเชื่อมต่อกันจะต้องมีรูปแบบสถาปัตยกรรม มาตรฐานการติดต่อเรียกว่าโปรโตคอล และการออกส่งเข้าถึงข้อมูล การพัฒนาโปรแกรมสำหรับระบบเครือข่าย แบ่งเป็นสามระดับคือ Presentation Logic, Business Logic และ Database Logic การจัดการสิ่งที่ให้บริการเรียกว่าเซิร์ฟเวอร์ที่ให้บริการ และรูปแบบการให้บริการเป็นแบบรวมศูนย์ แบบกระจาย รูปแบบการเขียนโปรแกรมเป็นไปตามมาตรฐานของโอเอสไอโมเดล โดยส่งผ่านข้อมูลระดับล่างคือไบต์สตรีมหรือแบบอ็อบเจกต์โอเรียลเตด และการให้บริการอาจเป็นแบบอิตเทอทิฟและแบบคอนเคอร์เรนซีเซิร์ฟเวอร์

แบบฝึกหัดท้ายบท

1. เครือข่ายคอมพิวเตอร์คืออะไร มีวัตถุประสงค์เพื่ออะไร
2. หากท่านต้องการเชื่อมต่อระบบเครือข่ายให้กับร้านขายหนังสือท่านจะออกแบบโดยใช้โมเดลเครือข่ายแบบใดเพราะเหตุใด
3. ไอพี และแมคแอดเดรส มีความสำคัญอย่างไร
4. ไอพีเวอร์ชัน 6 แตกต่างจากไอพีเวอร์ชัน 4 อย่างไร มีข้อดีข้อเสียอย่างไร จงอธิบาย
5. การแบ่งปันทรัพยากรใดที่จะใช้ในการเขียนโปรแกรมสำหรับการสนทนากัน
6. จงบอกหน้าที่และความสำคัญของ Presentation Logic Business Logic และ Database Logic
7. จงอธิบายรูปแบบการออกแบบเครือข่ายแบบ Peer-to-Peer และ Client-Server
8. เครือข่ายอินเทอร์เน็ตในปัจจุบันออกแบบโดยใช้สถาปัตยกรรมแบบใด จงยกตัวอย่างให้พอสังเขป
9. การเขียนโปรแกรมที่เรียกใช้งานเมธอดที่อยู่ในระยะไกลถือว่าการแบ่งปันทรัพยากรเพื่อใช้งาน
10. จงอธิบายเกี่ยวกับมาตรฐานของโปรโตคอลต่อไปนี้
 - TCP/IP
 - HTTP
 - POP3
 - FTP
 - SOAP

บทที่ 2

ซ็อกเก็ต

เนื้อหาการเขียนโปรแกรมเพื่อติดต่อสื่อสารกันผ่านระบบเครือข่ายในหนังสือเล่มนี้ เริ่มจากการพัฒนาโปรแกรมในระดับของการส่งข้อมูลในรูปแบบของสายของข้อมูล (Data Stream) และการพัฒนาโปรแกรมในระดับที่ทำงานได้อย่างมีประสิทธิภาพและรวดเร็วสำหรับการส่งผ่านข้อมูล ดังกล่าวก็คือการเขียนโปรแกรมเพื่อติดต่อกันแบบซ็อกเก็ต (Socket) ซึ่งเป็นการเขียนโปรแกรมที่ถือว่าส่งผ่านข้อมูลกันในระดับของชั้นเน็ตเวิร์กที่ต่ำที่สุดที่เรียกว่าการติดต่อกันแบบจุดต่อจุด ที่ต้องเขียนโปรแกรมสำหรับจัดการการติดต่อจัดการให้บริการ และการเข้าถึงข้อมูลต่าง ๆ เองทั้งหมด ในเนื้อหาบทนี้นำเสนอรูปแบบการเขียนโปรแกรมแบบนี้ โดยเริ่มจากหลักการเบื้องต้นของการเขียนโปรแกรม การพัฒนาโปรแกรมโดยเรียกผ่านไอพีแอดเดรส การพัฒนาโปรแกรมแถมติดต่อกับทีซีพีโดยผ่านซ็อกเก็ต

ความหมายและความสำคัญ

ซ็อกเก็ต คือ จุดปลายของการติดต่อสื่อสารในชั้นของทรานสปอร์ตเลเยอร์ (Transport layer) ในชั้นนี้เรามักจะกล่าวในระดับของระบบเครือข่ายคือติดต่อสื่อสารกันระหว่างเซิร์ฟเวอร์และไคลเอนต์ในลักษณะของจุดต่อจุด ซึ่งการติดต่อจะมีโปรโตคอลที่ใช้ในการสื่อสาร คือ ทีซีพี หากเปรียบเทียบซ็อกเก็ตกับโทรศัพท์ที่ต้องติดต่อกันระหว่างผู้รับกับผู้ส่งกล่าวคือต้องรอการตอบรับอีกฝั่งหนึ่งของการสื่อสารแต่ละเดิวกั้น โปรโตคอล ยูดีพี จะเปรียบเสมือนกับกล่องจดหมายที่ไม่จำเป็นต้องมีผู้รับอีกฝั่งอยู่ตลอดเวลาเพียงแต่ส่งจดหมายในกล่องแล้วก็เสร็จสิ้นภารกิจเป็นต้น การสื่อสารโดยใช้ซ็อกเก็ตนี้จะมีส่วนสำคัญอยู่สองส่วนคือไคลเอนต์และเซิร์ฟเวอร์ โครงสร้างข้อมูลของซ็อกเก็ตนั้นมีหลากหลายโครงสร้างจะขึ้นอยู่กับนำไปใช้ เช่น จุดปลาย (End-Point) ระหว่างโพรเซส (Process) ในเครื่องเดียวกัน กรณีนี้อาจแทบไม่ได้ใช้งาน จุดปลายระหว่างโพรเซสของเครือข่าย เป็นต้น

ชนิดของซ็อกเก็ต

1. SOCK_STREAM ซ็อกเก็ตประเภทนี้จะใช้สำหรับส่วนที่เรียกว่า Connection-Oriented คือจะใช้ ทีซีพี โปรโตคอลในการส่งจะต้องมีการสร้างการติดต่อกันระหว่างเซสชัน (Session) ทั้งสองด้านเสียก่อนแล้วจึงส่งข้อมูล โดยลักษณะการส่งข้อมูลนั้นจะทำการส่งแบบ Stream Oriented