บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 องค์กรข้อมูล (Data Organization)

ในแต่ละแฟ้มข้อมูลของฐานข้อมูลหนึ่งๆ จะประกอบไปด้วยส่วนย่อยของข้อมูล(Data Element) ซึ่งได้แก่ หมายเลขประจำตัว วันที่ ตำแหน่ง ที่อยู่ เป็นต้น แล้วแต่ชนิดของข้อมูล ส่วนย่อยของข้อมูลเหล่านี้จะได้รับการจัดรวบรวม จัดเก็บ และการเรียกใช้เมื่อต้องการจากฐานข้อมูลนั้นๆ ดังนั้นในการที่เราจะออกแบบฐานข้อมูลหนึ่งจึงมีความจำเป็นที่จะต้องเข้าใจเกี่ยวกับระบบแฟ้มข้อมูล ส่วนย่อยของข้อมูล ตลอดจนถึงวิธีการใช้งานด้วย

ในระบบคอมพิวเตอร์หน่วยที่เล็กที่สุดของข้อมูลคือ บิต (bit) ซึ่งย่อมาจาก binary digit หรือเลขฐานสองได้แก่ 0 และ 1 และได้มีการรวมกลุ่มบิตหลายๆบิตเข้าด้วยกันเพื่อแทนอักขระ 1 ตัว ลักษณะการจัดกลุ่มข้อมูลแบบนี้เรียกว่า การจัดองค์กรข้อมูล โดยมีการแบ่งเป็นลำดับชั้นดังนี้

2.1.1 อักขระ (Character) เป็นสัญลักษณ์ในภาษาเขียนของเรา ได้แก่ ตัวอักษร, ตัวเลข และสัญลักษณ์พิเศษ เช่น +, -, * , /, (,) เป็นต้น

2.1.2 เขตข้อมูล (Data Field) หมายถึง ส่วนย่อยของข้อมูลที่จะประกอบกันขึ้นเป็นระเบียน เกิดจากการรวมกลุ่มของอักขระตั้งแต่ 1 ตัวขึ้นไป มีความหมายถึงสิ่งหนึ่งสิ่งใด เช่น เลขที่, ชื่อ, วันเกิด, เงินเดือน เป็นต้น

2.1.3 ระเบียน (Record) หมายถึง ข้อมูลที่ได้จากการรวมกลุ่มของเขตข้อมูลที่เกี่ยวข้องเข้าด้วยกันเป็นข้อมูลหนึ่งหน่วย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

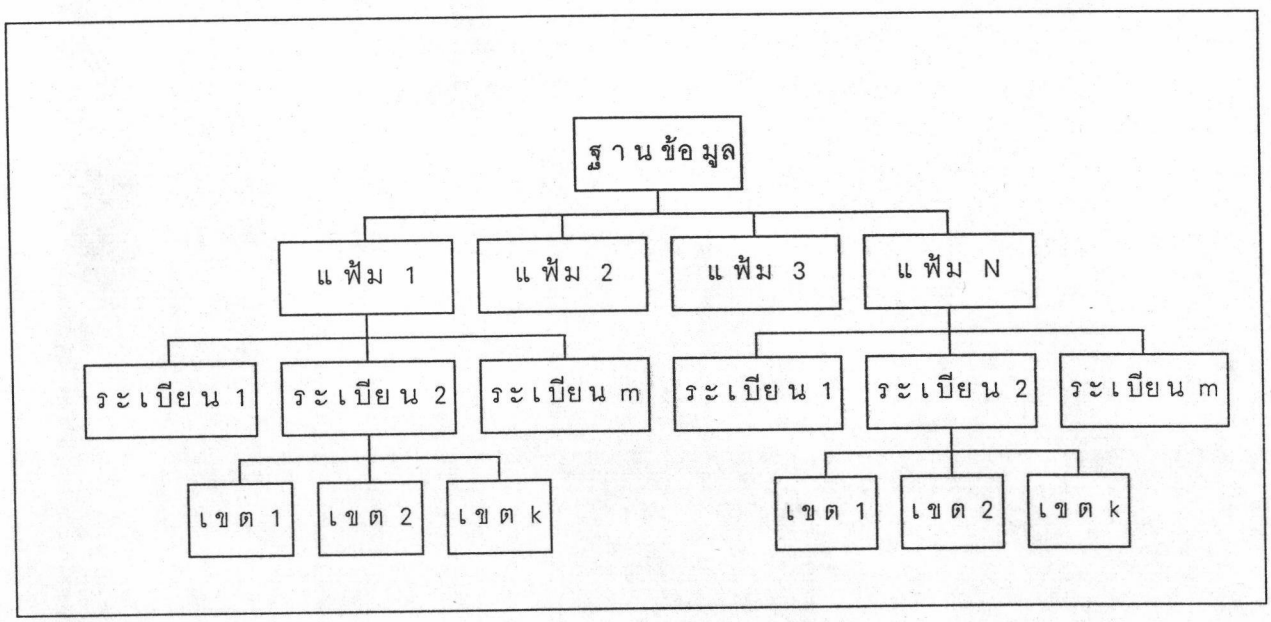
2.1.4 แฟ้มข้อมูล (File) หมายถึง กลุ่มของระเบียนตั้งแต่ 1 ระเบียนขึ้นไปรวมกันมีความหมายเกี่ยวพันในเรื่องเดียวกัน เช่นระเบียนของพนักงานทั้งหมดในบริษัท

2.1.5 ฐานข้อมูล (Database) หมายถึง กลุ่มของแฟ้มข้อมูลที่มีความสัมพันธ์เกี่ยวข้องซึ่งกันและกัน ภายใต้จุดประสงค์ร่วมกันอย่างหนึ่ง

การจัดองค์กรข้อมูล เป็นมองในทัศนะของผู้ใช้ ที่จะดูเฉพาะความสัมพันธ์ตามที่กล่าวมาข้างต้น โดยไม่ได้คำนึงว่าข้อมูลนั้นเก็บอยู่ที่ไหนบนสื่อบันทึกข้อมูล ซึ่งจะเรียกว่าเป็นการมองในเชิงตรรก (Logical View)

ชื่อ	ที่อยู่	เบอร์โทรศัพท์
นายเด่น ดวงดี	15/26 ถ.ลาดพร้าว กทม.	533-4432
นายนิยม กลิ่นหอม	457 ถ.สุขุมวิท กทม.	384-1099
นางน้อย พุ่มพวง	670/2 ถ.พหลโยธิน กทม.	279-0277

รูปที่ 2.1 ตัวอย่างลำดับชั้นขององค์กรแฟ้มข้อมูล



รูปที่ 2.2 องค์กรของแฟ้มข้อมูล

2.2 ระบบแฟ้มข้อมูลของยูนิกซ์ (UNIX File System)

2.2.1 ระบบแฟ้มข้อมูล (File System) เป็นส่วนประกอบที่สำคัญของระบบปฏิบัติการยูนิกซ์โดยปกติระบบแฟ้มข้อมูลจะประกอบด้วย

2.2.1.1 วิธีเข้าถึง (Access Methods) จะเกี่ยวกับลักษณะของการเข้าถึง (access) ข้อมูลที่ เก็บอยู่ในแฟ้มข้อมูล

2.2.1.2 การจัดการแฟ้ม (File Management) จะเกี่ยวกับข้อกำหนด และกลไกในการเก็บ (Store) การอ้างอิงถึง (Reference) การใช้ร่วมกัน (Share) และการป้องกันรักษา (Secure)

2.2.1.3 การจัดการหน่วยเก็บช่วย (Auxiliary Storage Management) จะเกี่ยวกับการกำหนดขนาดพื้นที่สำหรับแฟ้มข้อมูลภายในหน่วยเก็บรอง

2.2.1.4 บูรณภาพของแฟ้มข้อมูล (File Integrity) จะเกี่ยวกับการรับรองว่าสารสนเทศภายในแฟ้มข้อมูลจะถูกต้อง

2.2.2 หน้าที่ของระบบแฟ้มข้อมูล

2.2.2.1 ทำให้ผู้ใช้สามารถสร้าง แก้ไข และลบแฟ้มข้อมูล

2.2.2.2 ทำให้ผู้ใช้สามารถใช้แฟ้มข้อมูลร่วมกันได้กับผู้อื่นที่อยู่บนระบบเดียวกัน

2.2.2.3 กำหนดกลไกในการควบคุมการเข้าถึงแฟ้มข้อมูลที่ใช้ร่วมกันเช่น Read Access, Write Access, Execute Acess หรือรูปแบบการรวมกันของการเข้าถึงข้างต้น

2.2.2.4 ทำให้ผู้ใช้สามารถกำหนดโครงสร้างของแฟ้มข้อมูล ให้เหมาะสมกับโปรแกรมประยุกต์ (Application Program)

2.2.2.5 ทำให้ผู้ใช้สามารถถ่ายโอนข้อมูลระหว่างแฟ้มข้อมูล

2.2.2.6 มีความสามารถในการสำรอง (Backup) และการกู้ (Recovery) แฟ้มข้อมูล

2.2.2.7 ทำให้ผู้ใช้สามารถอ้างถึงชื่อแฟ้มข้อมูลโดยใช้ Symbolic Names มากกว่าจะใช้ Physical Device Names

2.2.2.8 ในสภาพแวดล้อมที่เปลี่ยนแปลงตลอดเวลา ข้อมูลต้องมีความปลอดภัยและเป็นเอกเทศ (Private) เช่น ระบบการโอนเงินทางคอมพิวเตอร์ หรือ ระบบทะเบียนอาชญากร เป็นต้น กรณีข้างต้น File System ต้องกำหนดให้มีความสามารถในการเข้ารหัสลับ (Encyption) และการถอดรหัสลับ (Decryption)

2.2.2.9 ทำให้ผู้ใช้มีความรู้สึกที่คุ้นเคย และเข้าใจได้ง่ายในการใช้แฟ้มข้อมูลในเชิงตรรกโดยผู้ใช้ไม่จำเป็นต้องรู้เรื่องเกี่ยวกับการเก็บข้อมูลภายในสื่อบันทึกว่าทำอย่างไร หรือเราจะเข้าถึงข้อมูลอย่างไร ที่เรียกว่า การมองเชิงกายภาพ (Physical View)

2.2.3 ระบบแฟ้มข้อมูลของยูนิกซ์นั้นประกอบด้วยแฟ้ม 3 ประเภท คือ

2.2.3.1 แฟ้มธรรมดา (Regular File) สำหรับแฟ้มข้อมูลที่อยู่ในระบบปฏิบัติการยูนิกซ์จะมีลักษณะที่แตกต่างจากแฟ้มข้อมูลที่อยู่ในระบบปฏิบัติการอื่นๆ คือข้อมูลแต่ละตัวจะอยู่เรียงลำดับกันไปไม่มีลักษณะของระเบียน หรือตัวจบแฟ้มข้อมูล (File Terminator) แต่ภายในยูนิกซ์ก็เอื้อให้สามารถเข้าหาข้อมูลทั้งในลักษณะแบบเรียงลำดับ (Sequential) และแบบสุ่ม (Random) โดยอาศัยตัวชี้ของแฟ้มข้อมูลช่วย การแทรกข้อมูลหรือการลบข้อมูลที่อยู่ส่วนกลางของแฟ้มข้อมูลนั้นไม่อาจทำได้ ซึ่งถ้าหากจำเป็นผู้ใช้ก็ต้องสร้างแฟ้มข้อมูลนั้นขึ้นมาใหม่

แฟ้มข้อมูลเหล่านี้จะมีหมายเลขประจำแฟ้มอยู่เรียกว่า i-number ซึ่งจะเป็นตัวชี้ไปยังรายการในแถวลำดับ (Array) ที่เรียกว่า i-node ที่เก็บรายละเอียดของแฟ้มข้อมูลนั้น เช่น ประเภทของแฟ้ม ขนาดของแฟ้ม วันที่ที่สร้าง วันที่ที่ปรับปรุงครั้งสุดท้าย

2.2.3.2 แฟ้มสารบบ (Directory File) แฟ้มสารบบก็คือ แฟ้มข้อมูลที่เก็บชื่อของแฟ้มข้อมูลต่างๆเอาไว้ (ซึ่งอาจเป็นชื่อของสารบบก็ได้) เพื่อใช้แบ่งแยกแฟ้มข้อมูลเหล่านั้นออกเป็นกลุ่ม โดยสามารถจะแสดงความสัมพันธ์ของกลุ่มเหล่านั้นได้ การที่สารบบสามารถมีสารบบซ้อนอยู่ได้ ทำให้โครงสร้างของแฟ้มข้อมูลต่างๆอยู่ในลักษณะลำดับขั้น เช่นเดียวกับรูปต้นไม้ (Tree) ซึ่งสารบบที่อยู่บนสุดคือ ราก(Root) โดยมีสัญลักษณ์ว่า / ภายในสารบบจะเก็บเพียง i-number (2Bytes) กับชื่อของแฟ้มข้อมูล (14Bytes) การเรียกใช้แฟ้มข้อมูลจะต้องระบุ Pathname ซึ่งประกอบด้วยสารบบที่อยู่ตั้งแต่บนสุดเรียงลำดับลงมา ซึ่งแต่ละสารบบต้องคั่น

ด้วยเครื่องหมาย / เช่น /dir1/dir2/file เมื่อทำการเรียกใช้แฟ้มใด เคอร์เนล (Kernel) จะหาค่าของ i-number ของสารบบแรกจากสารบบราก เพื่อเอาไปค้นหาในตาราง i-node จากตัวอย่าง เคอร์เนลทำการค้นหาชื่อ dir1 จากสารบบราก ซึ่งจะได้ i-number แล้วนำไปค้นในตาราง i-node เพื่อหาตำแหน่ง dir1 ในจานบันทึก (Disk) จากนั้นก็ค้นหาชื่อของ dir2 จากสารบบ dir1 ได้ค่า i-number เพื่อนำไปค้นในตาราง i-node อีก ให้ได้ตำแหน่งของ dir2 ในจานบันทึก จากนั้นก็ค้นหาชื่อ file ใน dir2 ได้ค่า i-number เพื่อนำไปค้นในตาราง i-node อีก ได้ตำแหน่งของ file ในจานบันทึก

2.2.3.3 แฟ้มพิเศษ (Special File) แฟ้มข้อมูลนี้จะแตกต่างจากแฟ้มข้อมูลธรรมดาตรงที่ไม่ได้ทำการเก็บข้อมูล แต่แฟ้มข้อมูลประเภทนี้เป็นเพียงชื่อที่แทนอุปกรณ์ต่างๆที่ต่อเข้ากับระบบ เช่น จานบันทึก แถบบันทึก เครื่องพิมพ์ เป็นต้น ซึ่งจะอยู่ในสารบบ /dev เช่น / dev/tty00 ซึ่งหมายถึงพอร์ตเทอร์มินอลหมายเลข 0 เป็นต้น

2.2.4 ตัวบ่งลักษณะแฟ้มข้อมูล (File Descriptor) เป็นกลุ่มระเบียนที่ใช้ควบคุม (Control Block) แฟ้มข้อมูล โดยจะเก็บสารสนเทศที่ระบบใช้บริหารแฟ้มข้อมูล เช่น

2.2.4.1 ชื่อภายใน (Symbolic File Name)

2.2.4.2 ตำแหน่งที่เก็บแฟ้มข้อมูลในหน่วยเก็บข้อมูลรอง

2.2.4.3 รูปแบบของแฟ้มข้อมูล (แถวอักขระ บล็อก เรียงลำดับ เป็นต้น)

2.2.4.4 ชนิดอุปกรณ์ที่ใช้เก็บแฟ้มข้อมูล

2.2.4.5 ข้อมูลที่ใช้ควบคุมการเข้าถึงแฟ้มข้อมูล (สิทธิในการใช้)

2.2.4.6 ชนิดของข้อมูลที่จัดเก็บ (Data File, Object Program, Source Program)

2.2.4.7 วันที่และเวลาที่สร้างแฟ้มข้อมูล

2.2.4.8 วันที่ลบแฟ้มข้อมูล

2.2.4.9 วันที่และเวลาที่แก้ไขครั้งสุดท้าย

2.2.4.10 ข้อมูลจำนวนครั้งของการเข้าถึงแฟ้มข้อมูล

ปกติตัวบ่งลักษณะแฟ้มข้อมูลจะเก็บที่หน่วยเก็บข้อมูลรอง แต่จะถูกนำมาไว้ที่หน่วยเก็บข้อมูลหลัก (Primary Storage) เมื่อตอนเปิดแฟ้มข้อมูล ตัวบ่งลักษณะแฟ้มข้อมูลนี้จะถูกควบคุมโดยระบบปฏิบัติการ

2.2.5 การควบคุมการเข้าถึงแฟ้มข้อมูลโดย User Classes

ปกติยูนิกซ์จะแบ่งประเภทของผู้ใช้ได้ ดังนี้

2.2.5.1 Owner ปกติจะหมายถึงผู้ใช้ที่เป็นคนสร้างแฟ้มข้อมูล

2.2.5.2 Specified User จะหมายถึงผู้ใช้ที่ Owner กำหนดว่าให้ใช้แฟ้มข้อมูลได้

2.2.5.3 Group หรือ Project จะหมายถึงผู้ใช้ที่เป็นสมาชิกของกลุ่มงานที่ต้องใช้แฟ้มข้อมูลนั้น

2.2.5.4 Public จะหมายถึงกำหนดให้ผู้ใช้ทุกคนสามารถเข้าถึงแฟ้มข้อมูลได้ ปกติแล้วจะกำหนดให้อ่าน (Read) หรือกระทำการ (Execute) กับแฟ้มข้อมูลได้เท่านั้น แต่ห้ามเขียน (Write) แฟ้มข้อมูล

2.3 ระบบฐานข้อมูล (DATABASE SYSTEM)

ระบบฐานข้อมูลเป็นการเก็บรวบรวมข้อมูลที่มีความสัมพันธ์กันไว้ด้วยกัน เพื่อให้ผู้ใช้และงานประยุกต์ต่างๆ สามารถดำเนินการกับข้อมูลนั้นๆได้ โดยอาศัยระบบจัดการฐานข้อมูล (Database Management System: DBMS) ซึ่งเป็นชุดคำสั่งหรือโปรแกรมที่คอยดูแลข้อมูลต่างๆในฐานข้อมูล

ในองค์กรที่มีการจัดการกับข้อมูลในรูปแบบแฟ้มข้อมูลแบบสัญนิยม (Conventional File) นั้น จะมีโปรแกรมประยุกต์แต่ละโปรแกรมจัดการกับแฟ้มข้อมูลของตัวเอง ดังนั้นข้อมูลในองค์กรเหล่านี้ไม่จำเป็นที่จะต้องใช้ร่วมกัน แต่บางครั้งโปรแกรมที่ต่างกันก็อาจต้องการใช้ข้อมูลชุดเดียวกันแต่อาจจะอยู่ในรูปแบบที่ไม่เหมือนกัน ดังนั้นผลก็คือ จะมีข้อมูลชุดเดียวกันอยู่หลายแฟ้มข้อมูล ซึ่งจะก่อให้เกิดปัญหาต่างๆ เช่น

2.3.1 เกิดความซ้ำซ้อนของข้อมูล (Data Redundancy) ซึ่งนำไปสู่การเปลืองเนื้อที่จัดเก็บข้อมูล และเปลืองค่าใช้จ่ายในการบำรุงรักษา

2.3.2 ข้อมูลมีความไม่สอดคล้องกัน (Data Inconsistency) หมายถึงมีข้อมูลชุดเดียวกันอยู่หลายแฟ้มข้อมูล ดังนั้นอาจเกิดการเปลี่ยนแปลงที่ไม่เหมือนกัน เช่น เมื่อเปลี่ยนแปลงข้อมูลในแฟ้มหนึ่ง แต่ไม่เปลี่ยนแปลงในอีกแฟ้มหนึ่ง ทั้งๆที่เป็นข้อมูลชุดเดียวกัน

2.3.3 ข้อมูลอยู่กันอย่างกระจัดกระจาย (Data Isolation) ดังนั้นอาจมีรูปแบบของแฟ้มข้อมูล (File Format) ที่แตกต่างกัน จึงยากแก่การเขียนโปรแกรมประยุกต์ขึ้นมาสำหรับสืบค้น (Retrieve) ข้อมูลที่เหมาะสมได้

2.3.4 ความปลอดภัยของข้อมูล (Data Security) เนื่องมาจากว่าในระบบแฟ้มข้อมูลแบบสัญนิยม ข้อมูลอยู่อย่างกระจัดกระจาย และไม่มีการกำหนดสิทธิในการเข้าถึงข้อมูล

2.3.5 ข้อมูลไม่มีบูรณภาพ (Nonintegrity) ในระบบแฟ้มแบบสัญนิยมเมื่อต้องการเพิ่มเงื่อนไขต่าง ๆ เงื่อนไขนั้นก็จะต้องถูกเพิ่มเข้าไปในแฟ้มข้อมูลทุกๆแฟ้ม ถ้าเพิ่มไม่ครบแล้วข้อมูลก็จะไม่มีบูรณภาพ

2.3.6 ข้อมูลไม่สามารถใช้ร่วมกันได้

2.3.7 ไม่สามารถกำหนดมาตรฐานใช้บังคับร่วมกันได้

2.4 คุณลักษณะของระบบฐานข้อมูล (Characteristics Of Database System)

2.4.1 โปรแกรมและข้อมูลเป็นอิสระต่อกัน (Program&Data Independence) ในการประมวลผลแฟ้มข้อมูลแบบสัญนิยม โครงสร้างของแฟ้มข้อมูลจะแทรกอยู่ในตัวโปรแกรม ดังนั้นการเปลี่ยนแปลงใด ๆ ของโครงสร้างของแฟ้มข้อมูลอาจจะทำให้ต้องมีการเปลี่ยนแปลงโปรแกรมทั้งหมดที่เรียกใช้แฟ้มข้อมูลนั้น ในทางตรงกันข้ามระบบจัดการฐานข้อมูลจะเรียกใช้โปรแกรมที่ถูกเขียนขึ้นอย่างเป็นอิสระจากแฟ้มข้อมูล โครงสร้างของแฟ้มข้อมูลจะถูกเก็บแยกจากโปรแกรมที่เรียกใช้

2.4.2 ข้อมูลมีความเป็นอันหนึ่งอันเดียวกัน (Data Integration) หมายความว่า ฐานข้อมูลเป็นการรวบรวมข้อมูลซึ่งไม่ควรมีข้อมูลที่ซ้ำซ้อนโดยไม่จำเป็น หรือเป็นข้อมูลที่ใช้ไม่ได้ เป็นต้น

2.4.3 มีความเป็นบูรณภาพของข้อมูล (Data Integrity) หมายความว่าข้อมูลมีความน่าเชื่อถือ ไม่ผิดพลาด มีกฎเกณฑ์ในการควบคุมค่าของข้อมูล และในการดูแลรักษาระบบการทำงาน เราต้องแน่ใจว่าข้อมูลมีความสอดคล้องและไม่มีความลดหลั่น (Inconsistencies) เกิดขึ้นในฐานข้อมูล

2.4.4 แยกระหว่างมุมมองของข้อมูลเชิงตรรกกับเชิงกายภาพ (Seperate Logical And Physical Views Of Data) ในการประมวลผลแฟ้มข้อมูลแต่ละแฟ้มข้อมูล อาจจะมีการกำหนดความยาวของระเบียน จำนวนไบต์ในแต่ละระเบียน และแต่ละเขตข้อมูลอาจถูกกำหนดโดยไบต์เริ่มต้นภายในระเบียนกับความยาวของเขตข้อมูลนั้นๆ แต่ในลักษณะของระบบฐานข้อมูล ผู้ใช้อาจไม่จำเป็นต้องสนใจว่า แต่ละเขตข้อมูลอยู่ตำแหน่งไหน เพียงแต่อ้างชื่อถึงเขตข้อมูลนั้น ๆ ก็จะได้ค่าที่ต้องการ นั่นคือภายใต้แนวความคิดฐานข้อมูลจะพยายามแยกมุมมองของข้อมูลเชิงตรรกออกจากรายละเอียดเกี่ยวกับลักษณะทางกายภาพ

2.5 ประเภทของระบบฐานข้อมูล

2.5.1 ระบบฐานข้อมูลแบบรวมศูนย์ (Centralized Database System)

ระบบฐานข้อมูลแบบรวมศูนย์ หมายถึง ลักษณะของการจัดเก็บข้อมูลโดยระบบฐานข้อมูลจะประกอบด้วยข้อมูล และระบบจัดการฐานข้อมูลอยู่บน Hardware เดียวกัน แม้ว่าจะมีการเรียกใช้ข้อมูลจากหลายๆจุดที่ห่างไกลจากเครื่อง ก็ยังถือว่าเป็นเทคนิคการจัดเก็บข้อมูลแบบรวมศูนย์ เพราะข้อมูลทั้งหมด และการประเมินผลที่เกี่ยวข้องกับฐานข้อมูลยังคงอยู่ที่คอมพิวเตอร์เครื่องเดียว โดยส่วนที่เป็น Terminal ไม่ได้ทำหน้าที่ในการจัดเก็บหรือประมวลผลข้อมูลแต่อย่างใด

2.5.2 ระบบฐานข้อมูลแบบกระจาย (Distributed Database System)

ระบบฐานข้อมูลแบบกระจาย หมายถึง ระบบฐานข้อมูลที่มีฐานข้อมูล เก็บอยู่บนระบบคอมพิวเตอร์หลายๆเครื่อง (อาจจะเป็นเครื่องประเภทเดียวกัน หรือแตกต่างกันก็ได้) ที่ติดตั้งอยู่ตามที่ต่างๆ โดยที่ระบบคอมพิวเตอร์เหล่านี้มีการสื่อสารกันได้ ซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลที่อยู่ในเครื่องใดๆก็ได้ และผู้ใช้ไม่จำเป็นต้องรับทราบว่าฐานข้อมูลที่ตนต้องการนั้นจัดเก็บบนเครื่องใด

2.5.3 ระบบฐานข้อมูลแบบผู้ใช้บริการ/ผู้ให้บริการ (Client&Server Database System)

ระบบฐานข้อมูลแบบผู้ใช้บริการ/ผู้ให้บริการ หมายถึง ระบบฐานข้อมูลที่มีการแบ่งการประมวลผลระหว่าง Client ที่ประมวลผลโปรแกรมประยุกต์ที่เกี่ยวข้องกับฐานข้อมูล (Front-End System) ซึ่งจัดการหน้าจอทั้งหมดและการตรวจสอบการเข้า/ออกข้อมูลของผู้ใช้ และ Server ที่ประมวลบางส่วนหรือทั้งหมดของระบบจัดการฐานข้อมูล (Back-End System) ซึ่งจัดการประมวลผลที่เกี่ยวกับฐานข้อมูล และการเข้าถึงข้อมูลในหน่วยความจำสำรอง โดยที่ระบบคอมพิวเตอร์ทั้ง 2 ส่วนนี้มีการสื่อสารกันได้ ซึ่งการแบ่งการประมวลผลระหว่าง 2 ส่วนนี้จะช่วยลดจำนวนข้อมูลที่ส่งผ่านบนข่ายสื่อสารลง

2.6 แนวความคิดในการนำระบบเครือข่ายมาใช้

หลังจากการที่การพัฒนาระบบเครือข่ายได้ก้าวหน้าขึ้น มีการพัฒนาระบบซอฟต์แวร์ที่ทำให้มีการเชื่อมโยงคอมพิวเตอร์หลาย ๆ ระบบเข้าเป็นระบบเดียวกันได้ ทำให้การใช้ทรัพยากรต่าง ๆ ในระบบเกิดประสิทธิภาพสูงสุด ซึ่งทรัพยากรที่สามารถใช้งานร่วมกันได้นั้นสามารถแยกเป็นประเภทต่าง ๆ ได้ดังนี้

2.6.1 ข้อมูล (Data)

การใช้ข้อมูลร่วมกันนั้น หมายถึงขั้นตอนการประมวลผลเรียกข้อมูลโดยผู้ใช้ ข้อมูลอาจอยู่บนระบบเดียวกับข้อมูลที่เก็บอยู่ หรืออยู่บนระบบอื่น ๆ ก็ได้ ซึ่งการทำวิทยานิพนธ์นี้จะเน้นในส่วนการใช้ข้อมูลร่วมกัน โดยข้อมูลจะกระจายอยู่ทั่วไปในระบบเครือข่าย ChulaNet ตามขอบเขตที่ได้กำหนด

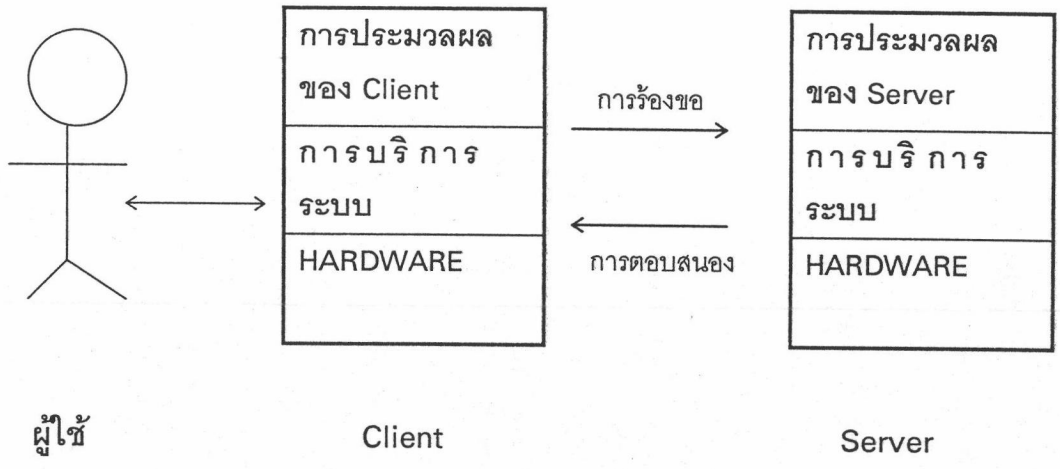
2.6.2 หน่วยประมวลผลกลาง (CPU)

การใช้หน่วยประมวลผลกลางร่วมกันนั้น สามารถทำได้โดยที่ผู้ใช้งานเรียกหรือทำการส่งโปรแกรมของตนมาประมวลผล ในขณะที่ผู้ใช้งานอื่น ๆ ก็ทำในลักษณะเดียวกันในช่วงเวลาเดียวกัน

2.6.3 โปรแกรม (Program)

การใช้โปรแกรมร่วมกันนั้นกระทำได้เหมือนกับการใช้หน่วยประมวลผลกลางร่วมกัน หรือเป็นการใช้ข้อมูลร่วมกัน โดยถือเสมือนว่าโปรแกรมคือข้อมูลชนิดหนึ่ง ซึ่งสามารถทำการประมวลผลข้อมูลอื่นได้

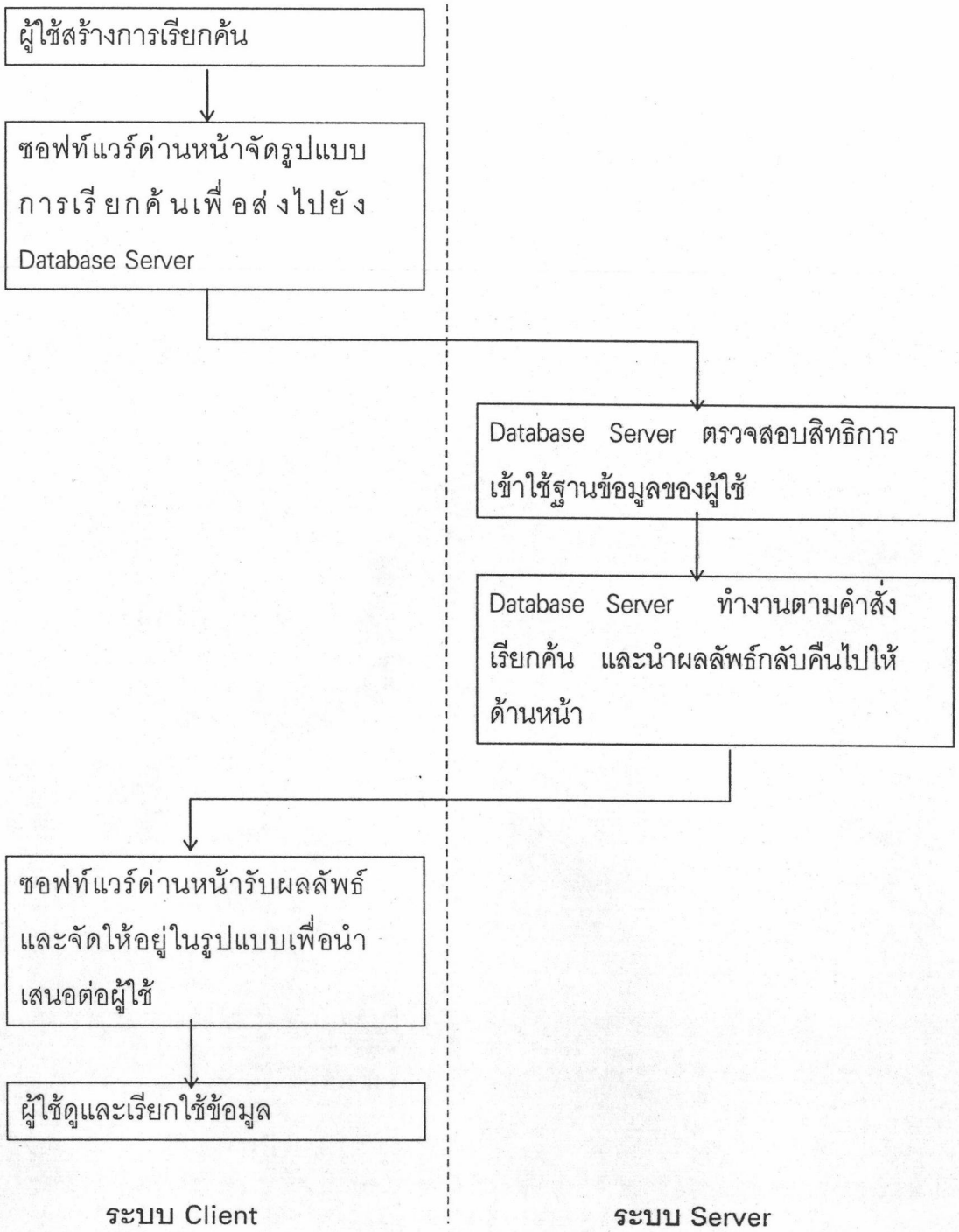
ระบบที่เน้นการใช้ทรัพยากรต่าง ๆ ร่วมกันโดยผู้ใช้ทรัพยากรนั้น ๆ ไม่มีความรู้สึกว่าตนเองใช้ทรัพยากรร่วมกับผู้อื่น คือระบบผู้ใช้บริการ & ผู้ให้บริการ (Client & Server)



รูปที่ 2.3 ตัวอย่างระบบ Client & Server

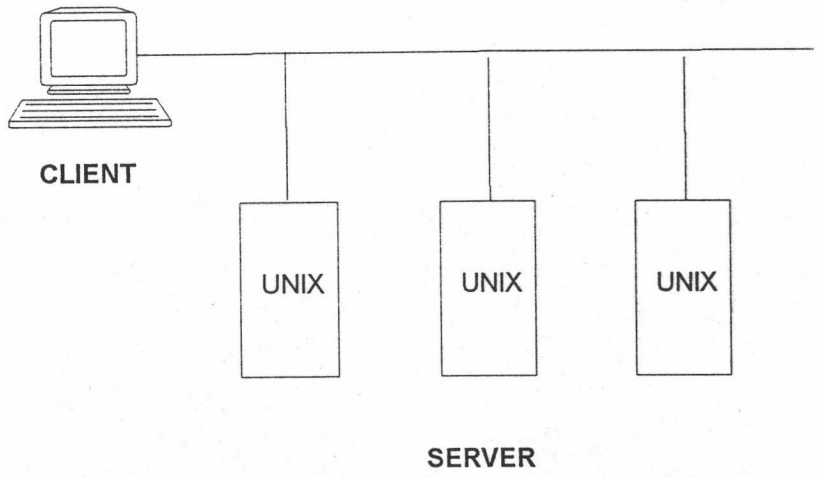
ระดับล่างสุดเป็นส่วนของฮาร์ดแวร์ ระดับกลางที่อยู่เหนือส่วนฮาร์ดแวร์ขึ้นไปจะเป็นชั้นการบริหารระบบ (System Services) ที่อาจประกอบด้วยโปรแกรมปฏิบัติการ โปรแกรมเครือข่าย โดยจะรวมถึงซอฟต์แวร์ต่างๆที่โปรแกรมประยุกต์ของชั้นที่เหนือกว่า ใช้เพื่อติดต่อไปควบคุมการทำงานของฮาร์ดแวร์ ระดับบนสุดเป็นระดับของโปรแกรมประยุกต์แยกเป็นส่วนของ Client ที่นิยมเรียกกันว่า Front - End Applications และส่วนของ Server เรียกกันว่า Back - End Database Engine

ขั้นตอนลำดับการเรียกค้นข้อมูลจากผู้ใช้ไปยังฐานข้อมูล แล้วนำข้อมูลกลับมายังผู้ใช้สามารถแสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 ขั้นตอนลำดับการเรียกค้นข้อมูลจากผู้ใช้ไปยัง Database Server และกลับมายังผู้ใช้

Client สามารถเลือกใช้ข้อมูลจาก Server ใด ๆ ในระบบได้ดังแสดงในรูปที่ 2.5



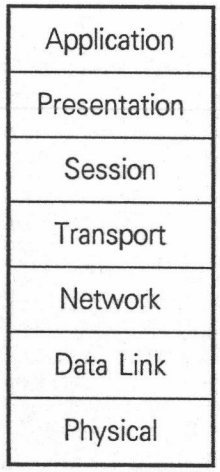
รูปที่ 2.5 ตัวอย่างระบบผู้ให้บริการและผู้ใช้บริการ

2.7 มาตรฐานของระบบเครือข่าย

เพื่อจะให้เกิดมาตรฐานเดียวกันบนระบบเครือข่าย องค์การกำหนดมาตรฐานระหว่างประเทศ ISO (International Standards Organization) จึงได้กำหนดมาตรฐานการสื่อสารข้อมูลแบบระบบเปิด OSI (Open System Interconnection) ขึ้น ทำให้เครื่องคอมพิวเตอร์ต่าง ๆ ที่เชื่อมต่อกันเป็นระบบเครือข่ายสามารถติดต่อสื่อสารกันได้ในรูปแบบเดียวกัน

หลักการในการติดต่อสื่อสาร ISO ได้แบ่งระดับการติดต่อออกเป็นชั้น ๆ ในแต่ละชั้นจะทำงานเพียงส่วนหนึ่งของหน้าที่ทั้งหมด ในการติดต่อกับคอมพิวเตอร์ระบบอื่นจะใช้หลักที่ว่าชั้นที่ต่ำทำการติดต่อที่ไม่สลับซับซ้อน ซึ่งชั้นเหล่านี้จะให้บริการกับชั้นที่สูงกว่า และขอบริการกับชั้นที่ต่ำกว่า หรืออีกนัยหนึ่งคือ การทำงานของระดับที่อยู่ด้านล่างจะเป็นตัว

สนับสนุนการทำงานในระดับสูงกว่า เมื่อมีการเปลี่ยนแปลงในชั้นใดชั้นหนึ่ง จะไม่จำเป็นต้องเปลี่ยนในชั้นอื่นด้วย ซึ่งชั้นต่าง ๆ เหล่านี้ ได้แสดงในรูปที่ 2.6



รูปที่ 2.6 OSI โมเดล

รูปที่ 2.6 แสดงถึง OSI โมเดล ประกอบด้วยชั้น (Layer) ต่าง ๆ จำนวน 7 ระดับ การสื่อสารในระดับต่าง ๆ จะต้องมีการควบคุม เพื่อทำให้ระบบการรับ-ส่งข้อมูลเป็นไปอย่างถูกต้องมีมาตรฐาน ซึ่งชั้นทั้ง 7 สามารถอธิบายโดยย่อได้ดังนี้

2.7.1 ระดับชั้น Physical Layer

เกี่ยวข้องกับการส่งข้อมูลในลักษณะบิต ผ่านสื่อการสื่อสารจริงที่ใช้เชื่อมต่อคอมพิวเตอร์ต่าง ๆ เข้าด้วยกัน เป็นการกำหนดมาตรฐานทางด้านฮาร์ดแวร์ที่ใช้ เช่น สื่อการสื่อสาร, interface card, ลักษณะของสัญญาณไฟฟ้า

2.7.2 ระดับชั้น Data Link Layer

เป็นการรวมข้อมูล หรือเกี่ยวข้องกับการส่งเฟรมของข้อมูล (dataframe) ไปยังระดับชั้น Physical Layer โดยใช้ส่วนควบคุมที่เรียกว่า แฟลก (Flag) ระบุจุดที่เริ่มต้น และจุดสุดท้ายของชุดข้อมูล หน้าที่แฟลกก็คือใช้ตรวจสอบความผิดพลาดระหว่างการส่งข้อมูล

2.7.3 ระดับ Network Layer

เป็นการทำแพ็กเกตสวิตชิง (packet switching) และการกำหนดวงจรเสมือน (virtual circuit) ซึ่งเกี่ยวข้องกับเส้นทางที่ข้อมูลจะต้องเดินทางไปเพื่อให้ถึงจุดหมายปลายทาง ใช้ในการเชื่อมต่อระบบคอมพิวเตอร์เข้าด้วยกัน ทำหน้าที่ขอการติดต่อ รักษาการติดต่อและยกเลิกการติดต่อ

2.7.4 ระดับ Transport Layer

เป็นการตรวจสอบความถูกต้องของข้อมูลระหว่างต้นทางและปลายทาง (end-to-end) แก้ไขข้อผิดพลาดหรือทำการกู้ (recovery) ข้อมูล รวมทั้งการทำมัลติเพล็กซ์หรือรวบรวมข้อมูลหลาย ๆ ข้อมูลเข้าด้วยกัน เพื่อส่งให้จุดรับต่าง ๆ

2.7.5 ระดับ Session Layer

เกี่ยวข้องกับการจัดระบบเครือข่าย (Network Management) เช่นการควบคุมตรวจสอบรหัสผ่าน การตรวจดูระบบเครือข่าย และใช้ในการจัดการติดต่อระหว่าง application process โดยกระบวนการขอการติดต่อ รักษาการติดต่อและยกเลิกการติดต่อระหว่าง application process ของแต่ละเครื่อง

2.7.6 ระดับ Presentation Layer

เกี่ยวข้องกับระบบรักษาความปลอดภัยของระบบเครือข่าย ใช้ในการแปลงข้อมูล และให้บริการทั่ว ๆ ไปในการติดต่อสื่อสาร เช่นการ encryption, compression

2.7.7 ระดับ Application Layer

เกี่ยวข้องกับโปรแกรมควบคุมระบบเครือข่าย การจัดการฐานข้อมูล และให้บริการแก่ผู้ใช้ทั่วไป เช่น transaction server, file transfer protocol, network management.

ในการติดต่อกันระหว่าง 2 ระบบ จะต้องมีระดับชั้นที่เหมือนกันและจำนวนที่เท่ากันทั้ง 2 ระบบ

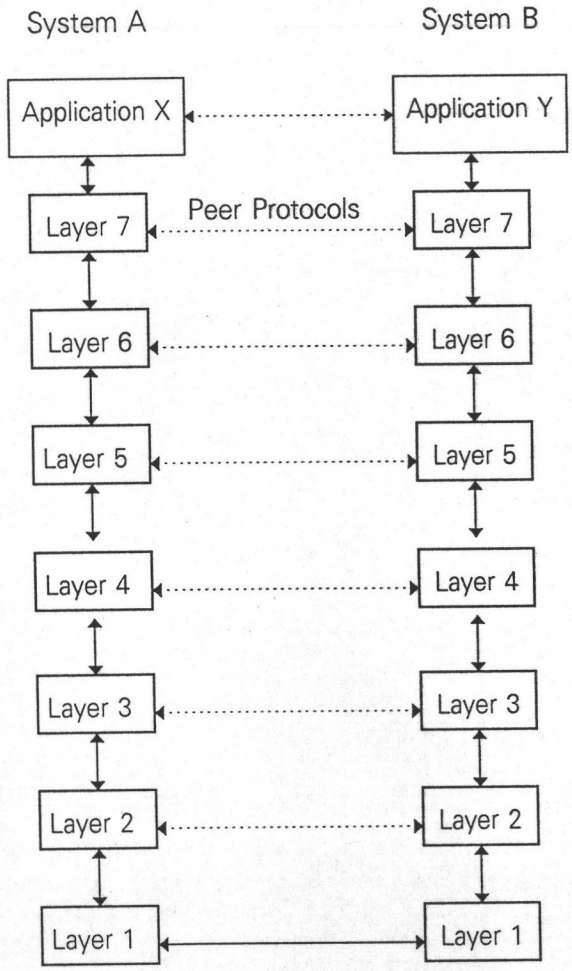
โปรแกรมหรือกระบวนการที่ทำหน้าที่ของแต่ละระดับจะเรียกว่า peer หรือ peer process การติดต่อสื่อสารจะสำเร็จก็เมื่อได้ peer ที่สอดคล้องกันของแต่ละเครื่องติดต่อกัน ตัวอย่างเช่น peer process ของระดับที่ 3 ในเครื่อง A ติดต่อกับ peer process ของระดับที่ 3 ในเครื่อง B ซึ่งลักษณะเช่นนี้เรียกว่า peer-to-peer แต่ในความเป็นจริงจะไม่มีข้อมูลจากระดับที่ n ของเครื่องหนึ่ง ส่งโดยตรงไปยังระดับที่ n ของอีกเครื่องหนึ่ง นอกจากที่ physical layer โดยปกติแล้วแต่ละระดับจะส่งข้อมูลและคำสั่งไปยังระดับที่อยู่ต่ำกว่าลงไป จนกระทั่งถึงระดับที่ต่ำสุด ซึ่งระดับที่ต่ำสุดนี้จะทำการติดต่อจริงกับระดับที่ต่ำสุดของอีกเครื่องหนึ่ง การติดต่อที่ระดับต่ำสุดนี้เรียกว่า physical communication ส่วนการติดต่อระหว่างระดับอื่น ๆ ที่สูงขึ้นไปเรียกว่า virtual communication แต่ละ peer layers ของเครื่องหนึ่งจะติดต่อกับ peer layers ของอีกเครื่องหนึ่ง ก็ต้องมีกฎหรือระเบียบแบบแผน กฎหรือระเบียบนี้คือโปรโตคอลซึ่งจะใช้กำหนด

- Syntax ซึ่งจะรวมถึงการจัดเรียงข้อมูล (format) และลักษณะของสัญญาณ (signal)
- Semantics ซึ่งจะรวมถึงข้อมูลที่ใช้ในการควบคุม (control) เพื่อความสอดคล้องในการทำงาน และการแก้ไขข้อผิดพลาด
- Timing ซึ่งจะรวมถึงความเร็วของการส่งผ่านข้อมูล และการจัดลำดับของข้อมูล (sequencing)

ระหว่างแต่ละระดับเช่นระหว่างระดับที่ 5 กับระดับที่ 6 จะมีตัวเชื่อมประสาน (interface) ซึ่งกำหนดลักษณะการบริการที่ระดับที่อยู่ต่ำกว่าบริการให้กับระดับที่อยู่สูงกว่า

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวเชื่อมประสานที่ดี จะทำให้การเปลี่ยนวิธีการปฏิบัติของระดับที่หนึ่งไปเป็นวิธีการอีกอย่างหนึ่งทำได้ง่าย ซึ่งจะนำไปถึงการที่ไม่ต้องเปลี่ยนวิธีการปฏิบัติของทุกระดับในทางปฏิบัติหรือไม่จำเป็นที่เครื่องทุก ๆ เครื่องจะต้องมีตัวเชื่อมประสานเหมือนกัน แต่ทุก ๆ เครื่องจะต้องมีระดับและโปรโตคอลที่เหมือนกัน



รูปที่ 2.7 การติดต่อกันระหว่าง Application

จากรูป 2.7 Application X ของ System A ต้องการติดต่อกับ Application Y ของ System B ดังนั้น Application X จะต้องติดต่อกับ layer 7 ของ System A ซึ่ง layer 7 ของ

System A นี้จะพยายามติดต่อกับระดับที่ 7 ของ System B โดยใช้ protocol ของระดับที่ 7 การติดต่อกันของระดับที่ 7 นี้ ต้องการบริการจากระดับที่ 6 ดังนั้นจึงทำให้ระดับที่ 6 ของทั้ง 2 ระบบติดต่อกันโดยใช้โปรโตคอลของระดับที่ 6 ทำอย่างนี้เรื่อยไปจนกระทั่งถึง physical layer ซึ่งข้อมูลจริง ๆ จะถูกส่งผ่านไปยัง System B

จากทั้งหมดนี้สรุปได้ว่า วิธีการของ OSI เป็นวิธีที่แก้ปัญหาการติดต่อสื่อสารระหว่างระบบคอมพิวเตอร์ต่าง ๆ ระบบคอมพิวเตอร์ 2 ระบบไม่ว่าจะแตกต่างกันอย่างไร ก็สามารถติดต่อกันได้ ถ้ามี

- ฟังก์ชันการติดต่อสื่อสาร (Communication functions) อย่างเดียวกัน
- ระดับที่แต่ละระดับจะต้องมีฟังก์ชันอย่างเดียวกัน แต่ไม่จำเป็นที่ฟังก์ชันเหล่านั้นจะมีวิธีการปฏิบัติ (implementation) ที่เหมือนกัน
- ระดับที่แต่ละระดับจะต้องมีโปรโตคอลอย่างเดียวกัน

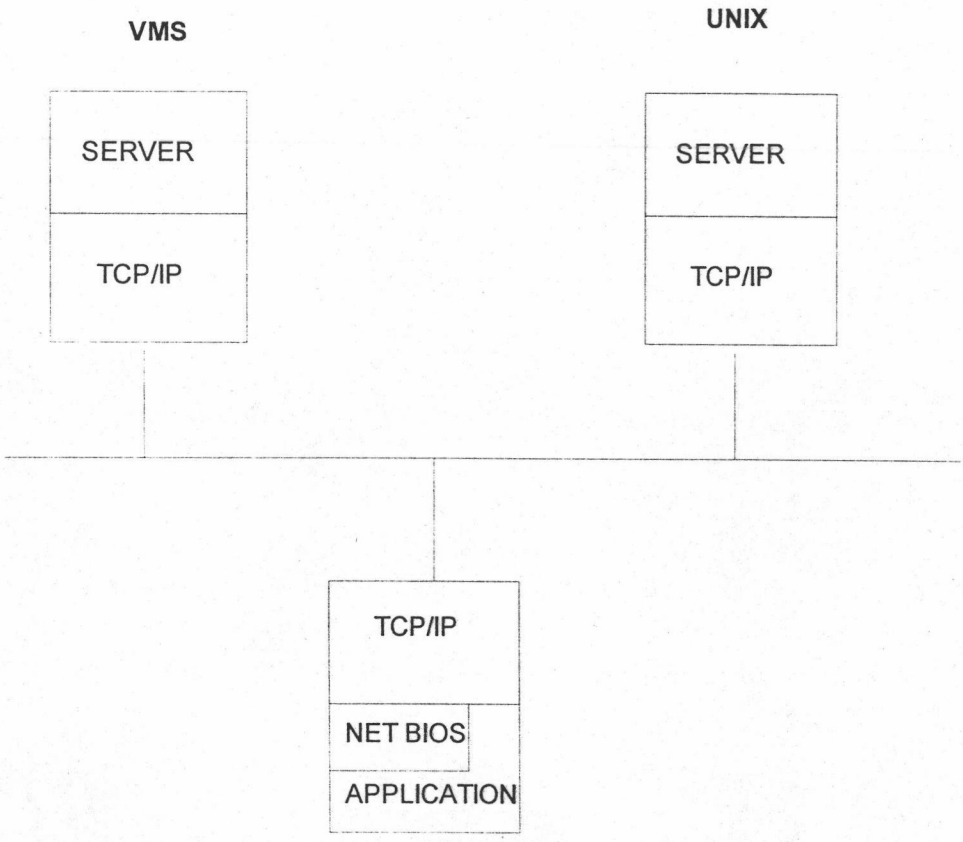
2.8 โปรโตคอลที่ใช้บนยูนิกซ์

การที่คอมพิวเตอร์บนระบบเครือข่ายจะมีการติดต่อสื่อสารกันอย่างมีประสิทธิภาพ โดยไม่เกิดความผิดพลาดหรือเหตุขัดข้องต่างๆ จำเป็นจะต้องมีมาตรฐานการเชื่อมโยงที่ถูกกำหนดขึ้น การใช้โปรโตคอลเป็นวิธีหนึ่งที่จะช่วยให้การติดต่อสื่อสารของระบบคอมพิวเตอร์ที่แตกต่างกัน เป็นไปได้อย่างราบรื่นและมีประสิทธิภาพดี

2.8.1 โปรโตคอล TCP/IP

TCP/IP เกิดขึ้นระหว่างปลายทศวรรษที่ 60 และทศวรรษที่ 70 จากการที่ Advance Reserch Projects Agency (ARPA) ของกระทรวงกลาโหมสหรัฐ (DoD) ได้สนับสนุนการพัฒนาระบบ เครือข่าย ชื่อ ARPANET เพื่อใช้ในการทหาร, มหาวิทยาลัย, และสำนักงานค้นคว้าวิจัย ที่ทำงานให้กับกระทรวงกลาโหมสหรัฐ ในปี 1980 (2523) ได้มีโปรโตคอลที่ถูก

กำหนดขึ้นให้ใช้กับ ARPANET ซึ่งมีชื่อว่า "DARPA Internet Protocol suite" แต่คนทั่วไปมักเรียกว่า TCP/IP ซึ่งย่อมาจาก Transmission Control Protocol / Intenet Protocol



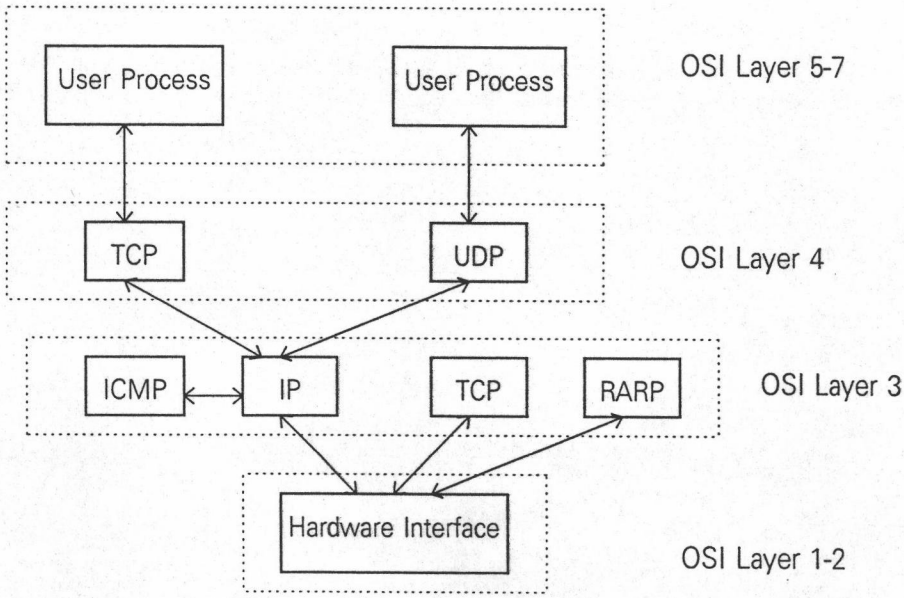
รูปที่ 2.8 การใช้ TCP/IP ในโมเดล Client & Server

ผลที่ได้จากการที่ TCP/IP ได้นำไปถึงการเชื่อมต่อเครือข่าย ต่าง ๆ เข้าด้วยกันเพื่อให้ดูเหมือนเป็นเครือข่ายใหญ่เครือข่ายเดียว เรียกว่า internet และเหตุที่ทำให้การใช้ TCP/IP เพิ่มขึ้นมากในทศวรรษที่ 80 นี้ ก็เพราะการที่ได้รวมใน BSD Unix ใน 1982

ข้อดีของ TCP/IP คือ

- ไม่ขึ้นกับผู้ผลิตรายใดรายหนึ่ง
- เป็นโปรโตคอลที่มีแพร่หลาย ตั้งแต่ PC จนถึง Mainframe
- สามารถใช้กับ Local Area Network (LAN) และ Wide Area Network (WAN)
- เป็นที่นิยมใช้แพร่หลาย ไม่เฉพาะแต่ใน DARPA หรือ DoD เท่านั้น

TCP/IP ประกอบไปด้วยโปรโตคอลต่าง ๆ ซึ่งจะแสดงในรูปที่ 2.9



รูปที่ 2.9 ภาพรวมของ TCP/IP

โปรโตคอลในภาพที่ 2.9 คือ

TCP Transmission Control Protocol เป็น connection oriented protocol ซึ่งวางใจได้ว่าถ้า TCP ยอมรับข้อมูลใด ข้อมูลนั้นจะถูกต้อง และเป็น full duplex

UDP User Datagram Protocol เป็น connectionless protocol ซึ่ง UDP เป็นโปรโตคอลที่ไว้ใจไม่ได้ เพราะ UDP ไม่ได้ประกันว่า ข้อมูล (datagrams) ที่ส่งไปจะถึงจุดหมายปลายทาง

ICMP Internet Control Message Protocol เป็นโปรโตคอลที่คอยดูแลเกี่ยวกับ error และ control information ระหว่าง gateways และ hosts

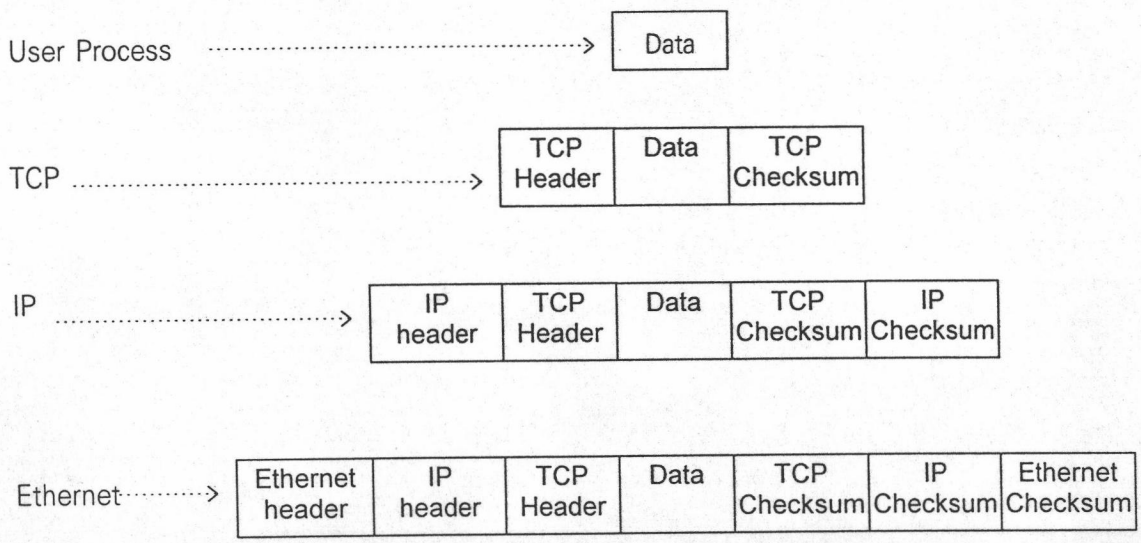
IP Internet Protocol เป็นโปรโตคอลที่ใช้ในการส่ง packet สำหรับ TCP, UDP และ ICMP

ARP Address Resolution Protocol เป็นโปรโตคอลที่ใช้ในการแปลงส่ง Internet address เข้ามาเป็น hardware address เช่น ethernet address

RARP Reverse Resolution Protocol เป็นโปรโตคอลที่ใช้ในการแปลงส่ง hardware address มาเป็น Internet address

2.8.3 ทีซีพีไอพีแพคเก็ต

การส่งข้อมูลในระบบเครือข่ายจะทำในรูป 'แพคเก็ต' ดังที่แสดงในรูป 2.10

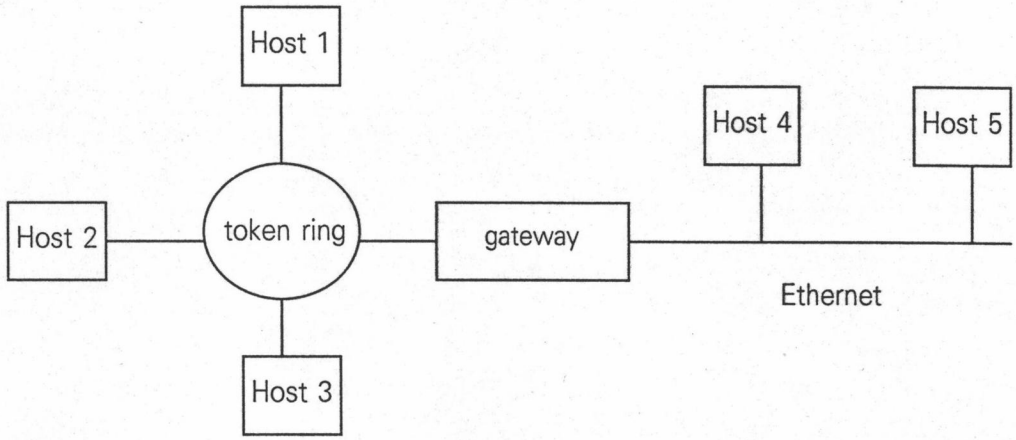


รูปที่ 2.10 TCP/IP packet

2.8.3 อินเตอร์เน็ต

Internet หรือ Internetwork คือการเชื่อมต่อเครือข่ายตั้งแต่ 2 เครือข่ายเข้าด้วยกัน เพื่อที่ว่าคอมพิวเตอร์บนเครือข่ายหนึ่งสามารถติดต่อกับคอมพิวเตอร์บนเครือข่ายอื่น ๆ จุดประสงค์ก็เพื่อที่จะซ่อนความแตกต่างของแต่ละเครือข่ายย่อย ๆ ให้ทำงานร่วมกันเหมือนกับเป็นหนึ่งเดียว

ตัวอย่างของอินเตอร์เน็ตในการเชื่อมต่อเครือข่าย 2 เครือข่ายเข้าด้วยกัน โดยใช้ gateway



รูปที่ 2.11 ตัวอย่างของ internet ที่ใช้ gateway

เรามีวิธีที่จะเชื่อมต่อระบบคอมพิวเตอร์เข้าด้วยกันหลายวิธีเช่น

1. Repeater ทำงานที่ OSI layer 1 โดยทำสำเนา (copy) สัญญาณไฟฟ้าจากเครือข่ายหนึ่งไปยังอีกส่วนหนึ่งของเครือข่ายหนึ่ง

2. Bridges ทำงานที่ OSI layer 2 โดยทำสำเนา (copy) เฟรม (frames) จากเครือข่ายหนึ่งไปยังอีกส่วนหนึ่งของเครือข่ายหนึ่ง

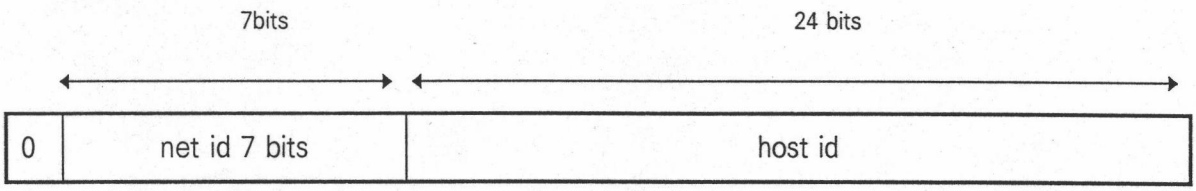
3. Routers ทำงานที่ OSI layer 3 โดยทำสำเนา (copy) กลุ่มข้อมูล (packets) จากเครือข่ายหนึ่งไปยังอีกส่วนหนึ่งของเครือข่ายหนึ่ง

4. Gateway เป็นคำทั่วไปที่ใช้เรียกสิ่งที่ใช้ในการเชื่อมต่อเครือข่าย ต่าง ๆ เข้าด้วยกันโดยปกติในวงการ TCP/IP gateway จะหมายถึง router

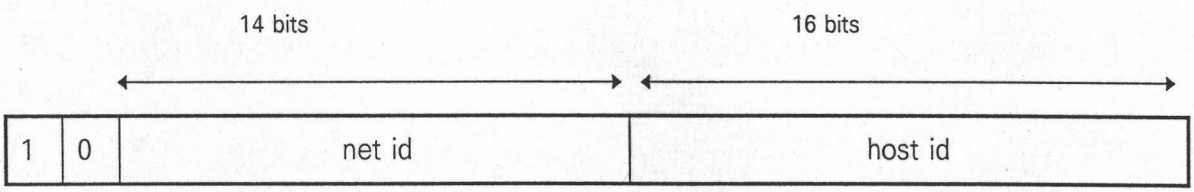
2.8.5 เลขที่อยู่บนอินเตอร์เน็ต (Internet Address)

เลขที่อยู่บนอินเตอร์เน็ตจะใช้ทั้งหมด 32 บิต ซึ่งภายใน 32 บิตประกอบด้วย network ID และ host ID โดยที่ทุกคอมพิวเตอร์เครือข่ายในอินเตอร์เน็ตจะต้องมีเลขที่อยู่ที่ไม่ซ้ำกันทั้ง 32 บิต และถูกจัดเป็น 4 รูปแบบดังรูป

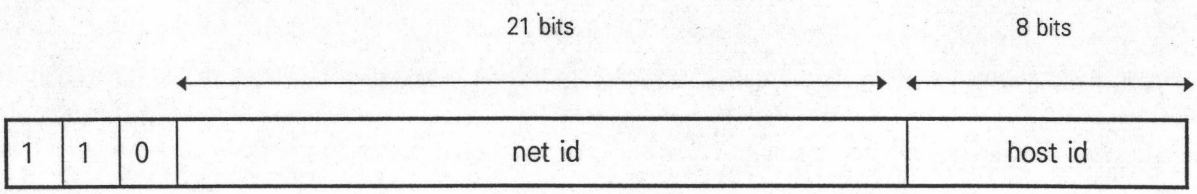
CLASS A



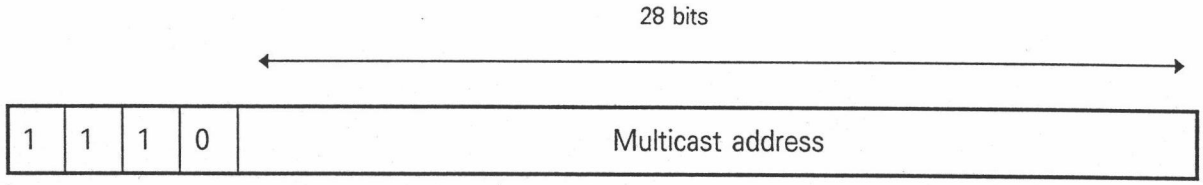
CLASS B



CLASS C



CLASS D



Class A address ถูกใช้กับเครือข่าย (network) ที่มีคอมพิวเตอร์แม่ข่าย (host) จำนวนมาก แต่มีเครือข่ายน้อยในขณะที่ Class C address จะมีเครือข่ายมากขึ้นแต่มีคอมพิวเตอร์แม่ข่ายน้อยลง และ Class D address ไม่ได้สนับสนุนโดย Network Information Center (NIC) ซึ่งเป็นองค์กรที่ใช้ออกเลขที่อยู่บนอินเตอร์เน็ต

ในเวลาที่แสดงค่าเลขที่อยู่บนอินเตอร์เน็ต จะจัดกลุ่มเป็นกลุ่มละ 8 บิต จำนวน 4 กลุ่มหรือ 4 octets ในแต่ละ octet ใช้เลขฐาน 10 ในการแทนค่าและแต่ละกลุ่มจะแยกออกจากกลุ่มอื่น ๆ โดยใช้ "." เช่นถ้ามีเลขที่อยู่ 00000001 11111111 00001010 00000010 สามารถเขียนเป็นเลขที่อยู่บนอินเตอร์เน็ตเป็น 1.255.10.2

เมื่อพิจารณา 8 บิตแรก จะเห็นว่า Class A จะมีค่าอยู่ระหว่าง 0 - 127, Class B มีค่าอยู่ระหว่าง 128 - 191 และ Class C มีค่าอยู่ระหว่าง 192 - 255 แต่ในความเป็นจริง จำนวน bit address ที่ใช้เจาะจงคอมพิวเตอร์แม่ข่ายตัวแรกและตัวสุดท้ายไม่สามารถใช้ได้ เช่น คอมพิวเตอร์แม่ข่าย Class C ควรจะมีเลขที่อยู่ตั้งแต่ 0 - 255 แต่จริง ๆ แล้วใช้ตั้งแต่ 1 ถึง 254 เท่านั้น

ถ้าพิจารณาจำนวนเครือข่ายในแต่ละ Class จะเห็นว่า Class A จะมีได้ 127 เครือข่าย Class B มีได้ 16,383 เครือข่าย Class C มีได้มากกว่า 2 ล้านเครือข่าย การจัดเส้นทาง (routing) ยิ่งสลับซับซ้อน โดยเฉพาะถ้าองค์กรมีจำนวนเครือข่ายมากขึ้น ตัวอย่างเช่น ถ้าใช้เลขที่อยู่บน Class C และมี 10 เครือข่ายซึ่งแต่ละเครือข่ายมีคอมพิวเตอร์แม่ข่าย 150 เครื่อง การเพิ่มคอมพิวเตอร์แม่ข่ายเข้าไป 1 เครื่องไม่ว่าบนเครือข่ายใด ก็จะต้องไปแก้ไขตารางเส้นทางของทุก gateway และถ้ามาดูจำนวนการจราจรในแต่ละเครือข่ายอาจพบว่าใน 150 คอมพิวเตอร์แม่ข่ายนี้ควรแบ่งเป็น 3 กลุ่ม กลุ่มละ 50 ตัวที่ชอบติดต่อกันอยู่เสมอ และเป็นส่วนน้อยที่มีการ

ติดต่อกับเครื่องนอกกลุ่ม แต่ถ้าเราจัดเครือข่ายนี้ใหม่เป็น 3 เครือข่าย จำนวนเครือข่ายก็จะมากขึ้น ทำให้การบริหารเครือข่ายทำได้ยากขึ้น จึงมีการแก้ไขปัญหานี้โดยใช้เครือข่ายย่อยแทน

เครือข่ายย่อยคือการแบ่งเครือข่าย TCP/IP ทางกายภาพหนึ่ง ๆ ลงไปเป็นเครือข่ายทางตรรกะ (logical network) ย่อย ซึ่งจะทำให้การจัดเส้นทางภายในเครือข่ายนั้นขึ้นอยู่กับเลขที่อยู่ของเครือข่ายและเลขที่อยู่ของเครือข่ายย่อย ส่วนการจัดเส้นทางนอกเครือข่ายยังคงขึ้นอยู่กับเลขที่อยู่ของเครือข่ายเพียงส่วนเดียวเหมือนเดิม การสร้างเลขที่อยู่ของเครือข่ายย่อยได้โดยการแบ่งฟิลด์ host id ในเลขที่อยู่บนอินเตอร์เน็ต ให้เป็นส่วนของ subnet id และ host id ตัวอย่างเช่น แทนที่จะใช้ Class C ให้ใช้ Class B แทน ดังนั้นจะมี 16 บิตสำหรับ host id ซึ่งถ้าแบ่ง 16 บิตนี้ออกไปเป็น 8 บิตแรกเป็น subnet id และ 8 บิตสุดท้ายเป็น host id ก็จะได้เครือข่ายย่อยถึง 256 เครือข่าย และมีคอมพิวเตอร์แม่ข่ายในแต่ละเครือข่ายย่อยได้ถึง 254 เครื่อง ดังนั้นถ้าเราใช้วิธีนี้ในการเพิ่มคอมพิวเตอร์แม่ข่าย ก็ไม่ต้องไปที่ gateway ทุกตัวเพื่อเพิ่มคอมพิวเตอร์แม่ข่ายใหม่นี้เข้าไป และเป็นข้อดีที่ทำให้แต่ละองค์กรมีเลขที่อยู่เครือข่ายเดียว ทำให้ง่ายต่อการติดต่อและจดจำ

2.9 BSD Socket

2.9.1 ความหมายและนิยาม

ซอคเก็ตเป็นกลุ่มของโปรแกรมชุดคำสั่งประจำย่อย (subroutine) ซึ่งกระทำด้าน IPC บนยูนิกซ์ ชุดคำสั่งนี้ใช้สำหรับสร้างเส้นทางการติดต่อระหว่างโปรแกรมบนเครื่องเดียวกัน และระหว่างโปรแกรมที่อยู่ต่างเครื่อง เราจะใช้ชุดคำสั่งซอคเก็ตเมื่อต้องการสร้างชุดคำสั่งที่ใช้ควบคุมการสื่อสารที่เป็นระดับต่ำ

2.9.2 ชนิดของซอคเก็ต

ชนิดของซอคเก็ต คือ

- Stream Socket (SOCK_STREAM)
- Datagram Socket (SOCK_DGRAM)
- Raw Socket (SOCK_RAW)
- Sequenced packet socket (SOCK_SEQPACKET)
- Reliably delivered message socket (SOCK_RDM)

ในที่นี้ขอกล่าวถึงซอคเก็ตที่สำคัญ 2 ชนิดคือ Stream Socket และ Datagram Socket

2.9.2.1 Stream Socket (SOCK_STREAM) เป็นการติดต่อสื่อสาร 2 ทางแบบโดยใช้โปรโตคอล TCP (The Transmission Control Protocol) ข้อมูลถูกส่งผ่านซอคเก็ตในรูปกระแสตัวอักษร และมีการจัดลำดับก่อนหลัง ตัวอย่างของ Stream Socket เช่น telnet, HTTP protocol เป็นต้น

2.9.2.2 Datagram Socket (SOCK_DGRAM) เป็นการสื่อสารแบบไร้การติดตั้งโดยใช้โปรโตคอล UDP (User Datagram Protocol) ข้อมูลถูกส่งผ่านซอคเก็ตในรูปกลุ่มตัวอักษร (packet) ที่เรียกว่า datagram และใช้การส่งข้อมูลโดยวิธี packet by packet ดังนั้นผู้รับไม่จำเป็นต้องได้รับข้อมูลเรียงตามลำดับที่ส่ง ตัวอย่างของ Datagram Socket เช่น ftp, tftp, bootp เป็นต้น

2.9.3 ระบบเลขที่อยู่ (address schemes)

ระบบเลขที่อยู่ที่ใช้กับซอคเก็ต มีดังนี้

- AF_UNIX Unix internal protocols
- AF_INET Internal Protocols

- AF_NS Xerox NS protocols
- AF_IMPLINK IMP link layer

ในที่นี้ขอกล่าวถึงระบบเลขที่อยู่ซอคเก็ตที่สำคัญ 2 ชนิดคือ

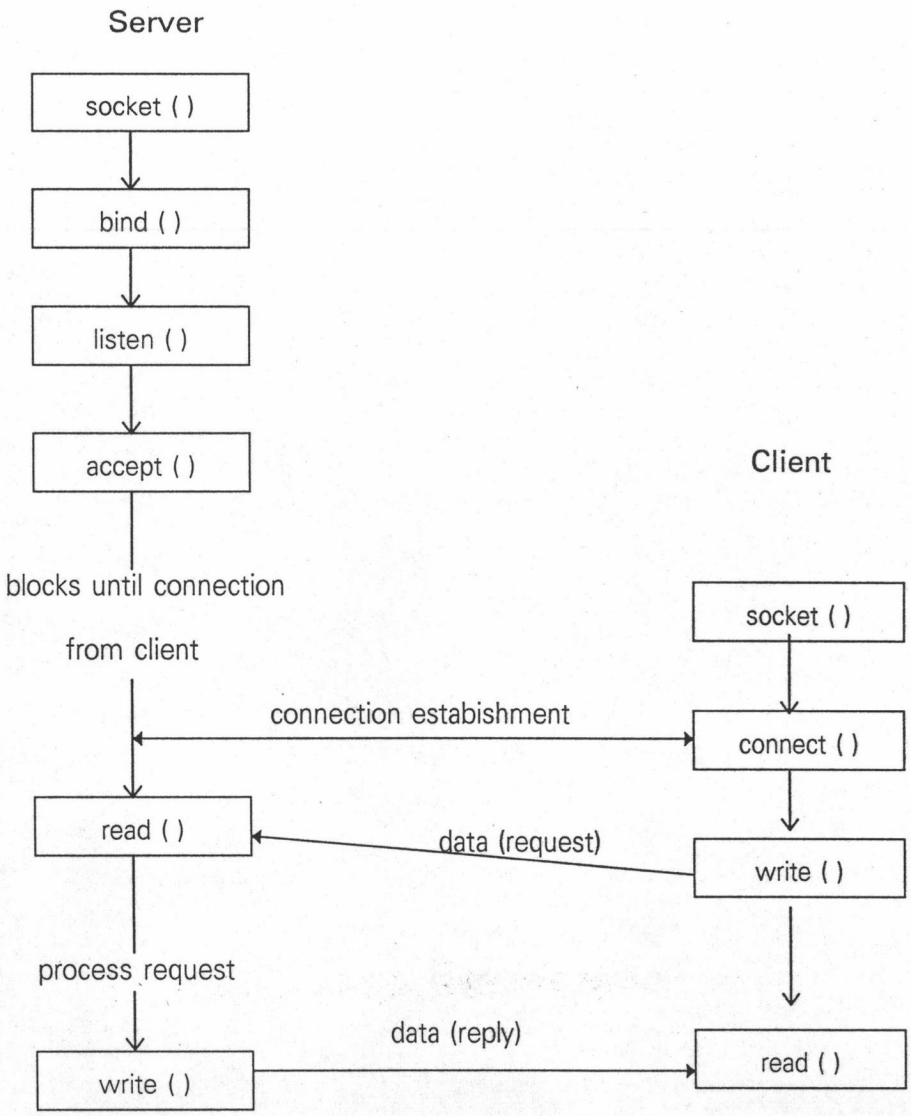
2.9.3.1 AF_UNIX เป็นระบบที่ใช้ทางเดินสารบบของยูนิกซ์ (Unix Pathname) ซึ่งมักใช้กับการติดต่อระหว่างกระบวนการภายในเครื่องเดียวกัน

2.9.3.2 AF_INET เป็นระบบที่ใช้ที่อยู่บนอินเตอร์เน็ตซึ่งเป็นจำนวน 4 ไบต์ และเขียนเป็นเลขฐานสิบคั่นด้วยจุด (เช่น 192.9.200.10) ซึ่งนอกจากจะเป็นเลขที่อยู่ของเครื่อง (machine address) แล้วยังมีหมายเลขทางเข้า/ออก (port number) ทำให้แต่ละเครื่องสามารถมี AF_INET ได้มากกว่าหนึ่ง

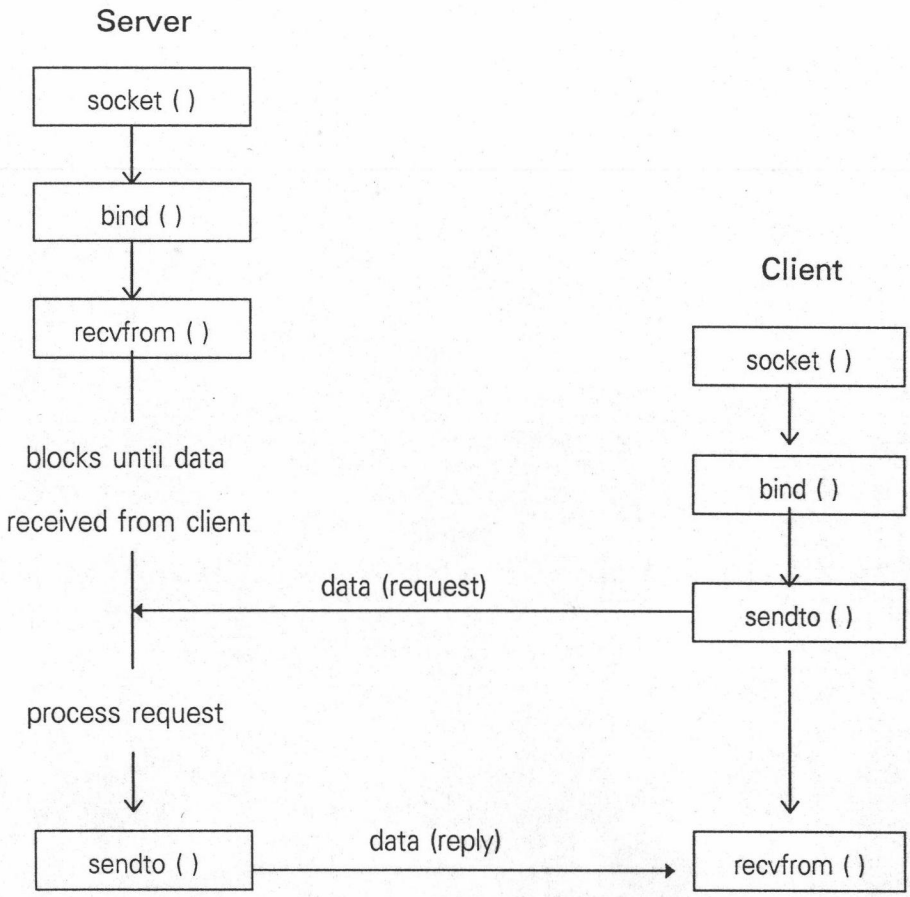
2.9.4 ชุดคำสั่งซอคเก็ต

โปรโตคอลการติดต่อระหว่างโปรแกรมโดยใช้ชุดคำสั่งซอคเก็ตมี 2 โปรโตคอล

- โปรโตคอลการติดต่อแบบติดตั้ง (Connection-oriented protocol)
- โปรโตคอลการติดต่อแบบไร้การติดตั้ง (Connectionless protocol)



รูปที่ 2.12 แผนภาพแสดงชุดคำสั่งซอคเก็ตสำหรับการติดต่อแบบติดตั้ง



รูปที่ 2.13 แผนภาพแสดงชุดคำสั่งซอคเก็ตสำหรับการติดต่อแบบไร้การติดตั้ง

ชุดคำสั่งซอคเก็ตที่สำคัญคือ

2.9.4.1 socket () ทำหน้าที่กำหนดโปรโตคอลและพารามิเตอร์ที่ใช้ในการติดต่อบนเครือข่าย โดยเป็น system call แรกที่ใช้ในกระบวนการติดต่อ

2.9.4.2 bind () ทำหน้าที่ตั้งชื่อให้ซอคเก็ตและกำหนดช่องทางเข้า/ออกของข้อมูลบนเครื่องคอมพิวเตอร์

2.9.4.3 connect () เป็นกระบวนการของเครื่องผู้ขอใช้บริการ (Client) ที่ทำหน้าที่ติดต่อกับผู้ให้บริการผ่านซอคเก็ต

2.9.4.4 listen () เป็นคำสั่งที่ใช้ในการติดต่อแบบติดตั้งบนเครื่องผู้ให้บริการ (Server) ที่บอกว่าพร้อมรับการติดต่อ

2.9.4.5 accept () เป็นคำสั่งที่ใช้ในการติดต่อแบบติดตั้งบนเครื่องผู้ให้บริการ (Server) ที่นำคำร้องของผู้ใช้บริการเข้าสู่คิวแรกแล้วสร้างซอคเก็ตใหม่ที่มีคุณสมบัติเหมือนเดิม

2.9.4.6 read () เป็นคำสั่งที่ใช้ในการติดต่อแบบติดตั้งเพื่อการอ่านข้อมูล หรือเป็นการนำข้อมูลเข้า

2.9.5.7 write () เป็นคำสั่งที่ใช้ในการติดต่อแบบติดตั้งเพื่อการเขียนข้อมูลหรือเป็นการส่งข้อมูลออก

2.9.5.8 send () และ sendto () เป็นคำสั่งที่ใช้ในการติดต่อแบบไร้การติดตั้งใช้เหมือน write ()

2.9.5.9 recv () และ recvfrom () เป็นคำสั่งที่ใช้ในการติดต่อแบบติดตั้งใช้เหมือน read ()

2.9.5.10 close () และ shutdown () เป็นคำสั่งที่ใช้ในการปิดซอคเก็ต

ท๒๒/๒๕๔๓