การพัฒนาโปรแกรมเชื่อมต่อฐานข้อมูลกับ LDAP สำหรับระบบบริหารผู้ใช้งาน

Software Development of Database-LDAP Gateway for User Management System

โดย

นายจิรวัฒน์ สันติมิตร

รหัส 44067080

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาโปรแกรมเชื่อมต่อฐานข้อมูลกับ LDAP สำหรับระบบบริหารผู้ใช้งาน
นักศึกษา	นายจิรวัฒน์ สันติมิตร
อาจารย์ที่ปรึกษา	อาจารย์อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

ในปัจจุบันระบบการใช้งานที่มีการแยกเก็บบัญชีผู้ใช้งานออกจากกันของแต่ระบบ ทำให้การใช้งานเกิดความซ้ำซ้อนของผู้ใช้งาน เราสามารถแก้ปัญหานี้โดยการทำข้อมูลต่างๆ มาเก็บและให้บริการจากจุดเดียว อาศัยการพัฒนาระบบ Directory Service โครงการพัฒนาโปรแกรมเพื่อโอนย้ายข้อมูลบัญชีผู้ใช้งานระหว่างฐานข้อมูลกับ Directory Service โดยใช้งานผ่านระบบ Web-base เพื่อให้สามารถใช้ข้อมูลร่วมกันได้ โดยระบบสามารถที่จะกำหนดลักษณะการย้ายของข้อมูล ให้มีความยืดหยุ่นในการใช้งาน และง่ายสะดวกต่อการใช้งาน รวมทั้งมีการสรุปผลการทำงานของโปรแกรม

การศึกษาและพัฒนาระบบโครงงานนี้โดยจัดทำระบบจำลอง เพื่อจัดการข้อมูลของบัญชีผู้ใช้งาน ซึ่งระบบจำลองมีลักษณะการทำงานอยู่บนพื้นฐานของ Client/Server โดยทำการติดตั้ง Oracle บน Window 2000 และติดตั้ง LDAP Server ขึ้นบน Solaris X86 ส่วนโปรแกรมที่พัฒนาเขียนโดยภาษา PHP ซึ่งเป็นเทคโนโลยีบน Web-Base ทำหน้าที่เป็นตัวกลางในการโอนย้ายข้อมูล โดยการดึงข้อมูลเฉพาะส่วนที่ต้องการจากฐานข้อมูล และทำการดัดแปลง หรือการกำหนดค่าขึ้นมาใหม่เพื่อให้มีความสมบูรณ์ แล้วจึงนำไปเก็บที่ปลายทางที่ระบุ เป็นการทำงานของระบบ ซึ่งสามารถนำไปประยุกต์ใช้ในปฏิบัติได้จริงในระบบที่ใช้อยู่ในปัจจุบัน เพื่อเสริมการปฏิบัติงานที่ทำอยู่ให้มีความสะดวก ที่จะทำให้เกิดประสิทธิผลมากขึ้นในการจัดการเกี่ยวกับข้อมูลบัญชีผู้ใช้งานระบบ เพื่อการใช้งานต่อไปในอนาคต

Title	Software Development of Database-LDAP Gateway for User Management System
Student	Mr. Chirawat Santimitra
Advisor	Mr. Akharin Khunkitti
Level of Study	Master of Science in information Technology
Major	Information Science
Academic Year	2002

ABSTRACT

In present, the organization had developed more and more information system. Each system has own user account separately. It increases cost and complexity. The Directory Service can help us to solve this problem. This project will implement and design the system that can access to database system and Directory Server for transferring user account data between them by using Web-based. Additionally, the software has flexibility and ease to use.

This software development project had made a prototype system for user account management. The system environment consist LDAP Server, Oracle and PHP web application on Window 2000 and Solaris X86. The PHP web application is the gateway for transfer information between database and LDAP Directory. The software creates the data from database and transfers to LDAP Server. The project can help us reduce deployment risk and have accommodation in the user account management.

กิตติกรรมประกาศ

ผู้เขียนต้องขอขอบคุณบุคคลดังต่อไปนี้ ที่มีส่วนสนับสนุนการพัฒนาโครงงานที่ได้จัดทำขึ้นนี้ให้สำเร็จลุล่วงตามวัตถุประสงค์ที่ต้องการ ได้แก่

1. บิดา มารดา ที่ค่อยดูแลเอาใจใส่ พี่น้องและเพื่อนๆ ที่เป็นกำลังใจให้ข้าพเจ้าเสมอ

2. อาจารย์อัครินทร์ คุณกิตติ อาจารย์ที่ปรึกษาโครงงาน ผู้ให้คำปรึกษาและแนะแนวทาง รวมถึงอาจารย์ในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ที่ได้มอบความรู้ทางวิชาการเพื่อนำมาใช้ในการพัฒนาระบบงาน

3. เจ้าหน้าที่ของสำนักวิจัยและบริการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ที่ให้ความร่วมมือให้การให้ข้อมูล

จิรวัฒน์ สันติมิตร

ผู้จัดทำ

สารบัญ

สารบัญตาราง

หน้า

ตารางที่

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในการศึกษาการทำงานของสำนักวิจัยและบริการคอมพิวเตอร์ ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นหน่วยงานมีหน้าที่ดูแลระบบสารสนเทศ และให้บริการระบบคอมพิวเตอร์ แก่นักศึกษาและบุคลากรของสถาบัน ซึ่งปัจจุบันสถาบันฯ มีการสร้างและเก็บข้อมูลของบัญชีผู้ใช้งานของนักศึกษา และบุคลากรแยกอิสระจากกัน ตามระบบงานต่างๆ จึงทำให้เกิดความซ้ำซ้อน และความแตกต่างของข้อมูลเกิดขึ้น

จากปัญหาที่เกิดขึ้น จึงมีแนวคิดที่จะพัฒนา ระบบ Directory Service ซึ่งเป็นระบบฐานข้อมูลอเนกประสงค์ ที่มีความสามารถในการค้นหาข้อมูล ได้อย่างรวดเร็ว และมีความยืดหยุ่นในการนำไปประยุกต์ใช้งานได้หลาย ซึ่งเราสามารถนำระบบ Directory Service มาประยุกต์ใช้งานเพื่อเก็บข้อมูลของบัญชีผู้ใช้งานมาบริการจากจุดเดียว เพื่อลดความซ้ำซ้อนในการเก็บข้อมูลผู้ใช้งานระบบ แต่ยังมีปัญหาในการโอนถ่ายข้อมูลจากระบบเดิมมายังระบบใหม่ เพื่อให้เกิดความสะดวกในการโอนถ่ายข้อมูลจากระบบเดิมมายังระบบใหม่ จึงได้เกิดโครงงานการพัฒนาโปรแกรมเพื่อเชื่อมต่อฐานข้อมูลกับ Directory Server โดยมุ่งเน้นในส่วนการเก็บข้อมูลของบัญชีผู้ใช้งาน เพื่อเป็นตัวกลางระหว่างระบบฐานข้อมูลเดิมกับระบบ Directory Service

1.2 วัตถุประสงค์ของการทำงาน

- เพื่อสร้างระบบบริการบัญชีผู้ใช้งาน และมีการบริหารการใช้งานจากศูนย์กลาง เพื่อเพิ่มประสิทธิภาพในการจัดการ และลดความซ้ำซ้อนของการเก็บข้อมูลผู้ใช้งานไว้ในหลายที่
- เพื่อสร้างระบบบริการบัญชีผู้ใช้งาน ที่สามารถใช้งานร่วมกับระบบที่มีอยู่ในปัจจุบัน และสามารถใช้งานร่วมกับระบบที่จะเกิดขึ้นในอนาคต
- เพื่อสร้างระบบบริการบัญชีผู้ใช้งาน ที่ความรวดเร็ว ความสะดวก ความปลอดภัย มีมาตรฐาน และมีความถูกต้อง เพื่อเพิ่มความน่าเชื่อถือในด้านการทำงานของระบบ
- เพื่อให้การพัฒนาโปรแกรมในการให้บริการผู้ใช้งานเป็นไปในแนวทางเดียวกันและพัฒนาได้อย่างรวดเร็วและมีประสิทธิ์ภาพรวมทั้งสามารถรองรับปริมาณงานที่เพิ่มขึ้นในอนาคต

- เพื่อย้ายระบบบริการบัญชีผู้ใช้งานจากระบบเดิม ไปสู่ระบบบริการบัญชีผู้ใช้งานจากระบบใหม่ได้โดยสะดวกตามความต้องการ

1.3 ขอบเขตของการทำงาน

- ศึกษาการใช้งาน Directory Server ของ OpenLDAP เพื่อใช้ในการเก็บข้อมูลบัญชีผู้ใช้งาน
- ศึกษาระบบ Account บน Unix เพื่อให้ใช้งานร่วมกับข้อมูลบัญชีผู้ใช้งานที่เก็บอยู่บน Directory Service
- พัฒนาระบบ Directory Service ที่สามารถรองรับข้อมูลของบัญชีผู้ใช้
- พัฒนาโปรแกรมที่สามารถโอนย้ายข้อมูลบัญชีผู้ใช้จากฐานข้อมูล Oracle ไปยัง Directory
- พัฒนาโปรแกรมที่สามารถปรับปรุงข้อมูลบนฐานข้อมูล Oracle จากข้อมูลบน Directory Service

1.4 ขั้นตอนการศึกษาและพัฒนาระบบ

- ศึกษาระบบงานในปัจจุบัน ลักษณะการทำงานของระบบลักษณะโครงสร้างข้อมูลที่ใช้ในระบบ และความเป็นไปได้ในการพัฒนาระบบและกำหนดขอบเขตในการพัฒนาระบบ ซึ่งใช้วิธีซักถามจากเจ้าหน้าที่มีหน้าที่รับผิดชอบในระบบงานสารสนเทศในสถาบันฯ
- ความต้องการของผู้ใช้งานในระบบ โดยค้นหาปัญหาที่เกิดจากการใช้งาน โดยใช้ข้อมูลที่ได้จากการศึกษาระบบงาน และมองถึงปัญหาที่อาจเกิดขึ้นในอนาคต
- คุณสมบัติของระบบ ทำการกำหนดความสามารถของระบบที่จะทำการออกแบบและพัฒนาเพื่อให้สามารถรองรับกับความต้องการของผู้ใช้งานระบบ
- ทำการศึกษาเลือกเครื่องมือที่จะใช้ในการพัฒนา โครงการ โดยพิจารณาตามความเหมาะสมในการใช้งาน
- ทำการพัฒนาโปรแกรม และทำการทดสอบโปรแกรม เพื่อให้โปรแกรม ทำงานได้อย่างถูกต้องและตรงตามความต้องการ

1.5 รายละเอียดของแต่ละบท

สำหรับเนื้อหาของเอกสารประกอบการพัฒนาโปรแกรมฉบับนี้ ได้มีการแบ่งเนื้อหาออกเป็นบทต่าง ๆ ดังต่อไปนี้

บทที่ 1 คือบทนี้จะกล่าวถึงความเป็นมา วัตถุประสงค์และความสำคัญของการพัฒนาระบบเพื่อให้ทราบเป้าหมายของการทำงาน และแนวทางให้การปฏิบัติ

บทที่ 2 จะกล่าวถึง Directory Service และระบบ Account บนระบบ Unix ซึ่งประกอบด้วยโครงสร้างการเก็บข้อมูล รายละเอียด คำนิยามและความหมายต่าง ๆ ที่ควรทราบ รวมถึงวิธีการทำงาน

บทที่ 3 จะเป็นการวิเคราะห์และออกแบบระบบ ซึ่งจะกล่าวถึงปัญหาของระบบปัจจุบัน รวมถึงการออกแบบโครงสร้างการทำงาน

บทที่ 4 จะเป็นขั้นตอนการพัฒนาโปรแกรม วิธีการให้การพัฒนาระบบ

บทที่ 5 เป็นการสรุปผลการทดลองการใช้งาน และขอเสนอแนะเพื่อเป็นประโยชน์แก่ผู้ที่จะนำไปใช้งาน หรือ พัฒนาโปรแกรมต่อไป

บทที่ 2

Directory Service

2.1 ระบบ Directory Service

การบริการไดเร็กทอรี เป็นระบบที่ให้การบริการสองลักษณะ คือ "User-Friendly naming" หรือ การตั้งชื่อที่สามารถเข้าใจได้ง่ายเพื่ออ้างอิงถึงสิ่งที่ต้องการ และ "Name-to-Address Mapping" หรือ การเชื่อมโยงชื่อของ Object กับตำแหน่งของ Object นั้น

ในหัวข้อต่อไปจะกล่าวถึงลักษณะโครงสร้างของไดเร็กทอรี องค์ประกอบและหน้าที่ของระบบ ลักษณะในการติดต่อ การควบคุมการเข้าถึง การทำสำเนาของไดเร็กทอรี และสุดท้ายจะเป็นสรุปบทความในเอกสารฉบับนี้

2.1.1 โครงสร้างของ Directory

ไดเร็กทอรี คือกลุ่มของระบบเปิดที่เก็บรวบรวมฐานข้อมูลเชิงตรรกะที่เป็นข้อมูลด้านต่างๆ ของสิ่งที่เป็นวัตถุหรือ Object ผู้ใช้งานหมายถึงคนหรือโปรแกรมคอมพิวเตอร์ จะสามารถ อ่านและทำการแก้ไขข้อมูลทั้งหมด หรือบางส่วนของข้อมูล ไดเร็กทอรีจะเก็บรายละเอียดของข้อมูลต่างๆ ในส่วนที่เรียกว่า Directory Information Base โดย DIB จะถูกจัดเก็บในรูปแบบของ Tree เรียกว่า Directory Information Tree (DIT)

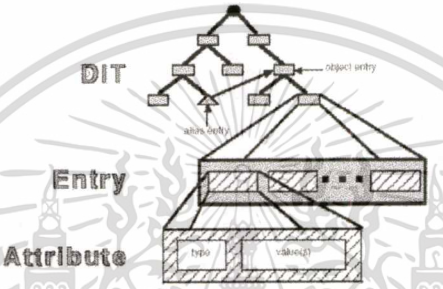
2.1.1.1 The Directory Information Base (DIB)

DIB เป็นส่วนที่ถูกสร้างขึ้นมาเพื่อเก็บข้อมูลต่างๆ ของ Object ที่เราสนใจ ซึ่งจะประกอบไปด้วย ส่วนที่เรียกว่า Entry ซึ่งแต่ละ Entry จะทำหน้าที่เก็บข้อมูลของ Object ได้ Object หนึ่ง โดยในแต่ละ Entry จะประกอบไปด้วยส่วนที่เรียกว่า Attributes ซึ่งแต่ละ Attribute อาจมีข้อมูลเก็บอยู่เพียงหนึ่งค่า หรืออาจจะมีการเก็บค่าหลายค่าไว้ใน Attribute ก็ได้ ชนิดของ Attribute จะขึ้นอยู่กับ Class ของ Object ที่ Entry นั้นๆ ได้ถูกกำหนดขึ้นมา

ไดเร็กทอรีจะทำการกำหนดกฎขึ้นมาเพื่อ ให้มั่นใจว่า DIB จะยังคงอยู่ในรูปแบบที่ดี แม้ว่าจะมีการแก้ไขข้อมูลก็ตาม กฎนี้เราเรียกว่า Directory Schema ซึ่งจะมีหน้าที่ป้องกันไม่ให้ Entry ต่างๆ มีค่าของ Attribute ผิดไปจากที่ได้กำหนดไว้ใน Class

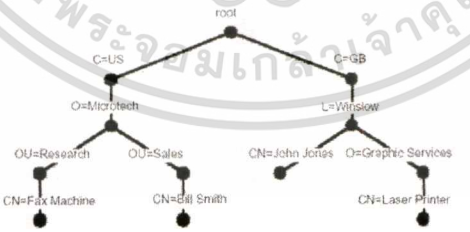
2.1.1.2 The Directory Information Tree (DIT)

แต่ละ Entry ของ DIB จะถูกแสดงในรูปแบบของ Tree เรียกว่า Directory Information Tree (DIT) ในส่วน Entry ที่อยู่สูงใน Tree มักจะถูกใช้ในการแสดง Object เช่น ประเทศ หรือ องค์กร ในขณะที่ Entry ที่อยู่ด้านล่างมักใช้แทนคน หรือการทำงานต่างๆ



รูปที่ 2.1 แสดงลักษณะโครงสร้าง Tree และ Entry

ในทุกๆ Entry จะมีชื่อที่ต่างกันเพื่อใช้ในการแบ่งแยก โดยชื่อที่ตั้งนี้จะไม่ซ้ำกันและ เป็นชื่อที่สื่อความหมายชัดเจน คุณสมบัติต่างๆ ของชื่อจะเกิดมาจากโครงสร้างของ Tree

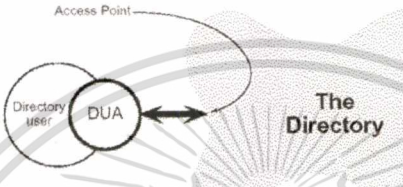


รูปที่ 2.2 แสดงตัวอย่าง DIT ของไดเร็กทอรี

บาง Entry ที่อยู่ในส่วนปลายสุดอาจจะเป็นสิ่งที่เรียกว่า Alias Entry ในขณะที่ Entry อื่นๆ เป็น Object Entry โดย Alias Entry จะทำการชี้ไปยัง Object Entry อีกที การที่มี Alias ทำให้เราสามารถมีชื่ออ้างอิง Object ได้มากกว่าหนึ่ง

2.1.2 องค์ประกอบและหน้าที่ของระบบ

ผู้ใช้งานหมายถึงคนหรือโปรแกรมคอมพิวเตอร์ จะสามารถ อ่านและทำการแก้ไขข้อมูลทั้งหมด หรือบางส่วนของข้อมูล ซึ่งลักษณะไดเร็กทอรีจะให้บริการในการอ่านข้อมูลได้ง่ายกว่าการแก้ไข แต่อย่างไรก็ตามต้องมีสิทธิ์ที่ยอมให้ สามารถทำสิ่งเหล่านี้ได้ ผู้ใช้งานแต่ละคน จะสามารถเข้าถึงไดเร็กทอรี โดยผ่านทาง Directory User Agent (DUA) ซึ่งแสดงในรูปที่ 2.3



รูปที่ 2.3 แสดงการติดต่อกับไดเร็กทอรี

ไดเร็กทอรีจะเก็บรายละเอียดของข้อมูลต่างๆ ไว้ที่ DIB ซึ่งไดเร็กทอรีจะให้บริการอ่าน แก้ไข และบริการพื้นฐานพื้นฐานอื่นๆ ผ่านทาง DUA นอกจากนี้ไดเร็กทอรียังมีคุณสมบัติในการกระจายโดยทำงานร่วมกับ Directory System Agent (DSA) ซึ่งจะช่วยให้สามารถทำงานร่วมกันกับระบบมากกว่าหนึ่งระบบ

2.1.2.1 The Directory User Agent (DUA)

ส่วนใช้ติดต่อในการเข้าใช้ระบบหรือ DUA ซึ่งให้ความสามารถในการใช้งานที่เป็นมาตรฐานโดยสามารถรองรับผู้ใช้ที่ต้องการค้นหา ในไดเร็กทอรีในหลายที่เพื่อให้ได้ข้อมูลที่ต้องการ นอกจาก DUA จะมีมาตรฐานการทำงานที่สามารถให้บริการกับผู้ใช้หลายราย ยังสามารถมาตรฐานให้บริการกับโปรแกรมอื่นๆ เช่น web-server gateways, โปรแกรมใช้ e-mail เป็นต้น และสามารถให้บริการกับเครื่องคอมพิวเตอร์ได้หลากหลาย

2.1.2.2 The Directory System Agent (DSA)

DSA เป็นระบบฐานข้อมูลที่ใช้เก็บข้อมูลของไดเร็กทอรี โดยมีโครงเป็นลำดับชั้น การทำงานของ DSA จะมีบทบาทที่จะให้บริการในการเข้าถึง DIB ซึ่ง DSA จะทำงานประสานงานกันในระบบ ซึ่งจะมีการให้บริการในการค้นหา หรือดึงข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ แต่ไม่เหมาะกับลักษณะการเก็บข้อมูลที่มีการเปลี่ยนแปลงบ่อย

2.1.3 Directory Protocol

เป็นกฎในการสื่อสารกันระหว่าง DUA กับกลุ่มของ DSA เรียกว่า Directory Access Protocol (DAP) เป็นการทำงานของโปรโตคอลในชั้น Application Layer ที่มีการทำงานงานประสานงานกัน เพื่อติดต่อกับ DUA และ DSA ในระบบให้ทำงานร่วมกัน มีโปรโตคอลต่างๆ ดังนี้

- **The Directory Access Protocol (DAP)** กำหนดการแลกเปลี่ยนของคำขอและผลการทำงานระหว่าง DUA กับ DSA
- **The Directory System Protocol (DSP)** กำหนดการแลกเปลี่ยนของคำขอและผลการทำงานระหว่าง DSA
- **The Directory Information Shadowing Protocol (DISP)** กำหนดการแลกเปลี่ยนของข้อมูลการทำสำเนาข้อมูลระหว่าง DSA ซึ่งได้มีการสร้าง shadowing agreement
- **The Directory Operational Binding Management Protocol (DOP)** กำหนดการแลกเปลี่ยนของข้อมูลสิทธิ์การทำงานระหว่าง DSA เพื่อที่จะทำการเชื่อมต่อระหว่างกัน

โปรโตคอลแต่ละตัวจะมีประกอบด้วยส่วนย่อยๆ เพื่อประสานการทำงาน เช่น DAP ประกอบด้วย ส่วนการซักถามและแก้ไขไดเร็กทอรี ซึ่งจะทำงานต้องทำงานประสานกัน

2.1.4 บริการของไดเรกทอรี

การให้บริการต่างสามารถทำงานได้โดยเรียกผ่านทาง Directory User Agent (DUA) ซึ่งสามารถแบ่งเป็นกลุ่มของการให้บริการได้ดังนี้

2.1.4.1 Service Control

จุดประสงค์หลักของบริการนี้ คือเพื่อจำกัดการใช้งานทรัพยากรของผู้ขอใช้บริการ ไม่ว่าจะเป็นเวลาการใช้งาน หรือขนาดของผลลัพธ์ ที่เกิดจากการค้นหาข้อมูล ของเขตการค้นข้อมูล รวมทั้งจัดลำดับความสำคัญของการขอใช้บริการ

2.1.4.2 Security Parameters

ในการร้องขอใช้บริการ เราสามารถกำหนดให้ข้อมูลเหล่านั้นมีการการรักษาความปลอดภัยได้ โดยสามารถใช้สิ่งต่าง ๆ เหล่านี้เพื่อรักษาความปลอดภัยของข้อมูลได้ เช่น โดยการใช้ Digital Signature

2.1.4.3 Filters

ในการใช้บริการอาจมีการขอใช้บริการการกรองข้อมูล โดยสามารถระบุเงื่อนไขที่ต้องการ หรือลำดับของข้อมูลได้ เพื่อให้ค่าที่ส่งออกไปยังผู้ขอรับบริการ มีเฉพาะค่าที่ต้องการเท่านั้น

2.1.4.4 Read

การร้องขอเพื่อใช้บริการจะเป็นลักษณะการขอใช้บริการที่มุ่งไปยัง Entry ที่สนใจโดยค่าที่ถูกเก็บอยู่ในทุกๆ Attribute ที่เกี่ยวข้องจะถูกส่งออกมาเป็นผลลัพธ์

2.1.4.5 Compare

การร้องขอเพื่อใช้บริการจะเป็นลักษณะการขอใช้บริการที่มุ่งไปยัง Entry ที่สนใจ แล้วไดเร็กทอรีจะทำการตรวจสอบว่า ค่าที่ให้มามีค่าเหมือนกัน หรือ ต่างกับค่าที่บรรจุอยู่ในไดเร็กทอรี เราสามารถนำความสามารถของบริการนี้ไปใช้ในการทำตรวจสอบรหัสการใช้งานหรือ password checking ได้โดยเรากำหนดให้ไม่สามารถอ่านได้ แต่ให้สามารถทำการเปรียบเทียบได้

2.1.4.6 List

เมื่อถูกร้องขอเพื่อใช้บริการนี้ ไดเร็กทอรีจะทำการส่งค่าที่อยู่ภายใต้ส่วนที่เราสนใจ จากข้อมูลที่อยู่ใน DIT

2.1.4.7 Search

เมื่อถูกร้องขอเพื่อใช้บริการนี้ ไดเร็กทอรีจะส่งค่าของข้อมูลกลับมาโดยจะส่งค่าที่เกี่ยวข้องกับสิ่งที่เราค้นหา บางส่วนหรือทั้งหมดออกมา โดยสามารถกำหนดการกรองข้อมูลได้ โดยค่าข้อมูลที่สามารถค้นหาได้นั้นต้องมีสิทธิ์ที่ยอมให้อ่านได้

2.1.4.8 Abandon

การขอใช้บริการนี้ จะเป็นการบอกให้ไดเร็กทอรี ยกเลิกการทำงานที่กำลังทำอยู่ เนื่องจากเราไม่ต้องการข้อมูลในส่วนที่กำลังงานทำอยู่แล้ว

2.1.4.9 Add Entry

เป็นการขอให้ไดเร็กทอรีเพิ่ม Entry ที่ส่วนปลายของ DIT ซึ่งอาจเป็น Object Entry หรือ Alias Entry ก็ได้

2.1.4.10 Remove Entry

เป็นการขอให้ไดเร็กทอรีทำการลบ Entry ที่ต้องการที่ส่วนปลายออกของ DIT ออกไป

2.1.4.11 Modify Entry

การร้องขอเพื่อเปลี่ยนแปลง Entry ที่สนใจ โดยไดเร็กทอรีจะทำการเปลี่ยนแปลงที่ละอย่าง เพื่อให้เกิดผลตามที่ต้องการ และผลที่ก็ขึ้นก็จะต้องลักษณะตามกฎที่กำหนดใน Scheme การเปลี่ยนแปลงที่อนุญาต ได้แก่ การเพิ่ม การลบ และการแทนที่ Attribute หรือค่าที่เก็บอยู่ใน Attribute

2.1.4.12 Modify Distinguished Name

เป็นการร้องขอเพื่อทำการเปลี่ยนแปลงชื่อของ Entry หรือ เพื่อทำการย้าย Entry ไปยังตำแหน่งอื่นใน DIT

2.1.4.13 Errors

การให้บริการอาจมีปัญหาเกิดขึ้นได้เช่น ผู้ใช้งานมีการกำหนดค่าที่ผิดพลาดในกรณีนี้ข้อผิดพลาดจะถูกรายงาน ไปยังผู้ใช้งาน

2.1.4.14 Referrals

การบริการอาจมีปัญหาเกิดขึ้นในกรณีที่ DUA เชื่อมกับจุดบริการไม่เหมาะสมที่จะส่งผลลัพธ์ไปให้ เช่น ข้อมูลที่ต้องการอยู่ห่างจากจุดเชื่อมต่อของผู้ร้องขอ ในกรณีนี้ไดเร็กทอรีจะทำการส่ง Referral อ้างถึงจุดเชื่อมต่ออื่นไปให้ DUA ซึ่งเพื่อทำให้ผู้ร้องขอใช้บริการสามารถทำการขอรับบริการจากจุดอื่นที่สามารถบริการได้

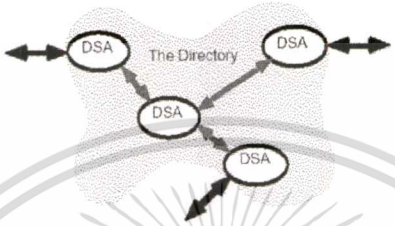
2.1.5 ลักษณะการทำงานแบบกระจายของไดเร็กทอรี

2.1.5.1 โครงสร้างการทำงาน

การทำงานของในไดเร็กทอรีมีส่วนที่เรียกว่า Directory System Agent (DSA) จะมีบทบาทที่จะให้บริการในการเข้าถึง DIB ซึ่ง DSA จะทำงานโดยใช้ข้อมูลที่เก็บอยู่ไว้ที่ตัวมันหรือส่งการร้องขอต่อไปยัง DSA ตัวอื่น หรืออีกทางเลือกหนึ่งคือ DSA จะส่งผู้ใช้ไปยัง DSA ที่ถูกต้องให้เชื่อมต่อกันโดยตรง

2.1.5.2 โครงสร้างการจัดการ

กลุ่มของ DSA และ DUA ที่ถูกจัดการในกลุ่มเดียวกันอาจจะกำหนดรูปแบบ Directory Management Domain (DMD) ซึ่งจะเป็นผู้ระบุลักษณะของไดเร็กทอรี (Directory Specification) ที่จะเป็นผู้ดูแลการติดต่อกันระหว่างส่วนต่างๆ ของกลุ่ม



รูปที่ 2.4 แสดงโครงสร้างการทำงานของไดเร็กทอรี

ลักษณะของไดเร็กทอรี (Directory Specification) อื่นๆ สามารถระบุลักษณะของคุณสมบัติของ DSA เพื่อให้กลุ่มของ DSA ที่อยู่ภายใน DMD มีคุณสมบัติเหมือนกัน

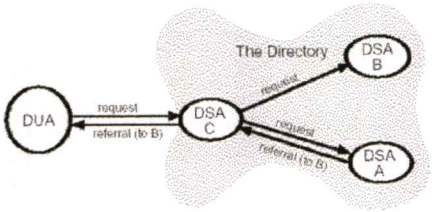
DMD แบ่งเป็น Administration DMD (ADDMD) และ Private DMD (PRDMD) ขึ้นอยู่กับลักษณะการให้บริการว่าเป็นแบบทั่วไปหรือแบบส่วนตัว

2.1.5.3 การทำงานของไดเร็กทอรี

DUA จะทำงานกับไดเร็กทอรีโดยการติดต่อกับ DSA โดย DUA ไม่จำเป็นต้องเชื่อมต่อกับ DSA อันใดอันหนึ่งโดยเฉพาะ DUA จะอาจติดต่อโดยตรงกับ DSA ใดใดก็ได้เพื่อทำการร้องขอ แต่มันก็เป็นไปได้ที่ DUA สามารถที่จะติดต่อกับไดเร็กทอรีผ่าน DSA เพียงตัวเดียว โดยไม่จำเป็นต้องติดต่อกับตัวอื่นเลย

DSA ที่ได้รับการร้องขอเพื่อใช้บริการจาก DUA แต่ไม่มีข้อมูลที่เหมาะสมอยู่ DSA ได้รับการเชื่อมต่อนั้นสามารถที่จะหาข้อมูลแทน DUA จาก DSA อื่นได้

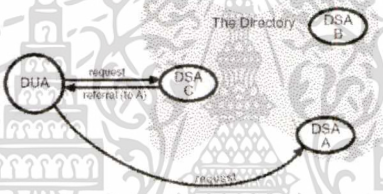
จากรูปที่ 2.5 เมื่อ DSA C ได้รับ referral การอ้างถึง DSA B จาก DSA A จะสามารถตอบสนองได้สองรูปแบบ คือ ส่งคำขอไปยัง DSA B เอง หรือ ส่งการอ้างถึงกลับไปยัง DUA ซึ่งเมื่อ DSA C ส่งการอ้างถึงกลับไป DUA แล้ว DSA C จะไม่ส่งคำขอไปยัง DSA B หรือกลับกัน เมื่อ DSA C ส่งคำขอไปยัง DSA B แล้ว DSA C จะไม่ส่งการอ้างถึงกลับไปยัง DUA



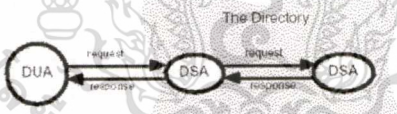
รูปที่ 2.5 แสดงลักษณะการส่งการอ้างถึง

จากรูปที่ 2.6 ได้แสดงลักษณะที่ DUA ได้รับการอ้างถึง DSA A จาก DSA C แล้ว DUA ได้ทำการติดต่อใหม่ไปยัง DSA A

จากรูปที่ 2.7 แสดงการเชื่อมต่อแบบสายเส้นเดียว โดย DUA จะส่งคำขอไปยัง DSA ที่ติดต่ออยู่ เมื่อ DSA ได้รับคำขอก็จะส่งต่อไปเรื่อยๆ จนได้ข้อมูลที่เหมาะสม

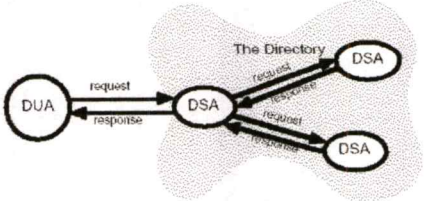


รูปที่ 2.6 แสดงการติดต่อเมื่อได้รับการอ้างถึง

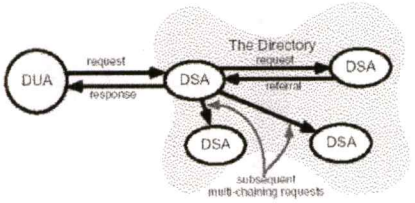


รูปที่ 2.7 แสดงการเชื่อมต่อแบบ Uni-chaining

จากรูปที่ 2.8 แสดงการเชื่อมต่อแบบสายหลายเส้น โดย DUA จะส่งคำขอไปยัง DSA ที่ติดต่ออยู่ เมื่อ DSA ได้รับคำขอก็จะส่งคำขอไปยังหลาย DSA ซึ่งแต่ละคำขอจะมีลักษณะเหมือนกัน



รูปที่ 2.8 แสดงการเชื่อมต่อแบบ Multi-chaining



รูปที่ 2.9 แสดงการเชื่อมต่อแบบผสมแบบ Hybrid

ในแต่ละวิธีการจะมีข้อดีของมัน เช่น ในรูปที่ 2.6 จะถูกใช้เมื่อไม่ต้องการให้มีภาระกับ DSA ภายใน และในสถานการณ์อื่นวิธี hybrid approach ซึ่งเป็นการรวมกันของการทำงานแบบต่างๆ ดังแสดงในรูปที่ 2.9

2.1.6 การควบคุมการใช้งาน

เพื่อที่จะเข้าถึงข้อมูลภายในไดเร็กทอรีต้องมีการกำหนดโดยนโยบายความปลอดภัยซึ่งถูกความคุมโดยผู้ดูแล โดยมีสองสิ่งที่ต้องคำนึงถึงคือ ขั้นตอนการพิจารณาเพื่อรับรอง และกลไกการทำงานรวมถึงกระบวนการตรวจสอบและการกระจาย

การกำหนดแบบแผนในการควบคุมการเข้าถึง ประกอบด้วย ขั้นตอนเพื่อพิจารณาข้อมูลการควบคุมการเข้าถึง ขั้นตอนพื่อควบคุมสิทธิ์การเข้าถึงตามข้อมูลการควบคุมการเข้าถึง และขั้นตอนเพื่อดูแลข้อมูลการควบคุมการเข้าถึง

การควบคุมสิทธิ์การเข้าถึงเป็นการรวมกันของการควบคุมการเข้าถึงข้อมูลของไดเร็กทอรีซึ่งสัมพันธ์กับโครงสร้างของ DIT ข้อมูลของผู้ใช้ ข้อมูลที่ใช้งาน และข้อมูลการควบคุมการเข้าถึง

การกำหนดแบบแผนการเข้าถึง สามารถใช้จากมาตรฐาน หรือจะสร้างแบบแผนใหม่ขึ้นได้ตามความต้องการ ซึ่งสิ่งที่ต้องคำนึงถึงของการควบคุมการเข้าถึง ได้แก่ ส่วนประกอบที่จะถูกเข้าถึง การทำงานที่ผู้ใช้ร้องขอ สิทธิ์ที่จำเป็นในการทำงาน และนโยบายที่ใช้ดูแลการเข้าถึง

2.2 Light Weight Directory Access Protocol (LDAP)

Lightweight Directory Service Protocol เป็นโปรโตคอลที่ทำหน้าที่ในการเข้าถึงข้อมูลใน Directory Server ในส่วนของ LDAP เองมีการทำงานอยู่บนพื้นฐานของ โปรโตคอล TCP/IP อีกทีหนึ่ง ในส่วนของข้อกำหนดต่าง ๆ ของ LDAP จะสามารถหาได้จาก RFC2251 "The Lightweight Directory Access Protocol (V3)"

LDAP เป็นมาตรฐานเพิ่มเติมจาก DAP โดยมีขนาดเล็กกว่า DAP แต่ยังคงมีประสิทธิภาพและงานต่อการใช้งาน ซึ่งใช้ในการติดต่อสื่อสารกันระหว่าง LDAP client และ LDAP server มาตรฐานจะช่วยให้ client และ server จากแต่ละแหล่งสามารถทำงานร่วมกันได้

การอธิบายมาตรฐานของ LDAP ได้มีการกำหนดรูปแบบเพื่อที่จะได้นำไปใช้กับ Directory และยังคงความยืดหยุ่นในการใช้งาน โดยมีรูปแบบดังนี้

- **LDAP information model** จะเป็นมาตรฐานในการกำหนด รูปแบบของข้อมูลที่สามารถที่จะจัดเก็บใน Directory Service ได้
- **LDAP naming model** จะเป็นมาตรฐานในการกำหนดรูปแบบของโครงสร้างของข้อมูลที่ถูกจัดเก็บอยู่ใน Directory Service รวมทั้งวิธีการอ้างถึงข้อมูล
- **LDAP functional model** เป็นการกำหนด รูปแบบการติดต่อและบริการต่างๆ ที่เราสามารถเรียกใช้งาน เพื่อทำการบริหารข้อมูลใน Directory Server
- **LDAP security model** เป็นการกำหนด รูปแบบในการป้องกันการเข้าถึงข้อมูลจากบุคคลที่ไม่มีสิทธ์ในการใช้งานข้อมูล

2.2.1 LDAP information model

ข้อมูลที่ถูกเก็บใน Directory Service จะประกอบไปด้วย ส่วนที่เรียกว่า entry โดยแต่ละ entry จะเก็บข้อมูลของสิ่งที่เราสนใจไว้ เช่น entry ของบุคคล อาจมีการเก็บ ชื่อ นามสกุล อายุ เพศ ซึ่งหัวข้อของคุณสมบัติที่เราสนใจนั้น เราเรียกว่า attribute และในแต่ละ attribute จะมีการกำหนดชนิดของข้อมูลและค่าของข้อมูลที่จะทำการเก็บ

โดยที่ชนิดของ attribute หรือ Type นั้นจะต้องสัมพันธ์กับ รูปแบบชนิดของข้อมูล หรือ Syntax ซึ่ง Syntax จะเป็นตัวกำหนดว่าข้อมูลชนิดใดบ้างที่ Directory Server สามารถเก็บได้ เช่น Attribute ของหมายเลขโทรศัพท์ จะมีชนิดของข้อมูลเป็น caseIgnoreString ซึ่งรูปแบบข้อมูลนี้จะไม่สนใจเครื่องหมาย "-" และช่องว่างในระหว่างการทำการค้นหาหมายเลขโทรศัพท์

2.2.2 LDAP naming model

โดยปกติแล้ว Entry ของข้อมูลจะมีโครงสร้างแบบ Tree ซึ่งจะมีโครงสร้างตามความสัมพันธ์ของข้อมูลที่เราทำการจัดเก็บลงไป แต่ละ entry จะมีชื่อเรียกโดยชื่อนี้จะอ้างอิงถึงตำแหน่งของข้อมูลตามลำดับชั้น เราเรียกชื่อที่ใช้อ้างอิงนี้ว่า Distinguished name (DN) และสำหรับการอ้างถึงส่วนประกอบต่าง ๆ ใน DN เราจะใช้ Relative distinguished name (RDN) ในการอ้าง

Directory Service จะเก็บรายละเอียดของข้อมูลต่างๆ ใน DIB (Directory Information Base) โดย DIB จะถูกจัดเก็บในรูปแบบของ Tree เรียกว่า DIT (Directory Information Tree) โครงสร้าง DIB ประกอบไปด้วย ส่วนที่เรียกว่า Entry ซึ่งแต่ละ Entry จะทำหน้าที่เก็บข้อมูลของ Object ได้ Object หนึ่ง โดยในแต่ละ Entry จะประกอบไปด้วยส่วนที่เรียกว่า Attributes ซึ่งแต่ละ Attribute อาจมีข้อมูลเก็บอยู่เพียงหนึ่งค่า หรืออาจจะมีการเก็บค่าหลายค่าไว้ใน Attribute ก็ได้ ชนิดของ Attribute จะขึ้นอยู่กับ Class ของ Object ที่ Entry นั้นๆ ซึ่งได้ถูกกำหนดขึ้นมา

2.2.3 LDAP functional model

ในส่วนของ Functional Model จะเป็นส่วนที่กำหนดวิธีการจัดการข้อมูลที่อยู่ใน Directory Service โดยจะแบ่งลักษณะการทำงานออก เป็นกลุ่ม ๆ ได้ดังนี้

กลุ่มการค้นหาและเปรียบเทียบข้อมูล โดยจะมีคำสั่ง Search และ Compare อยู่ในกลุ่มนี้

กลุ่มคำสั่งการแก้ไขข้อมูล โดยในกลุ่มนี้จะประกอบไปด้วยคำสั่ง Add, Delete, Modify, Modify RDN

กลุ่มคำสั่งทางด้านการตรวจสอบสิทธ์ ประกอบด้วยคำสั่ง Bind, Unbind, Abandon โดยที่คำสั่ง Bind และ Unbind ทำหน้าที่ในการตรวจสอบสิทธิ์การเข้าถึงข้อมูลใน Directory Service ส่วนคำสั่ง Abandonใช้ในการ ยกเลิกการทำงานที่กำลังดำเนินการอยู่

2.2.4 LDAP security model

ในส่วนนี้จะกล่าวถึงจะกล่าวถึงในส่วนของการตรวจสอบสิทธิ์การใช้งานข้อมูลและการเข้ารหัสข้อมูลเพื่อป้องกันการดักจับข้อมูลเมื่อทำการส่งข้อมูลผ่านระบบเครือข่าย โดยใน LDAP จะมีการป้องกันการเข้าถึงข้อมูล 3 แบบคือ

- ไม่มีการป้องกันการเข้าถึงข้อมูล ใช้ในกรณีที่ข้อมูลนั้นเป็นข้อมูลที่เปิดเผยและสามารถเรียกดูได้จากทุกคน เช่น สมุดโทรศัพท์ แต่จะมีการป้องกันการเข้าถึงข้อมูลบางอย่างได้
- การตรวจสอบสิทธิ์การเข้าถึงข้อมูลแบบปกติ
- การตรวจสอบสิทธิ์การเข้าถึงข้อมูลและการเข้ารหัสข้อมูล (Simple Authentication and Security Layer)

ในการเก็บข้อมูลลงไปยังระบบ Directory Service นั้น ในส่วนโครงสร้างของ LDAP จะมีวิธีการเก็บข้อมูลตามโครงสร้างของ "Entries" โดยในแต่ละ Entries นั้นจะประกอบไปด้วย Attributes ที่มีค่าไม่ซ้ำกันเลย ที่เรียกว่า Distinguished Name (DN) ดังนั้นเราจึงนำเอา DN มาใช้ในการอ้างอิงถึง Entries ที่อยู่ใน LDAP ในแต่ละ Attribute จะประกอบไปด้วย ชนิดของข้อมูลที่เก็บใน Attribute นั้น ๆ

และ ค่าของข้อมูลที่เก็บใน Attribute นั้น ซึ่งอาจจะมีเพียงค่าเดียวหรือ หลายค่าก็ได้ โดยส่วนมากแล้วชนิดของข้อมูลมักเป็นชนิดตัวอักษร ส่วนรูปแบบของของค่าที่สามารถเก็บได้นั้นจะขึ้นอยู่กับชนิดของ Attribute

ใน LDAP Entries จะถูกทำการจัดเรียงในรูปแบบโครงสร้างต้นไม้ โดยปกติแล้วรูปแบบโครงสร้างจะเป็นอย่างไรนั้น ขึ้นอยู่กับโครงสร้างของข้อมูลที่เราต้องการ

เราสามารถทำการอ้างถึงข้อมูลที่เราต้องการได้ โดยใช้สิ่งที่เรียกว่า Distinguished name ซึ่งประกอบมาจากชื่อของแต่ละ Entry นำมาประกอบกัน (ชื่อของ Entry บางครั้งถูกเรียกว่า Relative Distinguished Name หรือ RDN)

2.3 ระบบ Account บนระบบปฏิบัติการ Unix

Unix เป็นระบบปฏิบัติการแบบหลากผู้ใช้หลายงาน (Multiuser/Multitasking) คือระบบสามารถทำงานได้หลายอย่างพร้อมกันในเวลาเดียวกัน ระบบอนุญาตให้ผู้ใช้หลายคนเข้ามาใช้งานระบบ โดยที่ระบบมีวิธีการในการจำแนกผู้ใช้แต่ละคน ซึ่งใช้งานอยู่บนระบบเดียวกันได้ ผู้ใช้หลายคนสามารถใช้งานเครื่องเดียวกันพร้อมกันได้ โดยใช้ลักษณะของการแบ่งเวลา (Time sharing) และแบ่งทรัพยากร (Resource sharing)

ไฟล์	รายละเอียด
/etc/passwd	เก็บข้อมูลบัญชีผู้ใช้ (user account)
/etc/shadow	เก็บข้อมูล password ของผู้ใช้ในรูปของ encrypt
/etc/group	เก็บข้อมูลของกลุ่มผู้ใช้ (group)

ตารางที่ 2.1 แสดงไฟล์ที่เกียวข้องกับระบบ Account

2.3.1 ไฟล์ /etc/passwd

บนระบบ Unix จะต้องทำการเก็บข้อมูลของผู้ที่จะทำการตรวจสอบสิทธ์ไว้ด้วย โดยปกติบนระบบ Unix จะมี File ที่ชื่อ /etc/passwd ซึ่งจะทำการเก็บข้อมูลของผู้ใช้งานทั้งหมดบนระบบไว้ แต่ละบรรทัดของ File จะประกอบด้วยข้อมูลต่าง ๆ ของผู้ใช้งานระบบแต่ละคน

ไฟล์ passwd เป็นไฟล์ที่เก็บข้อมูลเกี่ยวกับผู้ใช้ทั้งหมด โดยจะเก็บหนึ่งบรรทัดต่อผู้ใช้หนึ่งคนของระบบ โดยมีตัว colon(:) จะใช้แยกแต่ละประเภทข้อมูล

Field	คำอธิบาย
Username	-ใน Field นี้จะเก็บชื่อ Login Name บนระบบ -โดยที่ชื่อแต่ละคนจะต้องไม่เหมือนกัน
Password	-ใน Field นี้จะเก็บ Password ของผู้ใช้ จะเป็นรูปแบบที่ถูก Encrypt แล้ว -ถ้าเลือกใช้แบบ shadow password ค่าที่เก็บจะเป็น (x) ส่วน password ที่ถูก encrypt นั้นจะเก็บไว้ที่ไฟล์ /etc/shadow
User ID	-ใน field จะเก็บ User ID (UID) -โดยที่ UID ของแต่ละคนนั้นจะต้องไม่เหมือนกัน -ค่าของ UID เริ่มตั้งแต่ 0 ขึ้นไป
Group ID	-ใน field จะเก็บ Group ID (GID) ที่เป็น primary group ของผู้ใช้
Comment	-ใน field จะเก็บชื่อเต็มของผู้ใช้
Home Directory	-ใน field จะเก็บ path ไดเร็กทอรี HOME ของผู้ใช้
Login Shell	-ใน field จะเก็บ login shell หรือคำสั่งที่ต้องการให้ผู้ใช้ทำงานเมื่อผู้ใช้ Login เข้ามาในระบบ

ตารางที่ 2.2 แสดงข้อมูลที่เก็บอยู่ในไฟล์ passwd

โดยปกติแล้ว File /etc/passwd จะไม่มีการเก็บรหัสผ่านของผู้ใช้งานระบบไว้จะมีเพียงตัวบ่งบอกว่าผู้ใช้งานมีการเก็บบันทึกรหัสผ่านไว้หรือไม่ ส่วนรหัสผ่านจะถูกเก็บไว้ใน File ต่างหากที่ชื่อว่า /etc/shadow โดยจะนำเอารหัสผ่านของผู้ใช้งานมาทำการเข้ารหัสแบบ Data Encryption Standard (DES) และจึงทำการบันทึกไว้ใน File

2.3.2 ไฟล์ /etc/shadow

ระบบ shadow password เป็นระบบที่ใช้เพื่อให้เกิดความปลอดภัยกับการเก็บ password ของผู้ใช้แยกในไฟล์ต่างหาก โดยจะเก็บหนึ่งบรรทัดต่อผู้ใช้หนึ่งคนของระบบ โดยมีตัว colon(:) จะใช้แยกแต่ละประเภทข้อมูล

Field	คำอธิบาย
Username	-ใน Field นี้จะเก็บชื่อ Login Name บนระบบ
Password	-ใน Field นี้จะเก็บ Password ของผู้ใช้ จะเป็นรูปแบบที่ถูก Encrypt แล้ว
Last	-วันสุดท้ายที่มีการเปลี่ยน password ตั้งแต่ 1 มกราคม ค.ศ.1970
May	-วันที่ก่อน password อาจถูกเปลี่ยน
Must	-วันที่หลัง password ต้องถูกเปลี่ยน
Warn	-วันที่ก่อน password หมดอายุและมีการเตือนผู้ใช้
Expire	-วันที่หลัง password หมดอายุและ account ไม่สามาถใช้งานได้
Disable	-วันที่ account ถูก disable ตั้งแต่ 1 มกราคม ค.ศ.1970
Reserved	-Field สำรอง

ตารางที่ 2.3 แสดงข้อมูลที่เก็บอยู่ในไฟล์ shadow

3.2.3 ไฟล์ /etc/group

นอกจากนี้บนระบบ Unix ยังมีการแบ่งกลุ่มผู้ใช้งาน ออกเป็นกลุ่ม ๆ อีกด้วยเพื่อง่ายต่อการกำหนดสิทธิ์การ เข้าถึง โดยการแบ่งแยกกลุ่มจะถูกระบุอยู่ใน File /etc/group โครงสร้างของ File จะประกอบไปด้วย ข้อมูลของกลุ่มที่มีอยู่ในระบบนั้น ๆ โดยจะมีรูปแบบดังต่อไปนี้

กลุ่มของผู้ใช้ (Group) เป็นกลุ่มของผู้ใช้ที่นำมารวมกันสำหรับวัตถุประสงค์ที่เหมือนกัน โดยไฟล์ /ete/group จะเก็บหนึ่งบรรทัดต่อกลุ่มผู้ใช้หนึ่งกลุ่ม โดยมีตัว colon(:) จะใช้แยกแต่ละประเภทข้อมูล

Field	คำอธิบาย
Group Name	-ใน Field นี้จะเก็บชื่อของกลุ่มผู้ใช้ -โดยที่ชื่อแต่ละกลุ่มจะต้องไม่เหมือนกัน
Password	-ใน Field นี้จะเก็บ Password ของกลุ่มผู้ใช้ -ถ้าเลือกใช้แบบ shadow password ค่าที่เก็บจะเป็น (x) ส่วน password ที่ถูก encrypt นั้นจะเก็บไว้ที่ไฟล์ /etc/gshadow
Group ID	-ใน field จะเก็บ Group ID (GID) -โดยที่ GID ของแต่ละกลุ่มนั้นจะต้องไม่เหมือนกัน -ค่าของ GID เริ่มตั้งแต่ 0 ขึ้นไป
Users	-รายชื่อของผู้ใช้ที่อยู่ในกลุ่มโดยจะให้ (,) คั้น

ตารางที่ 2.4 แสดงข้อมูลที่เก็บอยู่ในไฟล์ group

บทที่ 3
การวิเคราะห์และออกแบบ

บทนี้จะกล่าวถึงการศึกษาและวิเคราะห์ระบบงานในปัจจุบัน เพื่อหาต้องการของระบบ รวมถึงการออกแบบโครงงานในพัฒนาระบบ ซึ่งประกอบด้วยการออกแบบโครงสร้างของข้อมูล และการทำงานของโปรแกรม

3.1 ลักษณะของระบบงานปัจจุบัน

สำนักวิจัยและบริการคอมพิวเตอร์ ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นหน่วยงานมีหน้าที่ดูแลระบบสารสนเทศ และให้บริการระบบคอมพิวเตอร์เพื่อประโยชน์ทางด้านวิชาการ การเรียนการสอน และงานบริหารแก่คณะ และหน่วยงานต่างๆ ของมหาวิทยาลัยฯ โดยมีฝ่ายงานที่ความเกี่ยวข้องกับข้อมูลบัญชีผู้ใช้งานมี 3 ฝ่าย คือ ฝ่ายระบบและโปรแกรม ฝ่ายควบคุมเครื่อง ฝ่ายระบบเครือข่าย

3.1.1 ฝ่ายงานและความรับผิดชอบ

3.1.1.1 ฝ่ายควบคุมเครื่อง

ทำหน้าที่ จัดการบัญชีผู้ใช้งาน ดูแลเครื่องคอมพิวเตอร์ให้สามารถบริการผู้ใช้งาน สร้างบริการต่างๆ เพื่อตอบสนองความต้องการของผู้ใช้ และดูแล ซ่อมบำรุง เครื่องและบริการต่างๆ ให้ทำงานได้อย่างมีประสิทธิภาพ ซึ่งมีงานบริการหลัก ดังนี้

- งานบริการด้านอินเทอร์เน็ต ใช้เครื่องคอมพิวเตอร์ชื่อ เจ้าคุณ (Chaokhun) เป็นผลิตภัณฑ์ ของ SUN SPARC S2000E เพื่อตอบสนองงานบริการและโปรแกรมประยุกต์ ทางอินเทอร์เน็ต แก่ข้าราชการ นักศึกษาของสถาบัน เช่น Web Browser, ftp, email และอื่นๆ
- งานบริการด้านโปรแกรมประยุกต์ ใช้เครื่องคอมพิวเตอร์ชื่อ แคแสด (Khaesad) เป็นผลิตภัณฑ์ของ SUN SPARC S2000E เพื่อบริการ โปรแกรมประยุกต์งานออกแบบทางวิศวกรรม
- งานบริการด้านฐานข้อมูล ใช้เครื่องคอมพิวเตอร์ชื่อ ร่มเกล้า (Romklo) เป็นผลิตภัณฑ์ของ HP T-520 ใช้ซอฟต์แวร์ ORACLE เพื่อรองรับระบบสารสนเทศของสถาบันฯ ที่มีการใช้ระบบฐานข้อมูล

- งานบริการคอมพิวเตอร์ทำงานแบบขนานความ เร็วสูง ใช้เครื่องคอมพิวเตอร์ชื่อ นวมาส (Nawamas) เป็นผลิตภัณฑ์ของ HP-CONVEX Exemplar SPP1600/XA ให้บริการแก่ นักศึกษา ข้าราชการ ทั้งภายในและ ภายนอกสถาบัน ที่ต้องการ ทำงานวิจัย ที่ต้องใช้การคำนวณความเร็วสูงมาก

3.1.1.2 ฝ่ายระบบเครือข่าย

ทำหน้าที่ควบคุมและจัดการบริการบนเครือข่าย ดูแลการเชื่อมต่อของระบบเครือข่ายทั้งหมดภายในสถาบันฯ ดูแลการเชื่อมต่อระบบอินเทอร์เน็ท และดูแลการให้บริการใช้งานระบบจากการทำงานระยะไกลผ่านการเชื่อมต่อคอมพิวเตอร์ผ่านโมเดม ซึ่งในส่วนของการเชื่อมต่อคอมพิวเตอร์ผ่านโมเดมจะมี PC Server ให้บริการอยู่ 2 เครื่อง เครื่องแรกทำงานบนระบบปฏิบัติการ Microsoft Windows NT Advanced Server Version 3.51 และ Microsoft Back-Office มีโปรแกรม Shiva Manager เพื่อดูแลการบริการเชื่อมต่อผ่านโมเดม ส่วนอีกเครื่องทำงานระบบปฏิบัติการ Sun เพื่อทำหน้าที่เป็น Authentication Server ให้แก่เครื่องเครื่องแรก

3.1.1.3 ฝ่ายระบบและโปรแกรม

ทำหน้าเขียนโปรแกรมและพัฒนาระบบงานต่างๆ เพื่อใช้ในสถาบันฯ เช่น ระบบทะเบียนนักศึกษา ระบบการลงทะเบียนผ่านอินเทอร์เน็ต ระบบบุคลากรของสถาบัน ระบบการเงิน กองคลัง เงินรายได้ ระบบหลักสูตร ระบบอาคารสถานที่ พัสดุ ครุภัณฑ์ ระบบยานพาหนะ ระบบทุนการศึกษา เป็นต้น ซึ่งเป็นการทำงานจะเป็นการเขียนระบบโปรแกรม เพื่อติดต่อกับฐานข้อมูล ORACLE ซึ่งให้บริการอยู่บนเครื่องร่มเกล้า ตัวอย่างระบบงานของฝ่ายระบบแบบและโปรแกรม คือ ระบบทะเบียนนักศึกษา ซึ่งสามารถให้บริการ โดยต้องทาการข้อรหัสจากหน้าจอผ่านอินเทอร์เน็ต จากนั้นระบบจะส่งรหัสผ่านสำหรับผู้ใช้ไปทางอีเมล์ของผู้ใช้ที่มีอยู่กับทางสถาบันฯ จึงนำเอา ชื่อผู้ใช้และรหัสผ่านมา ที่หน้าจอเดิมเพื่อใช้บริการต่างๆ ต่อไป

3.1.2 ขั้นตอนการปฏิบัติงาน

ระบบที่เกี่ยวของกับข้อมูลบัญชีรายชื่อผู้ใช้ มีข้อมูลที่เกี่ยวข้อง คือ ชื่อเข้าใช้และรหัสผ่านของนักศึกษาและบุคลากรทั้งในและนอกสถาบันฯ มีการทำงานเริ่มจากทะเบียนมีการรายชื่อของนักศึกษาหรือบุคลากรใหม่ที่เพิ่มเข้ามา จะมีการกำหนดรหัสประจำตัวนักศึกษา หรือบุคลากรขึ้นมาให้กับแต่ละคน และทำการสร้างรหัสผ่านโดยการสุ่ม เพื่อใช้งาน หรือเพื่อต้องการดูข้อมูล ซึ่งจะมีการเก็บในฐานข้อมูล ORACLE บนเครื่องเซิร์ฟเวอร์ร่มเกล้า

ระบบควบคุมเครื่องจะต้องการรายชื่อ รหัสประจำตัว และรหัสคณะ เพื่อนำไปสร้างรายชื่อผู้ใช้งานในระบบอีเมล์และระบบผู้ใช้งานบนเครื่องเซิร์ฟเวอร์ จะทำการร้องขอเจ้าหน้าที่ของฝ่ายระบบและโปรแกรมให้ส่ง รายชื่อ รหัสประจำตัว และรหัสคณะมาให้ เมื่อได้รายชื่อ และข้อมูลต่างๆ แล้ว เพื่อให้รายชื่อใหม่สามารถใช้งานบนเครื่องเซิร์ฟเวอร์เจ้าคุณ ซึ่งทำงานบนระบบปฏิบัติการ Unix เพื่อให้สามารถบริการอินเทอร์เน็ต และอีเมล์ได้ โดยเจ้าหน้าที่จะทำการแปลรายชื่อที่ได้ให้เป็นไฟล์ข้อความ โดยแก้ไข และเพิ่มข้อมูลสำหรับรายละเอียดของผู้ใช้บนระบบปฏิบัติการ Unix ที่จำเป็น และได้เปลี่ยนแปลงข้อมูลบางส่วน โดยเพิ่มตัวอักษรเข้าไปข้างหน้าของรหัสประจำตัว เพื่อใช้เป็น Username ส่วน Password จะทำการสุ่มขึ้นมาใหม่ กลุ่มของผู้ใช้และส่วนอื่นๆ ก็ถูกกำหนดเพิ่มให้ครบ ซึ่งทำงานด้วยโปรแกรมที่เขียนจากภาษาซี โดยไฟล์ข้อความที่เกิดจากโปรแกรมสามารถนำไปต่อกับข้อมูลของผู้ใช้ระบบ Unix เดิมได้ทันที

ในฝ่ายระบบเครือข่าย ระบบที่เกี่ยวข้องกับบัญชีผู้ใช้งาน คือบริการเชื่อมต่อคอมพิวเตอร์ผ่านโมเด็ม ซึ่งจะใช้ชื่อและรหัสผ่านตัวเดียวกับฝ่ายควบคุมเครื่อง ซึ่งในปัจจุบันจะทำโดยใช้วิธีคัดลอกไฟล์ผู้ใช้จากเครื่องเจ้าคุณ ของฝ่ายควบคุมเครื่อง ไปสู่เครื่องของฝ่ายระบบเครือข่ายที่มีระบบปฏิบัติเดียวกัน เพื่อให้ได้ข้อมูลของ ชื่อและรหัสผ่าน จากนั้นจะใช้โปรแกรม Shiva Manager ซึ่งเป็นโปรแกรมควบคุมการการใช้งานของผู้ใช้ที่เชื่อมต่อคอมพิวเตอร์ผ่านโมเดม ซึ่งทำงานบนเครื่องระบบปฏิบัติการ Window NT ดึงข้อมูลชื่อและรหัสจาก เครื่องระบบปฏิบัติการ Sun ที่ได้รับข้อมูลบัญชีผู้ใช้ที่คัดลอกมา ทำหน้าที่เป็น Authentication Server อีกที

3.1.3 ปัญหาและข้อจำกัดของระบบงานเดิม

- ข้อมูลในแต่ละฝ่าย ทั้งฝ่ายควบคุมเครื่อง ฝ่ายระบบเครือข่าย และฝ่ายระบบและโปรแกรมมีการแยกเก็บเป็นของตนเอง ทำให้เมื่อมีการปรับปรุงข้อมูล ข้อมูลไม่มีการเปลี่ยนทันทีในทุกฝ่าย ทำให้ข้อมูลอาจมีความขัดแย้ง ไม่ตรงกัน
- ข้อมูลของผู้ใช้ ฝ่ายควบคุมเครื่องกับฝ่ายระบบและโปรแกรม มีลักษณะที่แตกต่างกัน และใช้รหัสผ่านของผู้ใช้เป็นคนละตัวกัน ทำให้มีลักษณะเป็นข้อมูลคนละชุด ซึ่งทำให้การใช้งานต้องจำรหัสผ่านหลายชุด ทำให้ผู้ใช้เกิดความไม่สะดวกในการใช้งาน
- ข้อมูลของผู้ใช้เชื่อมต่อคอมพิวเตอร์ผ่านโมเดม ของฝ่ายระบบเครือข่าย ซึ่งใช้วิธีคัดลอกข้อมูลบัญชี ผู้ใช้จากฝ่ายควบคุมเครื่อง มาไว้ที่เครื่องที่ทำหน้าที่ Authentication Server ซึ่งไม่ได้ใช้ประโยชน์อย่างอื่น ทำให้สูญเสียทรัพยากร เนื่องจากไม่ได้ใช้งานเครื่องให้เต็มประสิทธิภาพ

- การปรับปรุงข้อมูลของบัญชีผู้ใช้จากระบบหนึ่งไปสู่อีกระบบ ใช้วิธีคัดลอก และใช้แรงงานคน ถึงแม้ในปัจจุบันไม่มีปัญหาในการใช้งาน แต่ก็ทำให้เกิดความไม่สะดวก และใช้เวลามากกว่าระบบอัตโนมัติ
- ขาดความยืดหยุ่นในการขยายระบบในอนาคต เนื่องจากเมื่อมีการเพิ่มเครื่องเซิร์ฟเวอร์ แล้วต้องใช้แรงงานคนในการคัดลอกข้อมูลบัญชีผู้ใช้ไปให้ระบบใหม่ ซึ่งจะเป็นการไม่สะดวก และอาจทำมีปัญหาในอนาคต

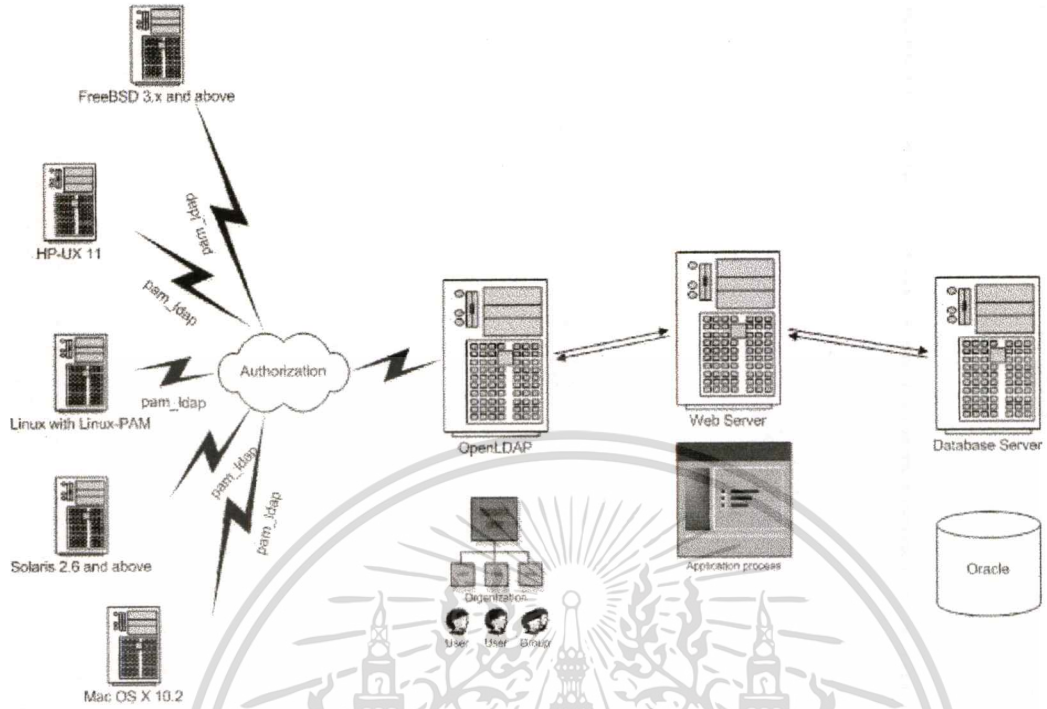
3.2 ความต้องการของระบบ

จากการศึกษาระบบที่ผ่านมา รวมทั้งพิจารณาทฤษฎีที่เกี่ยวข้องในบทที่ผ่านมา จึงสามารถกำหนดความต้องการเพื่อให้ระบบที่พัฒนาขึ้นสามารถแก้ปัญหาต่างๆ ดังต่อไปนี้ได้

- สามารถลดความซ้ำซ้อน และการใช้งานจากศูนย์กลางในการบริหารบัญชีผู้ใช้งาน
- สามารถสร้างระบบที่น่าเชื่อถือ ทั้งทางด้านการทำงาน และความปลอดภัย เพื่อใช้เก็บข้อมูลบัญชีรายชื่อของผู้ใช้
- สามารถสร้างระบบที่ใช้เก็บข้อมูลบัญชีผู้ใช้งานที่มีมาตรฐานชัดเจน และทำให้สามารถใช้ร่วมกับระบบอื่น ทั้งที่มีอยู่แล้ว และระบบใหม่ ๆ ที่จะนำมาใช้งานในอนาคต
- เพื่อให้การทำงานของระบบเป็นระบบที่ใช้งานได้ง่าย สามารถให้บริการผู้ใช้งานด้วยข้อมูลมีความทันสมัย รวดเร็ว และสะดวกสบาย ลดเวลาในการเรียนรู้
- สามารถสร้างระบบที่มีความสามารถในการอ่านข้อมูล ค้นหาข้อมูล ดูและเปรียบเทียบข้อมูลได้อย่างถูกต้อง และรวดเร็ว

3.3 การออกแบบระบบ

การทำงานของระบบ Unix ที่รับรองสิทธิของผู้ใช้งานโดยใช้ LDAP Directory และใช้ LDAP Directory เพื่อใช้ในการเก็บข้อมูลรหัสผ่าน และสิทธิต่างๆ จะมีลักษณะการทำงานดังรูปที่ 3.1 โครงงานที่พัฒนาจะทำการสร้างโปรแกรมเพื่อทำการย้ายข้อมูลระหว่างฐานข้อมูลเดิมบน Oracle กับ LDAP Directory



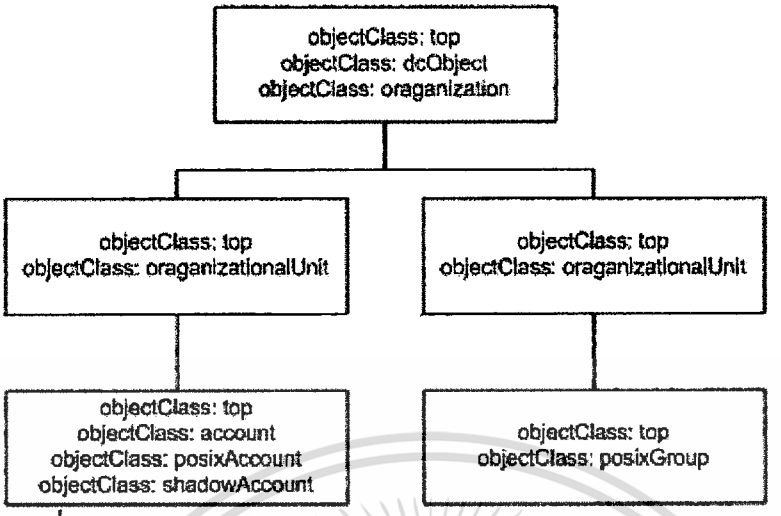
รูปที่ 3.1 แสดงโครงสร้างของระบบที่ใช้ LDAP Directory เพื่อใช้ในการ Authorization ระบบ Unix

3.3.1 องค์ประกอบหลักของระบบ

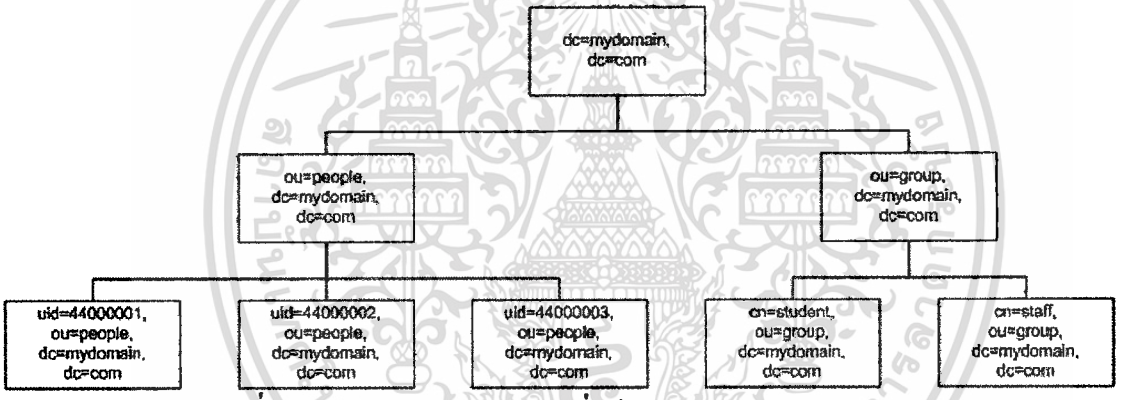
- ระบบปฏิบัติการ Unix
- ระบบ Directory Service
- ระบบฐานข้อมูล
- ระบบ Web Server
- โปรแกรมที่ทำให้ระบบปฏิบัติการรับรองผู้ใช้งานผ่าน LDAP Directory
- โปรแกรมโอนถ่ายข้อมูลระหว่างฐานข้อมูลกับ Directory Server

3.3.2 การออกแบบ Schema

ในการเก็บข้อมูลของผู้ใช้งานระบบเพื่อให้สามารถเก็บข้อมูลของผู้ใช้งานในระบบได้อย่างครบถ้วนและถูกต้อง เราจึงต้องออกแบบลักษณะของข้อมูลและกลุ่มของข้มมูลที่มีความสัมพันธ์ โดยจะแบ่งออกเป็นกลุ่ม ดังนี้



รูปที่ 3.2 แสดง Schema ของ LDAP Directory Tree ในแต่ละ Entry



รูปที่ 3.3 แสดงตัวอย่างของ DN ที่ใช้ใน LDAP Directory Tree

3.3.2.1 Entry type

posixAccount เป็น Entry ที่ใช้แสดงถึงรายละเอียดเกี่ยวกับข้อมูลทั่วของผู้ใช้ และข้อมูลเกี่ยวกับข้อมูลที่จำเป็นสำหรับ Account บนระบบปฏิบัติการ Unix

(nisSchema.2.0 NAME 'posixAccount' SUP top AUXILIARY

DESC 'Abstraction of an account with POSIX attributes'

MUST (cn $ uid $ uidNumber $ gidNumber $ homeDirectory)

MAY (userPassword $ loginShell $ gecos $ description))

Name	Equality	
cn	caseIgnoreMatch	SINGLE-VALUE
uid	caseIgnoreMatch	SINGLE-VALUE
uidNumber	integerMatch	SINGLE-VALUE
gidNumber	integerMatch	SINGLE-VALUE
homeDirectory	caseExactIA5Match	SINGLE-VALUE
userPassword	octetStringMatch	SINGLE-VALUE
loginShell	caseExactIA5Match	SINGLE-VALUE
gecos	caseIgnoreMatch	SINGLE-VALUE
description	caseIgnoreMatch	SINGLE-VALUE

ตารางที่ 3.1 แสดงรายละเอียดของ Attribute ของ posixAccount

shadowAccount เป็น Entry ที่ใช้แสดงถึงรายละเอียดเกี่ยวกับ รหัสผ่านของผู้ใช้ ซึ่งจะประกอบไปด้วยการกำหนดช่วงเวลาที่สามารถเข้ามาใช้งานระบบได้ รวมทั้งรหัสผ่านที่ได้มีการเข้ารหัสไว้

```
( nisSchema.2.1 NAME 'shadowAccount' SUP top AUXILIARY
  DESC 'Additional attributes for shadow passwords'
  MUST uid
  MAY ( userPassword $ shadowLastChange $ shadowMin
      shadowMax $ shadowWarning $ shadowInactive $
      shadowExpire $ shadowFlag $ description ) )
```

Name	Equality	
uid	caseIgnoreMatch	SINGLE-VALUE
userPassword	octetStringMatch	SINGLE-VALUE
shadowLastChange	integerMatch	SINGLE-VALUE
shadowMin	integerMatch	SINGLE-VALUE
shadowMax	integerMatch	SINGLE-VALUE
shadowWarning	integerMatch	SINGLE-VALUE
shadowInactive	integerMatch	SINGLE-VALUE
shadowExpire	integerMatch	SINGLE-VALUE
shadowFlag	integerMatch	SINGLE-VALUE
description	caseIgnoreMatch	SINGLE-VALUE

ตารางที่ 3.2 แสดงรายละเอียดของ Attribute ของ shadowAccount

poxisGroup เป็น Entry เพื่อเป็นใช้ระบุถึงกลุ่มของผู้ใช้งาน ว่าผู้ใช้งานอยู่ในกลุ่มใด

(nisSchema.2.2 NAME 'posixGroup' SUP top STRUCTURAL
DESC 'Abstraction of a group of accounts'
MUST (cn $ gidNumber)
MAY (userPassword $ memberUid $ description))

Name	Equality	
Cn	caseIgnoreMatch	SINGLE-VALUE
gidNumber	integerMatch	SINGLE-VALUE
userPassword	octetStringMatch	SINGLE-VALUE
memberUid	caseExactIA5Match	
description	caseIgnoreMatch	SINGLE-VALUE

ตารางที่ 3.3 แสดงรายละเอียดของ Attribute ของ posixGroup

3.3.2.2 Attribute type

- **cn** เป็นชื่อคนทั่วไปใช้ในการอ้างถึง Entity หรือ common name มีรายละเอียดใน RFC2256

  (2.5.4.3 NAME 'cn' SUP name)

- **description** เป็นข้อมูลที่ใช้แสดงรายละเอียด มีรายละเอียดใน RFC2256

  (2.5.4.13 NAME 'description' EQUALITY caseIgnoreMatch

  SUBSTR caseIgnoreSubstringsMatch

  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024})

- **userPassword** เป็นรหัสของผู้ใช้งาน มีรายละเอียดใน RFC2256/2307

  (2.5.4.35 NAME 'userPassword' EQUALITY octetStringMatch

  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{128})

- **uid** เป็นค่าที่ใช้ระบุถึงผู้ใช้งาน เป็นได้ทั้งตัวเลขหรือตัวอักษร มีรายละเอียดใน RFC1274

  (0.9.2342.19200300.100.1.1

  NAME ('uid' 'userid')

  DESC 'RFC1274: user identifier'

  EQUALITY caseIgnoreMatch

  SUBSTR caseIgnoreSubstringsMatch

  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256})

- **uidNumber** เป็นตัวเลขที่ใช้ในการอ้างอิงถึงผู้ใช้งานในระบบ

  (nisSchema.1.0 NAME 'uidNumber'

  DESC 'An integer uniquely identifying a user in an

  administrative domain'

  EQUALITY integerMatch SYNTAX 'INTEGER' SINGLE-VALUE)

- **gidNumber** เป็นตัวเลขที่ใช้ในการอ้างอิงถึงกลุ่มในระบบ

  (nisSchema.1.1 NAME 'gidNumber'

  DESC 'An integer uniquely identifying a group in an

  administrative domain'

  EQUALITY integerMatch SYNTAX 'INTEGER' SINGLE-VALUE)

- **gecos** เป็นชื่อ หรือ common name

  (nisSchema.1.2 NAME 'gecos'
  DESC 'The GECOS field; the common name'
  EQUALITY caseIgnoreIA5Match
  SUBSTRINGS caseIgnoreIA5SubstringsMatch
  SYNTAX 'IA5String' SINGLE-VALUE)

- **homeDirectory** เป็น path ที่ใช้แสดงถึง home directory ของผู้ใช้งาน

  (nisSchema.1.3 NAME 'homeDirectory'
  DESC 'The absolute path to the home directory'
  EQUALITY caseExactIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE)

- **loginShell** เป็น path ที่ใช้แสดงถึง Login Shell ของผู้ใช้งาน

  (nisSchema.1.4 NAME 'loginShell'
  DESC 'The path to the login shell'
  EQUALITY caseExactIA5Match
  SYNTAX 'IA5String' SINGLE-VALUE)

- **shadowLastChange** เป็นจำนวนวันของวันสุดท้ายที่มีการเปลี่ยน password ตั้งแต่ 1 มกราคม ค.ศ.1970

  (nisSchema.1.5 NAME 'shadowLastChange'
  EQUALITY integerMatch
  SYNTAX 'INTEGER' SINGLE-VALUE)

- **shadowMin** ผู้ใช้งานสามารถเปลี่ยน password ได้หลังจากจำนวนวันที่กำหนด

  (nisSchema.1.6 NAME 'shadowMin'
  EQUALITY integerMatch
  SYNTAX 'INTEGER' SINGLE-VALUE)

- **shadowMax** ผู้ใช้งานจะต้องเปลี่ยน password ได้หลังจากจำนวนวันที่กำหนด

  (nisSchema.1.7 NAME 'shadowMax'
  EQUALITY integerMatch
  SYNTAX 'INTEGER' SINGLE-VALUE)

- **shadowWarning** เป็นจำนวนวันที่มีการเตือนผู้ใช้ก่อนวันหมดอายุของ password

  (nisSchema.1.8 NAME 'shadowWarning'

  EQUALITY integerMatch

  SYNTAX 'INTEGER' SINGLE-VALUE)

- **shadowInactive** เป็นจำนวนวันตั้งแต่ 1 มกราคม ค.ศ.1970 ซึ่ง account จะถูก disable

  (nisSchema.1.9 NAME 'shadowInactive'

  EQUALITY integerMatch

  SYNTAX 'INTEGER' SINGLE-VALUE)

- **shadowExpire** เป็นจำนวนวันหลัง password หมดอายุและaccountจะไม่สามารถใช้งานได้

  (nisSchema.1.10 NAME 'shadowExpire'

  EQUALITY integerMatch

  SYNTAX 'INTEGER' SINGLE-VALUE)

- **shadowFlag** เป็นข้อมูลสำรอง ยังไม่มีการใช้งาน

  (nisSchema.1.11 NAME 'shadowFlag'

  EQUALITY integerMatch

  SYNTAX 'INTEGER' SINGLE-VALUE)

- **memberUid** เป็นรายชื่อของผู้ใช้ที่อยู่ในกลุ่ม

  (nisSchema.1.12 NAME 'memberUid'

  EQUALITY caseExactIA5Match

  SUBSTRINGS caseExactIA5SubstringsMatch

  SYNTAX 'IA5String')

3.3.3 การทำงานของระบบ

ระบบการทำงานจะทำการจับคู่ข้อมูลจากฐานข้อมูลนำมาใส่ให้กับ Directory แต่เนื่องจากให้ฐานข้อมูลการเก็บข้อมูลบางอย่างเท่านั้น จึงจะต้องมาการเพิ่มเติมข้อมูลเข้ามาเพื่อให้เกิดความสมบูรณ์ สำหรับข้อมูลบัญชีผู้ใช้งานบนระบบ UNIX ซึ่งตัวอย่างข้อมูลที่เก็บในฐานข้อมูลเป็นดังนี้

User_ID: 44067080

User_Name: Chirawat Santimitra

User_Password: {CRYPT}3nOriDvnbnStU

แต่ในการเก็บใน Directory เพื่อให้สามารถนำไปใช้งานกับระบบบัญชีผู้ใช้งานได้อย่างสมบูรณ์ จะต้องมีการเก็บดังที่แสดงต่อไปนี้

cn: Chirawat Santimitra

uid: 44067080

userPassword: {CRYPT}3nOriDvnbnStU

uidNumber: 44067080

gidNumber: 500

loginShell: /bin/bash

homeDirectory: /home/s4067080

gecos: Chirawat Santimitra

description: Chirawat Santimitra

shadowLastChange: 17560

shadowMin: -1

shadowMax: -1

shadowWarning: -1

shadowInactive: -1

shadowExpire: -1

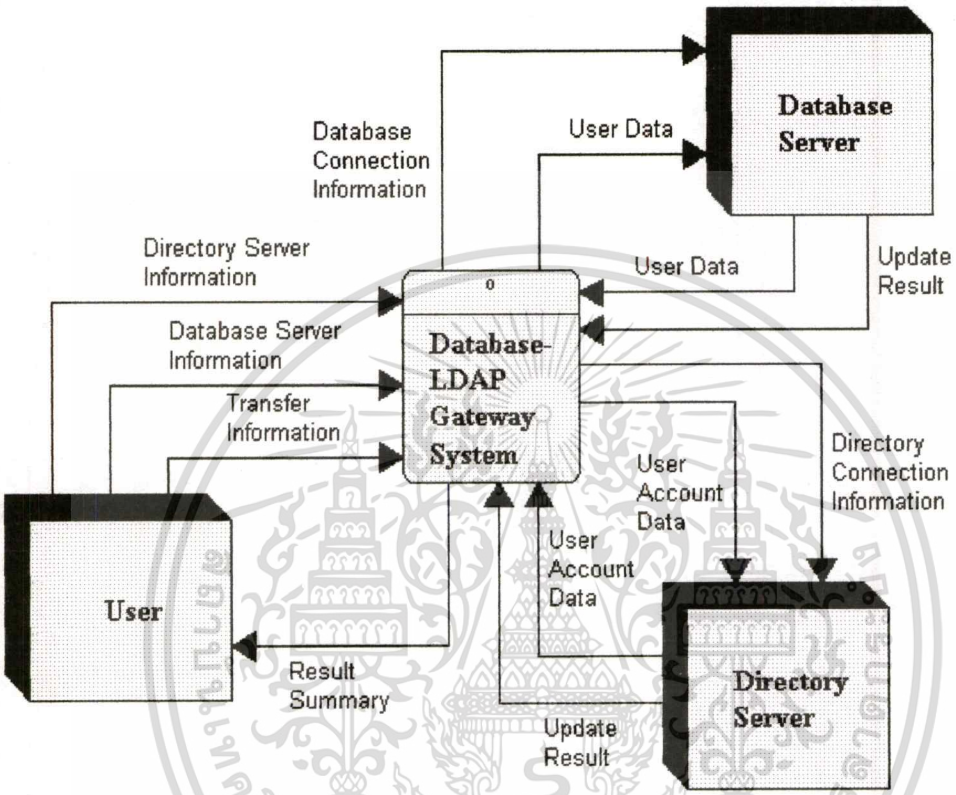
shadowFlag: -1

จากการสังเกตจะมีข้อมูลบ้างตัวสามารถนำมาใช้ได้เลยบางส่วนจะต้องมีการดัดแปลง และบางส่วนจะต้องมีการกำหนดค่าขึ้นมาใหม่ ดังที่แสดงในตารางที่ 3.4

LDAP Schema	Column
cn	User_Name
uid	User_ID
userPassword	User_Password
uidNumber	User_ID
gidNumber	ต้องกำหนดเพิ่ม
loginShell	ต้องกำหนดเพิ่ม
homeDirectory	ต้องกำหนดเพิ่ม
gecos	User_Name
description	ต้องกำหนดเพิ่ม
shadowLastChange	ต้องกำหนดเพิ่ม
shadowMin	ต้องกำหนดเพิ่ม
shadowMax	ต้องกำหนดเพิ่ม
shadowWarning	ต้องกำหนดเพิ่ม
shadowInactive	ต้องกำหนดเพิ่ม
shadowExpire	ต้องกำหนดเพิ่ม
shadowFlag	ต้องกำหนดเพิ่ม

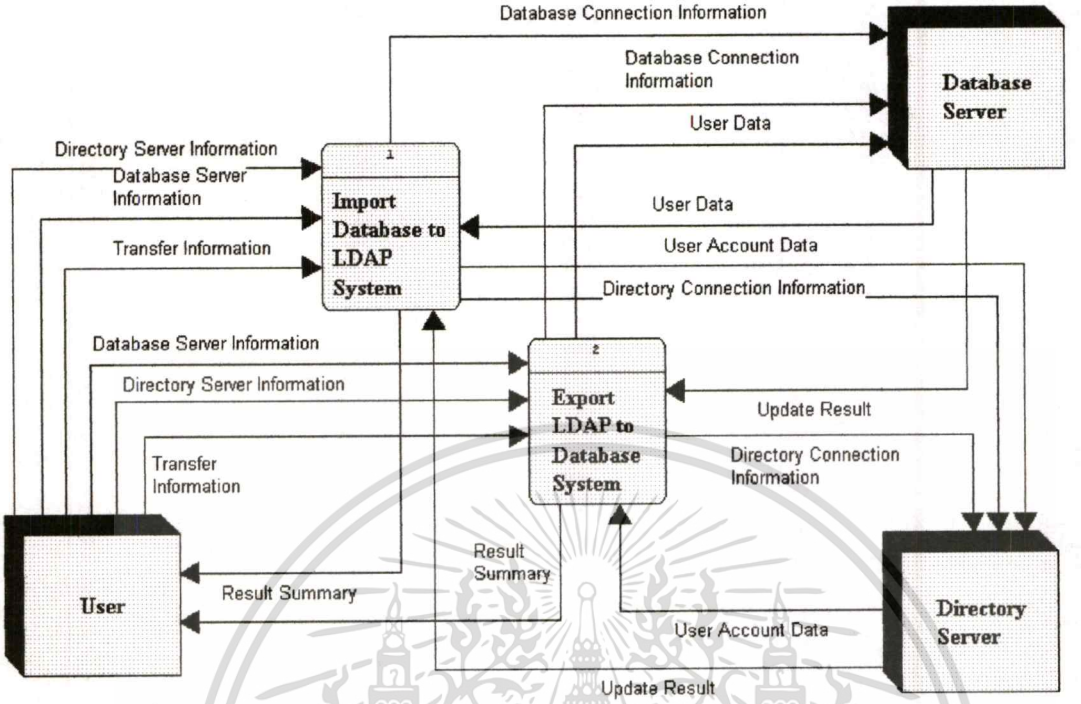
ตารางที่ 3.4 แสดงรายละเอียดของ Attribute ที่ต้องกำหนดเพิ่ม เพื่อเก็บใน Directory

ระบบ Database-LDAP Gateway System จากการวิเคราะห์ถึงความต้องการแล้ว จึงได้ออกแบบ Data Flow Diagram ดังที่แสดงตามต่อไปนี้



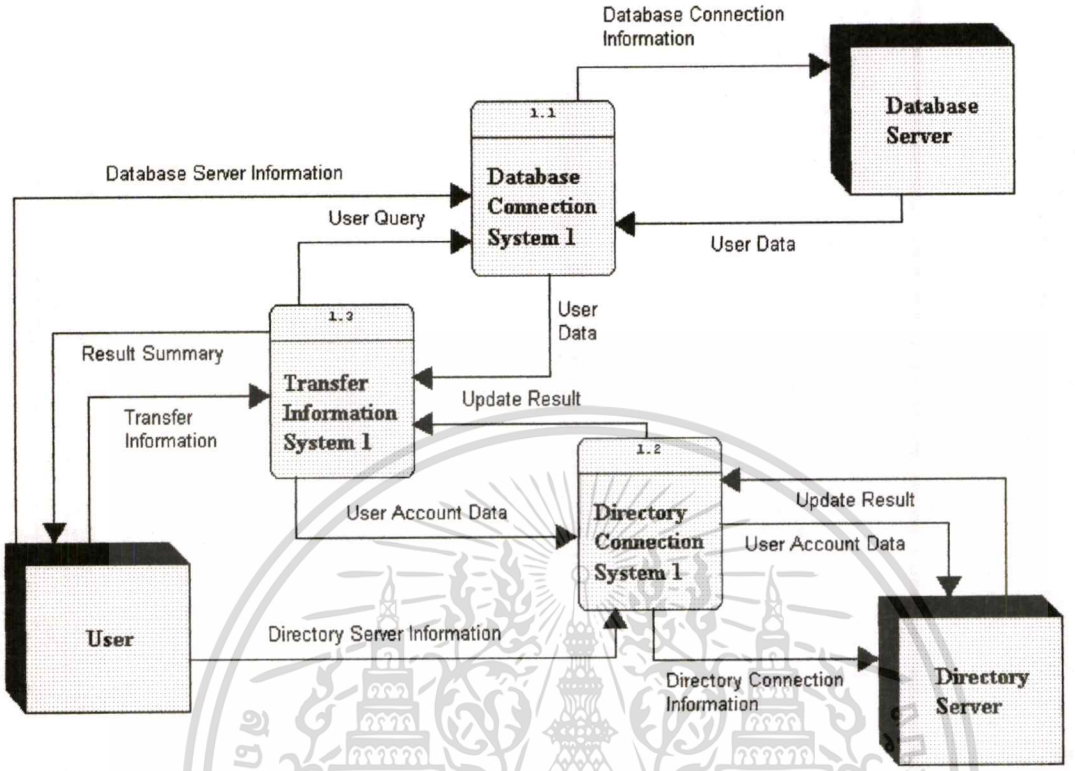
รูปที่ 3.4 แสดง Context Diagram ของระบบ Database-LDAP Gateway System

ระบบการทำงานจะประกอบด้วยผู้ใช้งาน ระบบฐานข้อมูลและระบบ Directory Service ที่จะทำการติดต่อด้วย ดังที่แสดงในรูปที่ 3.4 ซึ่งเป็น Context Diagram โดยผู้ใช้ต้องส่งรายละเอียดเกี่ยวกับฐานข้อมูล Directory และรายละเอียดเกี่ยวกับการโอนย้ายข้อมูล ไปยังระบบ Gateway ซึ่งระบบจะนำไปใช้ในการติดต่อกับระบบที่เกี่ยวข้องอีกที



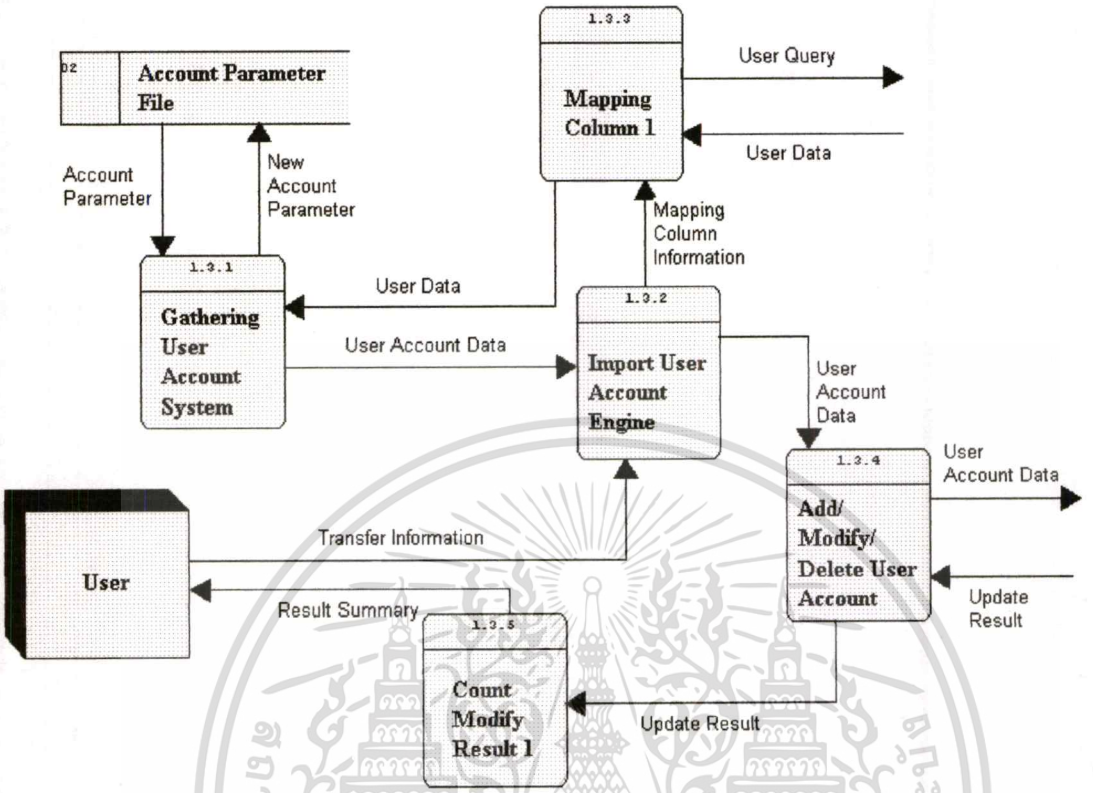
รูปที่ 3.5 แสดง Diagram 0 DFD ของระบบ Database-LDAP Gateway System

ในระบบ Database-LDAP Gateway System จะแบ่งระบบภายในออกเป็นสองส่วน คือ ส่วนที่ใช้การโอนย้ายข้อมูลจากฐานข้อมูลไปยัง Directory และส่วนที่ใช้ในการโอนย้ายข้อมูลจาก Directory ไปยังฐานข้อมูล ดังแสดงในรูปที่ 3.5



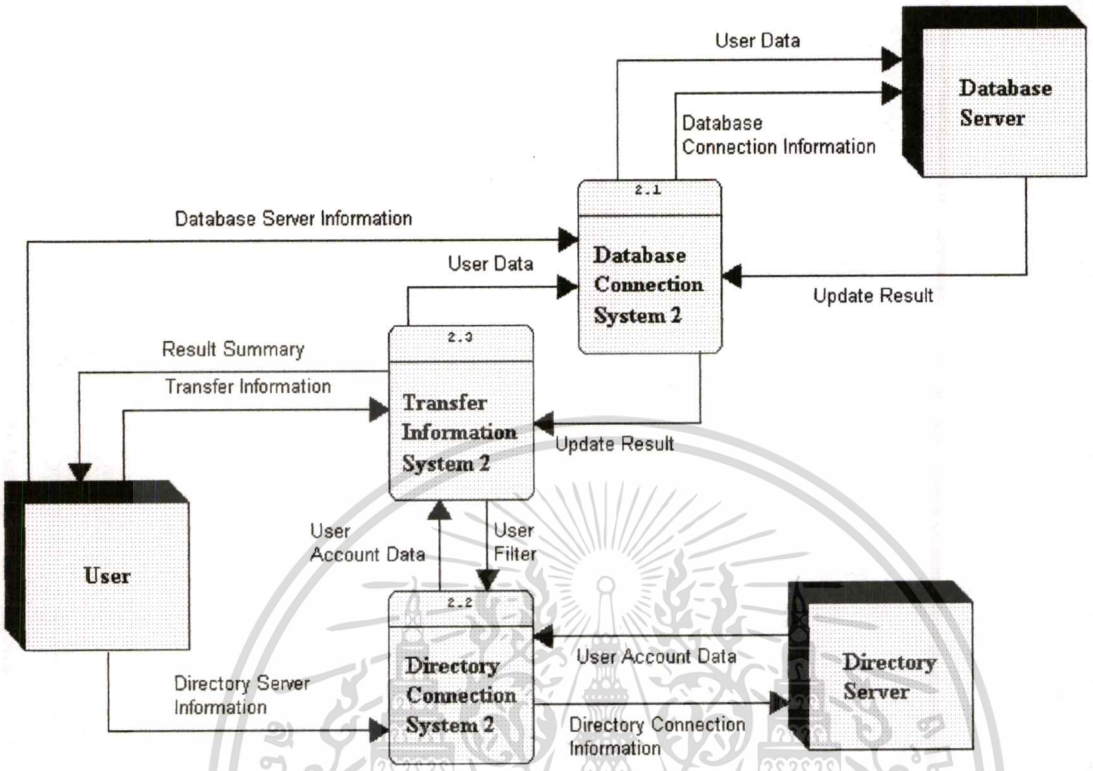
รูปที่ 3.6 แสดง Diagram 1 DFD เกี่ยวกับระบบ Import Database to LDAP System ของระบบ Database-LDAP Gateway System

ในรูปที่ 3.6 แสดงถึงระบบย่อยของระบบโอนย้ายข้อมูลจากฐานข้อมูลไปยัง Directory ซึ่งระบบภายในประกอบไปด้วยระบบ Database Connection System ทำหน้าที่ติดต่อกับฐานข้อมูลเพื่อดึงข้อมูลที่ต้องการ ระบบ Directory Connection System ทำหน้าที่ติดต่อกับ Directory เพื่อทำการ บริหารข้อมูล ซึ่งมีทั้งการเพิ่ม แก้ไขและลบข้อมูลใน Directory และส่วนสุดท้ายระบบ Transfer Information System ทำหน้าที่จัดการรวบรวมข้อมูลเพื่อใช้ในการติดต่อ และข้อมูลผู้ใช้งานที่จะทำการโอนย้ายให้สมบูรณ์ครบถ้วน



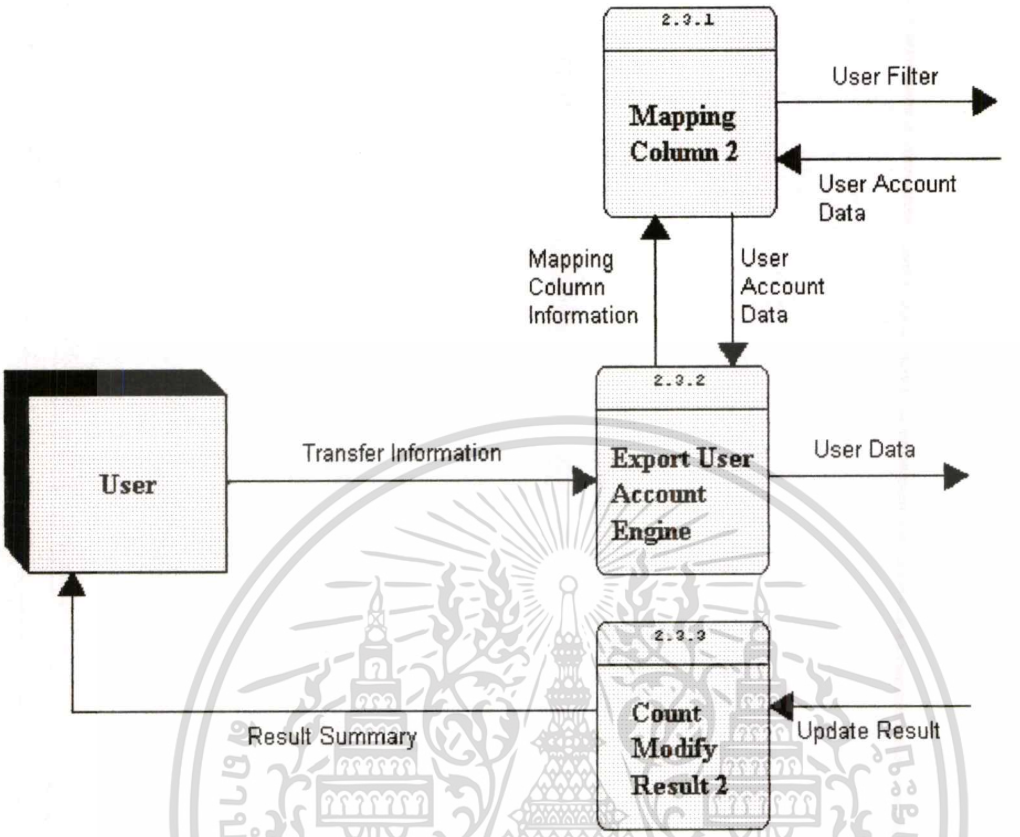
รูปที่ 3.7 แสดง Diagram 2 DFD เกี่ยวกับการทำงานของระบบ Transfer Information System 1 ของระบบ Database-LDAP Gateway System

ระบบ Transfer Information System 1 ดังแสดงในรูปที่ 3.7 ประกอบไปด้วยระบบต่างๆ เพื่อจัดการรวบรวมข้อมูลเพื่อใช้ในการติดต่อ และข้อมูลผู้ใช้งานที่จะทำการโอนย้าย รวมทั้งรวบรวมผลการทำงานที่เกิดขึ้นเพื่อแสดงสถานะของการทำงาน โดยระบบ Import User Account Engine จะรวบรวมข้อมูลที่สมบูรณ์ของบัญชีผู้ใช้งานจากระบบ Gathering User Account System และระบบ Mapping Column แล้วส่งข้อมูลไปยังระบบ Add/ Modify/ Delete User Account เพื่อส่งไปยังระบบเชื่อมต่อกับ Directory แล้วได้ผลลัพธ์ส่งไปยังระบบ Count Modify Result 1 เพื่อรวบรวมผลการทำงาน



รูปที่ 3.8 แสดง Diagram 1 DFD เกี่ยวกับระบบ Export LDAP to Database System ของระบบ Database-LDAP Gateway System

ในภาพ 3.8 แสดงถึงระบบย่อยของระบบโอนย้ายข้อมูลจาก Directory ไปยังฐานข้อมูล ซึ่งระบบภายในประกอบไปด้วยระบบ Directory Connection System ทำหน้าที่ติดต่อกับ Directory เพื่อดึงข้อมูลที่ต้องการ ระบบ Database Connection System ทำหน้าที่ติดต่อกับฐานข้อมูล เพื่อทำปรับปรุงข้อมูลในฐานข้อมูลให้ตรงกันกับ Directory และส่วนสุดท้ายระบบ Transfer Information System ทำหน้าที่จัดการรวบรวมข้อมูลเพื่อใช้ในการติดต่อ และข้อมูลผู้ใช้งานที่จะทำการโอนย้ายให้สมบูรณ์ครบถ้วน



รูปที่ 3.9 แสดง Diagram 2 DFD เกี่ยวกับการทำงานของระบบ Transfer Information System 2 ของระบบ Database-LDAP Gateway System

ระบบ Transfer Information System 2 ดังแสดงในรูปที่ 3.9 ประกอบไปด้วยระบบต่างๆ เพื่อจัดการรวบรวมข้อมูลเพื่อใช้ในการติดต่อ และข้อมูลผู้ใช้งานที่จะทำการโอนย้าย รวมทั้งรวบรวมผลการทำงานที่เกิดขึ้นเพื่อแสดงสถานะของการทำงาน โดยระบบ Export User Account Engine จะเลือกข้อมูลจะใช้ในการปรับปรุงจากระบบ Mapping Column ที่ติดต่อกับระบบเชื่อมต่อ Directory แล้วส่งข้อมูลไปยังระบบเชื่อมต่อกับฐานข้อมูล แล้วระบบ Count Modify Result 2จะรับผลลัพธ์ เพื่อรวบรวมผลการทำงาน

บทที่ 4

การพัฒนาระบบงาน

4.1 อุปกรณ์และโปรแกรมที่ใช้ในการพัฒนาระบบ

4.1.1 คอมพิวเตอร์และอุปกรณ์เครือข่าย

เนื่องจากโปรแกรมที่ทำการพัฒนามีหลักการทำงานอยู่บนพื้นฐานของ Client/Server ดังนั้น เพื่อให้มีสภาวะการทำงานที่เสมือนมีการใช้งานจริงมากที่สุด จึงได้มีการติดตั้ง Unix Server ขึ้นโดยได้นำเอา Solaris X86 มาติดตั้งบนเครื่องคอมพิวเตอร์ ทำหน้าที่เป็นเครื่องผู้ให้บริการ LDAP Directory ส่วนทางด้าน Database Server นั้นได้มีการจัดหาเครื่องคอมพิวเตอร์ อีก 1 เครื่องที่มีการทำงานบนระบบปฏิบัติการ Microsoft Window ใช้ในการพัฒนาโปรแกรมตลอดจนการจัดทำเอกสารต่าง ๆ แต่เนื่องจาก ระบบปฏิบัติการ Solaris X86 ต้องการ Hardware ที่เหมาะสมและสามารถใช้งานร่วมกันได้ ทางผู้พัฒนาโปรแกรมจึงต้องทำการจัดเตรียมเครื่องที่มีความเหมาะสมในการใช้งานดังมีรายละเอียดดังต่อไปนี้

CPU	Pentium4 1.5 GHz
Memory	256 MB
Hard-Disk	30 GB
Network-Card	3 Com 3c590
Graphic-Card	nVidia TNT 2
OS	Solaris X86
Application	Openldap Berkeley DB pam_ldap

ตารางที่ 4.1 แสดงเครื่อง LDAP Server

CPU	Pentium III 677 MHz
Memory	128 MB
Hard-Disk	20 GB
Network-Card	Novell NE2000
Graphic-Card	Sis 6326
OS	Windows 2000
Application	Oracle 8 Apache 1.3.1 PHP 4.3.1

ตารางที่ 4.2 แสดงเครื่องที่ทำหน้าที่เป็น Web Server & Database Server

Hup	Linksys 10/100 Hup 5 Ports

ตารางที่ 4.3 แสดงอุปกรณ์เครือข่าย

4.1.2 เครื่องมือที่ใช้ในการพัฒนา

ในส่วนการติดตั้งโปรแกรมอื่นๆ ที่เกี่ยวข้องโดยตรงกับการพัฒนาระบบงานนั้น ยังคงมีโปรแกรมอีกหลายตัว ในที่นี้ผู้พัฒนา ได้เลือกใช้โปรแกรมที่เป็นรุ่นที่มีเถียรภาพและใหม่ที่สุด ที่โปรแกรมตัวนั้น ๆ มีให้ดังมีรายชื่อโปรแกรมดังต่อไปนี้

Software Name	Version	Software Web Site
Berkeley DB	4.1.25	http://www.sleepycat.com
Openldap	2.1.12	http://www.openldap.org
Apache	1.3.27	http://www.apache.org
PHP	4.3.1	http://www.php.net
GNU Gcc	3.2.2	http://www.sunfreeware.com
SED	4.0	http://www.sunfreeware.com
pam_ldap	-	http://www.padl.com

ตารางที่ 4.4 แสดงรายชื่อ Software ที่ใช้ในการพัฒนาระบบงาน

ในการพัฒนาโปรแกรม ระบบปฏิบัติการที่ใช้เป็นระบบ Unix ของ บ. Sun Microsystems ที่ทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล ดังนั้นในการติดตั้งโปรแกรมสนับสนุนจึงจำเป็นที่ต้องหาโปรแกรมที่สามารถทำงานร่วมกันได้กับระบบปฏิบัติการดังกล่าว โดยจะมีวิธีการที่นิยมใช้กันอยู่ 2 วิธี ด้วยกันคือ วิธีแรก หาโปรแกรมที่ได้ถูกทำการ Compile สำหรับระบบปฏิบัติการนั้น ๆ ไว้ล่วงหน้าแล้วนำมาติดตั้งใช้งานบนเครื่องของเรา และวิธีที่สอง ทำการ Compile โปรแกรมที่ต้องการบนระบบปฏิบัติการที่ใช้งานอยู่

โปรแกรมที่จะทำหน้าที่เป็น LDAP Server จะทำงานภายใต้ระบบปฏิบัติการ Unix ซึ่งในการติดตั้งโปรแกรมนั้น เราจำเป็นต้องมีการ Compile โปรแกรมใหม่ โดยจะให้ GCC 3.2 เป็น Compiler โดยสำหรับ GCC 3.2 สำหรับ Solaris X86 นั้นเราสามารถทำการ Download ได้โดยไม่มีค่าใช้จ่ายได้จาก http://www.sunfreeware.com สำหรับการติดตั้งโปรแกรม GCC นั้นก่อนอื่นต้องทำการคลายการบีบอัดก่อนโดยใช้คำสั่ง

"gunzip gcc-3.2.2-sol8-intel-local.gz"

หลังจากนั้นจึงสามารถทำการติดตั้งโปรแกรมได้โดยใช้สิทธิ์ root ในการติดตั้ง โดยการ Run คำสั่งดังต่อไปนี้

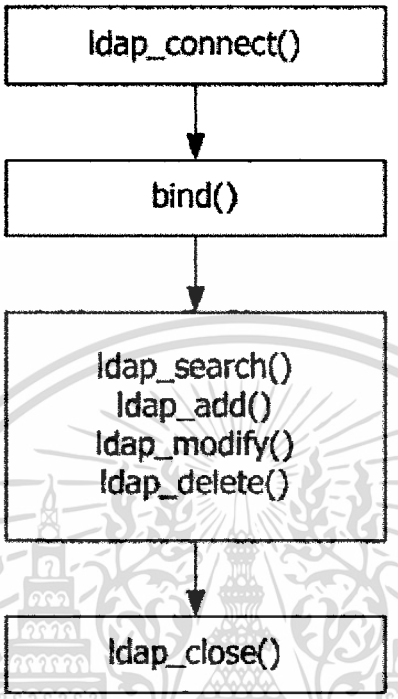
"pkgadd -d gcc-3.2.2-sol8-intel-local"

ในการติดตั้ง โปรแกรม GCC นี้ระบบจะทำการติดตั้งโปรแกรมภายใต้ Directory /usr/local นอกจากนี้ให้ทำการติดตั้งโปรแกรม sed-4.0-sol8-intel-local.gz ซึ่งสามารถ Download ได้จากที่ http://www.sunfreeware.com เหมือนกับ GCC ส่วนวิธีการติดตั้งโปรแกรมก็ใช้วิธีเดียวกัน

โปรแกรมที่เราได้ทำการติดตั้งไปก่อนหน้านี้เป็นโปรแกรมที่ใช้ในการ Compile Software ที่เราจะนำมาติดตั้งเพิ่มเติม ดังนั้นหลังจากติดตั้งโปรแกรมดังกล่าวแล้วให้ทำการกำหนด Search Path ไปยังตำแหน่งที่โปรแกรมนั้นอยู่ด้วยโดยทำการกำหนดผ่าน ตัวกำหนดค่าที่ใช้ชื่อว่า "PATH"

"PATH=/usr/local/bin:$PATH "

4.2 การเขียนโปรแกรมติดต่อกับ OpenLDAP



รูปที่ 4.1 แสดงลำดับการเรียกฟังก์ชั่นในการทำงานกับ OpenLDAP

4.2.1 การเพิ่มข้อมูล

ในการเพิ่มข้อมูลเพื่อที่จะใส่ใน Directory ต้องมีการเตรียมข้อมูลโดยเก็บไว้ใน Array แล้วจึงเรียกใช้ฟังก์ชั่น ldap_add() ดังนี้

```
// prepare data
$info["uid"]="s4067080";
$info["cn"]="Chirawat Santimitra";
    :
$info["objectclass"]="top";
$info["objectclass"]="posixAccount";
$info["objectclass"]="shadowAccount";
// add data to directory
$r=ldap_add($ds, "uid=s4067080,ou=student,dc=mydomain,dc=com ", $info);
```

4.2.2 การค้นหาข้อมูล และอ่านค่าเพื่อแสดงผล

```
$org="student";
$person="C ";
$dn = "dc=mydomain, dc=com";
$filter="((ou=$org)(cn=$person*))";
$justthese = array( "ou", "cn", "gecos");
$sr=ldap_search($ds, $dn, $filter, $justthese);
$info = ldap_get_entries($ds, $sr);
$info = ldap_get_entries($ds, $sr);
echo  "Number of entires returned is " .ldap_count_entries($ds,$sr)."<br>";
echo "Data for ".$info["count"]." items returned:";
// loop though ldap search result
for ($i=0; $i<$info["count"]; $i++) {
        echo "dn: " . $info[$i]["dn"];
        for ($ii=0; $ii<$info[$i]["count"]; $ii++) {
                echo $info[$i][$ii] . ": ";
                $attrib = $info[$i][$ii];
                eval("echo \$info[\$i][\"$attrib\"][0];");
        }
}
```

4.2.3 การเปลี่ยนแปลงข้อมูล

ในการเปลี่ยนแปลงข้อมูลใน Directory ต้องมีการเตรียมข้อมูลโดยเก็บไว้ใน Array แล้วจึงเรียกใช้ฟังก์ชั่น ldap_modify() ดังนี้

```
// prepare data
$info["uid"]="s4067080";
$info["cn"]="Chirawat Santimitra";
    :
$info["objectclass"]="top";
$info["objectclass"]="account";
```

```
$info["objectclass"]="posixAccount";
$info["objectclass"]="shadowAccount";
// modify data to directory
$r=ldap_modify($ds, "uid=s4067080,ou=student,dc=mydomain,dc=com ", $info);
```

4.2.4 การลบข้อมูล

```
function myldap_delete($ds,$dn,$recursive=false){
        if($recursive == false){
                return(ldap_delete($ds,$dn));
        }else{
                //searching for sub entries
                $sr=ldap_list($ds,$dn,"ObjectClass=*",array(""));
                $info = ldap_get_entries($ds, $sr);
                for($i=0;$i<$info['count'];$i++){
                        //deleting recursively sub entries
                        $result=myldap_delete($ds,$info[$i]['dn'],$recursive);
                        if(!$result){
                                //return result code, if delete fails
                                return($result);
                        }
                }
                return(ldap_delete($ds,$dn));
        }
}
```

4.3 การเขียนโปรแกรมติดต่อกับ Oracle

ในการติดต่อกับ Oracle จะต้องมี Oracle8 client libraries และก่อนที่จะใช้งานได้จะต้องมีการกำหนด Oracle environment variables สำหรับ Oracle user ซึ่งได้แก่

ORACLE_HOME

ORACLE_SID

LD_PRELOAD

LD_LIBRARY_PATH

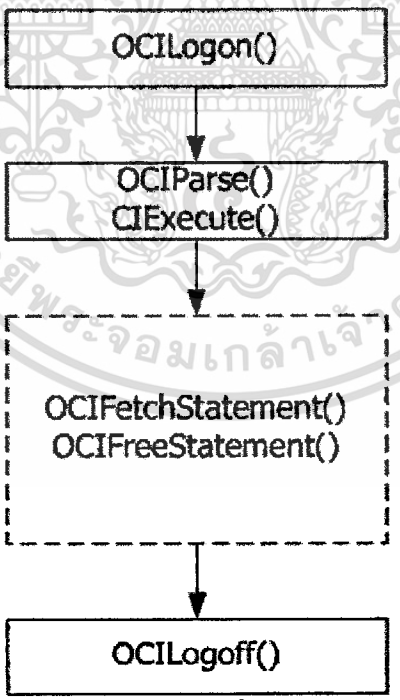
NLS_LANG

ORA_NLS33

หลังจากมีการกำหนดค่าแล้วจะต้องมีการเพิ่ม user ใน oracle group เพื่อให้สามารถเชื่อต่อโดยใช้ user นี้

ในการ compile PHP บน Unix จะใช้ option เป็น --with-oci8[=DIR] โดยที่ DIR เป็น defaults ของ environmment variable ของ ORACLE_HOME.

ถ้าเป็น PHP บน Window จะต้องมีการโหลด Extension ที่เกี่ยวข้องกับระบบที่จะติดต่อดังนี้ php_dbx.dll, php_oci8.dll



รูปที่ 4.2 แสดงลำดับการเรียกฟังก์ชั่นในการติดต่อกับ Oracle

ส่วนการเขียนโปรแกรมเพื่อติดต่อกับข้อมูลบน Oracle จะมีลำดับการเรียกฟังก์ชั่น เริ่มตั้งแต่การเชื่อมต่อกับเซิร์ฟเวอร์ ด้วย OCILogon() ซึ่งต้องกำหนดค่าเกี่ยวกับที่ของและฐานข้อมูลที่จะทำการเรียก จากนั้นจึงใช้ SQL เพื่อดึงข้อมูลจาก จากเรียกฟังก์ชั้น CIExecute() แล้วแสดงผลลัพธ์ด้วย OCIFetchStatemnet() และ OCIFreeStatement() จากนั้นก็ปิดการเชื่อมต่อด้วย OCILogoff()

```
$db = "testdb";
$conn = OCILogon("ldapuser","ldappassword",$db);
$stmt = OCIParse($conn,"select studentid,studentname,studentpassword from student");
CIExecute($stmt);
$nrows = OCIFetchStatement($stmt,$results);
if ( $nrows > 0 ) {
        while ( list( $key, $val ) = each( $results ) ) {
                $key
        }
        for ( $i = 0; $i < $nrows; $i++ ) {
                reset($results);
                while ( $column = each($results) ) {
                        $data = $column['value'];
                        $data[$i]
                }
        }
}
OCIFreeStatement($stmt);
OCILogoff($conn);
```

บทที่ 5

ผลการทดสอบและข้อเสนอแนะ

5.1 การทดสอบการทำงาน

5.1.1 เครื่องมือที่ใช้ในการทดสอบ

โปรแกรมที่ใช้ในการตรวจสอบผลการทำงาน ระบบที่พัฒนาขึ้นทำงานให้ผลลัพธ์ถูกต้องมีดังต่อไปนี้

- SQL Plus เพื่อใช้สร้างและตรวจสอบข้อมูลในฐานข้อมูล Oracle ซึ่งเป็นเครื่องมือที่ให้มากับฐานข้อมูล Oracle อยู่แล้ว
- LDAP Browser/Editor version 2.8.1 เป็นเครื่องมืออีกตัวหนึ่ง เพื่อใช้ตรวจสอบข้อมูลในระบบ Directory สามารถโหลดได้จาก http://www.iit.edu/~gawojar/ldap/

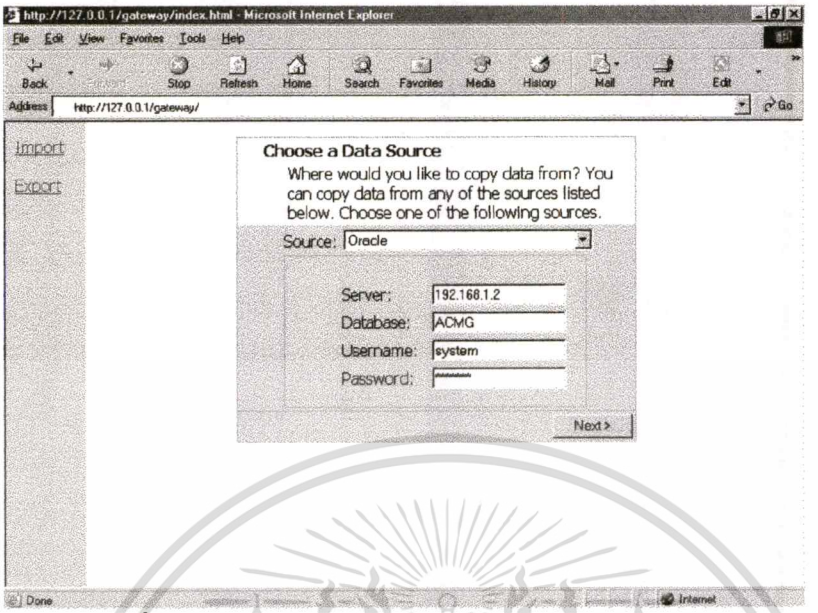
5.1.2 ขั้นตอนการทดสอบ

ในการทดสอบจะต้องทำการสร้างข้อมูลในฐานข้อมูล Oracle เพื่อเป็นข้อมูลที่ใช้ในการโอนย้าย โดยสร้างฐานข้อมูลชื่อ ACMG และสร้างตารางชื่อ useraccounts ที่มีโครงสร้างตามตารางที่ 5.1 แล้วใส่ข้อมูลของผู้ใช้งานจำนวนหนึ่งในฐานข้อมูล

Field name	Field type	
user_id	Number(8)	Primary key
user_name	Varchar2(50)	
user_password	Varchar2(50)	

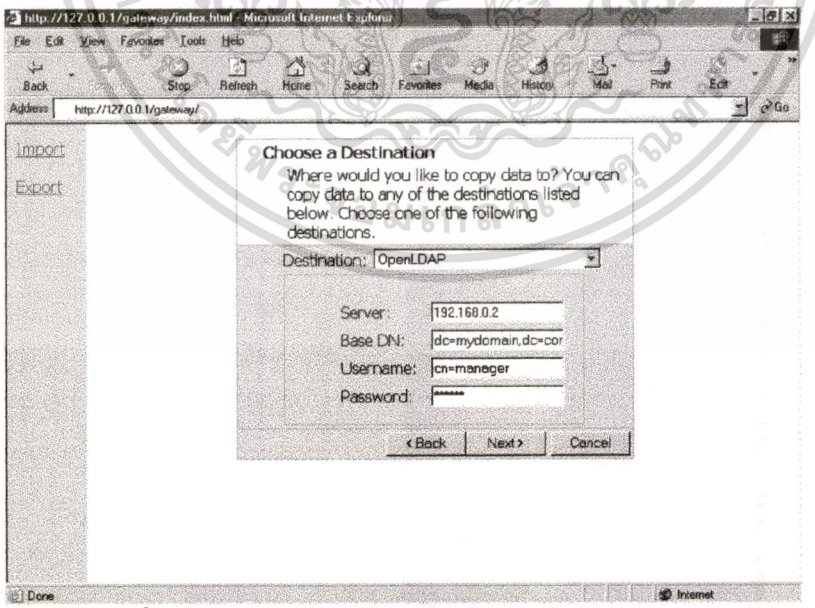
ตารางที่ 5.1 แสดงโครงสร้างของตาราง useraccounts ในฐานข้อมูล Oracle

ในการทดสอบโปรแกรม ซึ่งโปรแกรมมีการทำงานออกเป็นสองส่วน คือ ส่วน Import เพื่อทำการโอนย้ายข้อมูลจากฐานข้อมูลไปยัง Directory และส่วน Export เพื่อทำการดึงข้อมูลบน Directory เพื่อไปปรับปรุงข้อมูลในฐานข้อมูล เราจะทำการสอบในส่วนแรกก่อนทดลองย้ายข้อมูลจากฐานข้อมูลเข้าไปสู่ Directory

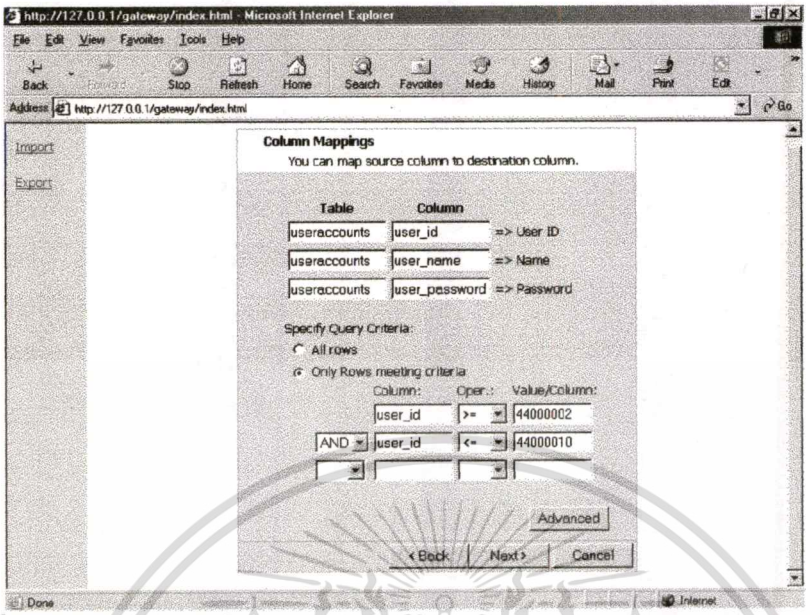


รูปที่ 5.1 แสดงหน้าจอใส่ข้อมูลเพื่อติดต่อกับฐานข้อมูล

ทดสอบการทำงานโดยเรียกใช้โปรแกรมผ่านทาง Browser เลือกส่วนการ Import จากเมนูทางซ้ายมือ ใส่ข้อมูลเพื่อใช้ในการติดต่อกับฐานข้อมูลในหน้าจอที่หนึ่งดังแสดงในรูปที่ 5.1 และใส่ข้อมูลเกี่ยวกับ Directoryในหน้าจอที่สองดังแสดงในรูปที่ 5.2



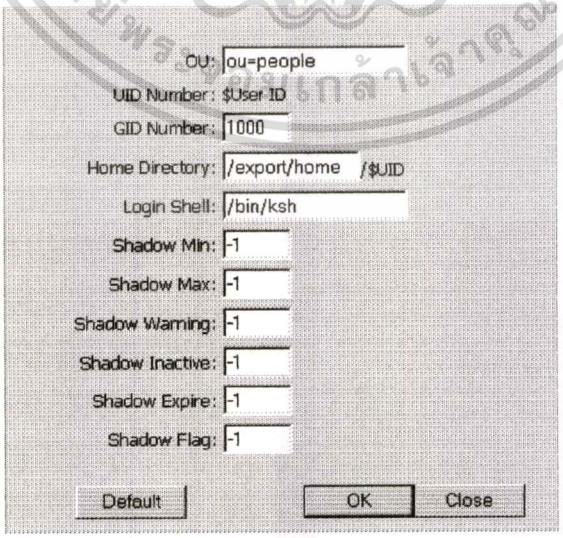
รูปที่ 5.2 แสดงหน้าจอใส่ข้อมูลเพื่อติดต่อกับ Directory



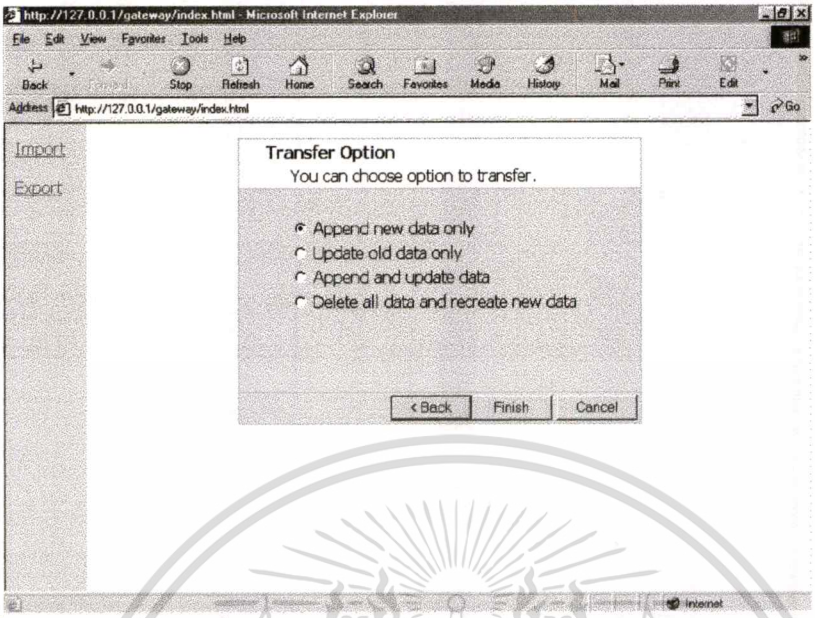
รูปที่ 5.3 แสดงหน้าจอเพื่อกำหนดข้อมูลที่จะโอนย้ายจากฐานข้อมูลไป Directory

ในหน้าจอที่สาม จะทำการกำหนดข้อมูลที่จะเอามาจากฐานข้อมูล เพื่อนำไปใช้เป็นข้อมูลบัญชีผู้ใช้งาน ซึ่งมีข้อมูลที่จำเป็นสามตัวคือ User Id, Name, Password ที่จะทำการดึงจากฐานข้อมูล และส่วนล่างจะเป็นการกำหนดกลุ่มของข้อมูลที่ต้องการ โดยมีลักษณะการใช้งานเหมือนภาษา SQL

ส่วนข้อมูลอื่นๆที่จำเป็นเพื่อให้ข้อมูลของบัญชีผู้ใช้งานมีความสมบูรณ์สามารถกำหนดได้จากหน้าจอโดยกดปุ่ม Advanced ด้านล่างดังแสดงหน้าจอในรูปที่ 5.4



รูปที่ 5.4 แสดงหน้าจอเพื่อกำหนดค่าเพิ่มเติมของบัญชีผู้ใช้งาน

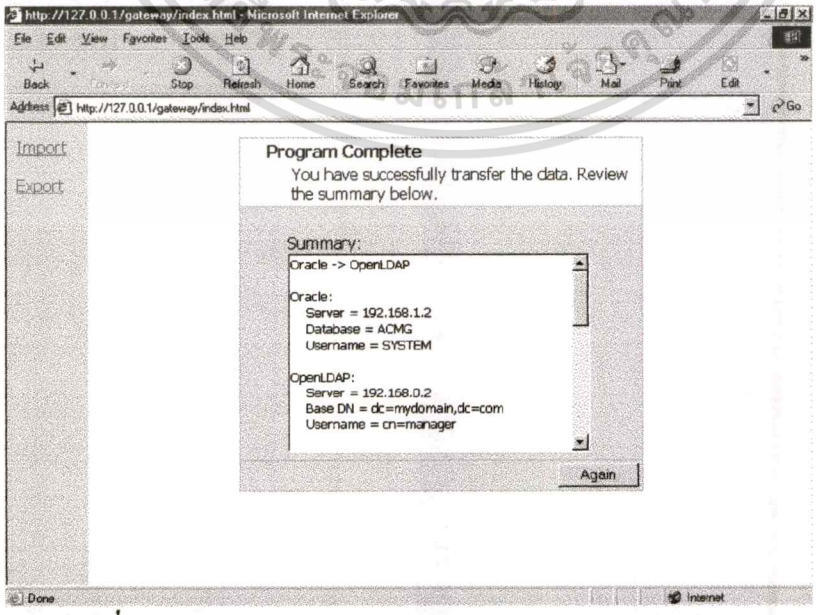


รูปที่ 5.5 แสดงหน้าจอเพื่อกำหนดลักษณะการโอนย้ายจากฐานข้อมูลไป Directory

ในหน้าต่อไปจะทำการเลือกลักษณะการย้ายข้อมูล ประกอบด้วย

- Append new data only คือ ทำการเพิ่มข้อมูลที่ไม่ซ้ำกับข้อมูลเดิมโดยดูจาก UID
- Update old data only คือ ทำการปรับปรุงข้อมูลเดิมโดยดูจาก UID
- Append and update data คือ ทำการเพิ่มข้อมูลที่ไม่ซ้ำและปรับปรุงข้อมูลเดิมโดยดูจากUID
- Delete all data and recreate new data คือลบข้อมูลเดิมออกทั้งหมดแล้วใส่ข้อมูลใหม่เข้าไป

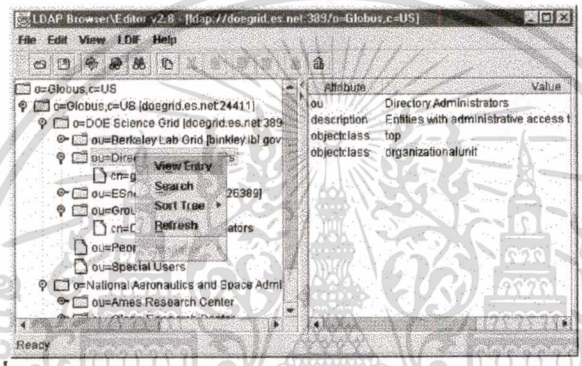
และผลลัพธ์ในการทำงานจะแสดงในหน้าต่อไปดังแสดงในรูปที่ 5.6



รูปที่ 5.6 แสดงผลการย้ายข้อมูลจากการทำงานของโปรแกรม

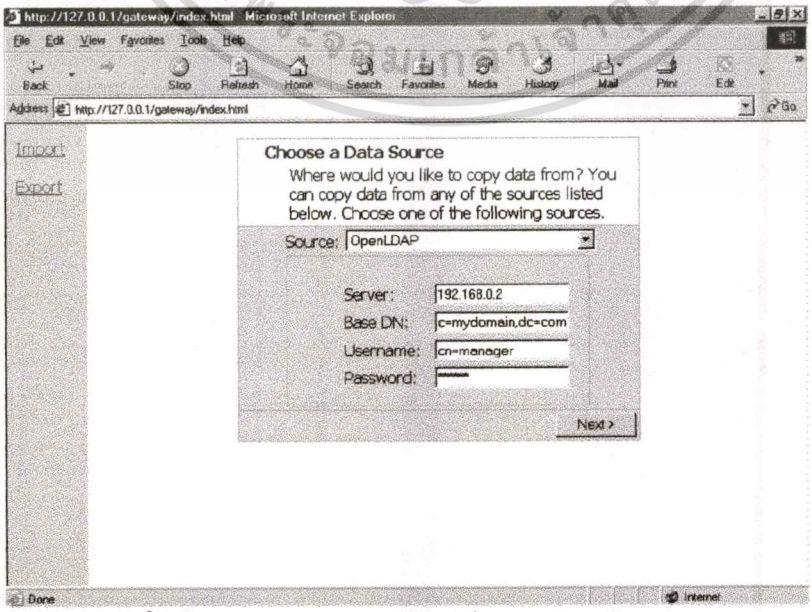
เราสามารถตรวจสอบการทำงานโดยใช้โปรแกรม LDAP Browser/Editor เพื่อตรวจสอบผลการทำงานของระบบ โดยใช้โปรแกรม LDAP Browser/Editor เพื่อแสดงผลการเปลี่ยนแปลงที่เกิดขึ้นใน Directory จากนั้นทำการเปลี่ยนตัวเลือกหรือกำหนดค่าต่างๆ ให้กับระบบ แล้วตรวจสอบผลการทำงานจากโปรแกรม LDAP Browser/Editor เพื่อเป็นการยืนยันการทำงานของระบบว่ามีความถูกต้องของระบบในส่วน Import

ต่อไปเราจะทำการทดสอบการใช้งานในส่วน Export เพื่อทำการดึงข้อมูลบน Directory เพื่อไปปรับปรุงข้อมูลในฐานข้อมูล

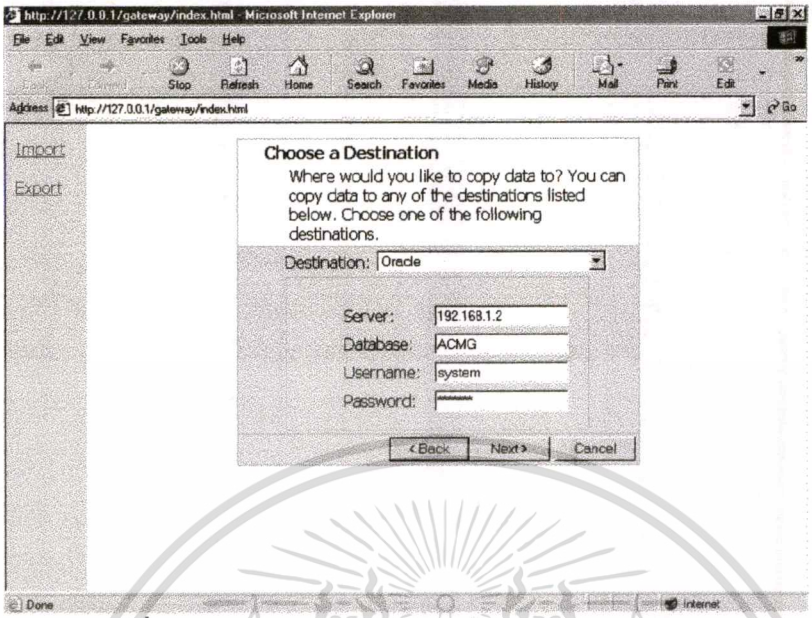


รูปที่ 5.7 แสดงหน้าจอของโปรแกรม LDAP Browser/Editor

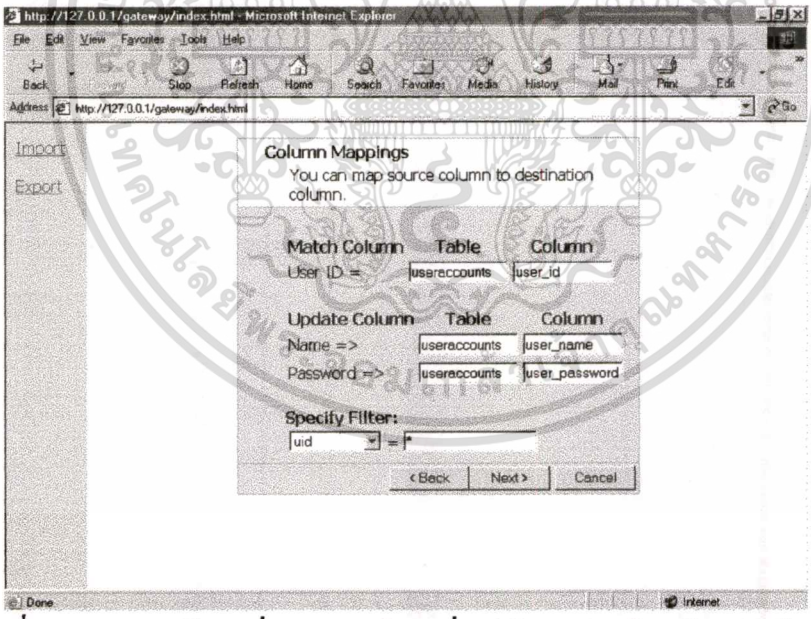
เพื่อที่จะทดสอบทำงานของระบบส่วน Export จาก Browser ให้เลือกส่วนการ Export จากเมนูทางซ้ายมือ ใส่ข้อมูลเพื่อใช้ในการติดต่อกับฐานข้อมูลดังแสดงในรูปที่ 5.8 แล้วใส่ข้อมูลเกี่ยวกับ Directory ในหน้าจอดังแสดงในรูปที่ 5.9



รูปที่ 5.8 แสดงหน้าจอใส่ข้อมูลเพื่อติดต่อกับ Directory

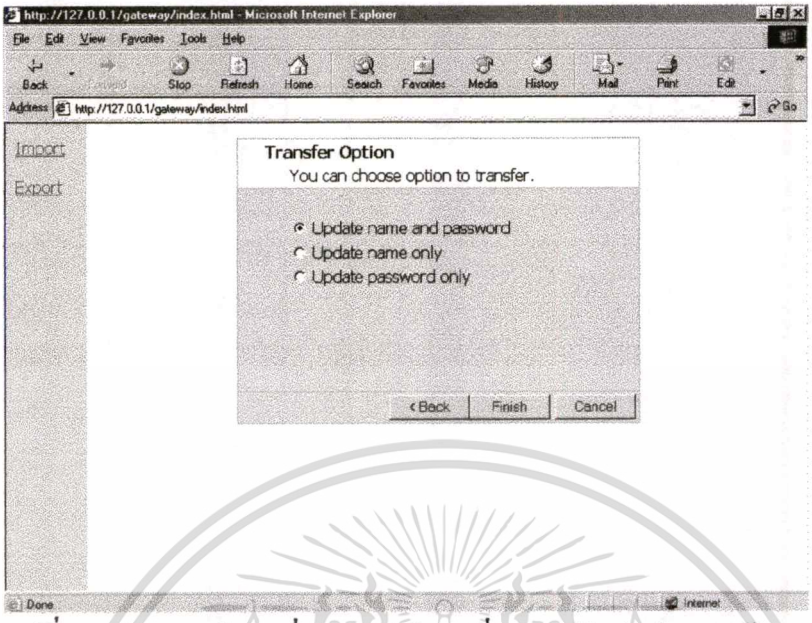


รูปที่ 5.9 แสดงหน้าจอใส่ข้อมูลเพื่อติดต่อกับฐานข้อมูล



รูปที่ 5.10 แสดงหน้าจอเพื่อกำหนดข้อมูลที่จะใช้ปรับปรุงข้อมูลในฐานข้อมูล

ในรูปที่ 5.10 แสดงหน้าจอที่ใช้ทำการกำหนดข้อมูลที่จะถูกปรับปรุงในฐานข้อมูล เพื่อ ซึ่งสามารถถูกปรับปรุงได้จากข้อมูลใน Directory สามตัวคือ User Id, Name, Password และส่วนล่างจะเป็นการกำหนดกลุ่มของข้อมูลที่ต้องการ โดยมีลักษณะการใช้งานเหมือน Search ใน Attribute ที่ต้องการ

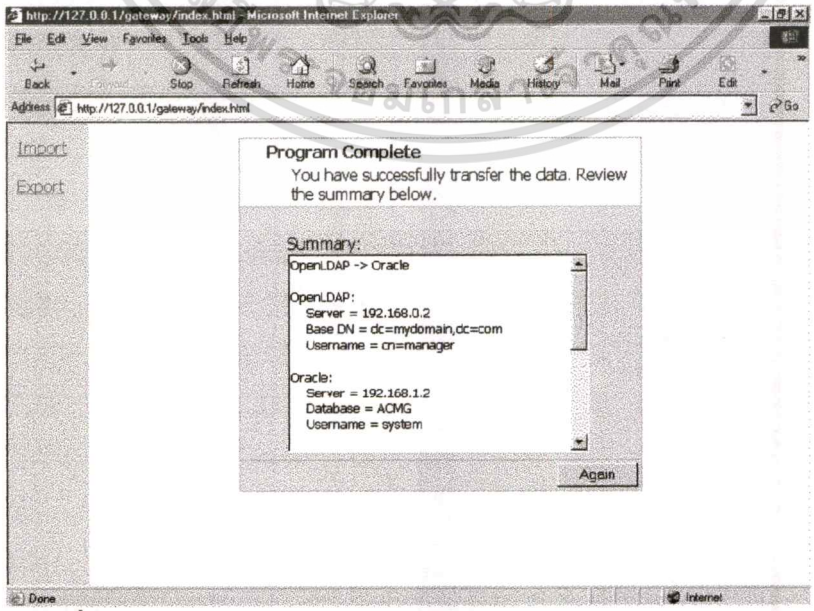


รูปที่ 5.11 แสดงหน้าจอเพื่อกำหนดข้อมูลที่จะถูกปรับปรุงในฐานข้อมูล

ในหน้าต่อไปจะทำการเลือกลักษณะการย้ายข้อมูล ประกอบด้วย

- Update name and password คือ ทำการปรับปรุงข้อมูลในฐานข้อมูลทั้งชื่อและรหัส
- Update name only คือ ทำการปรับปรุงข้อมูลในฐานข้อมูลเฉพาะชื่อ
- Update password only คือ ทำการปรับปรุงข้อมูลในฐานข้อมูลเฉพาะรหัส

และผลลัพธ์ในการทำงานจะแสดงในหน้าต่อไปดังแสดงในรูปที่ 5.12



รูปที่ 5.12 แสดงผลการปรับปรุงข้อมูลจากการทำงานของโปรแกรม

เราสามารถตรวจสอบการทำงานโดยใช้โปรแกรม SQL Plus เพื่อตรวจสอบผลการทำงานของระบบ โดยใช้โปรแกรม SQL Plus เพื่อแสดงผลการเปลี่ยนแปลงที่เกิดขึ้นในฐานข้อมูล Oracle จากนั้นทำการเปลี่ยนตัวเลือกหรือกำหนดค่าต่างๆ ให้กับระบบ หรือใช้โปรแกรม LDAP Browser/Editor เพื่อแก้ไขค่าใน Directory แล้วตรวจสอบผลการทำงานจากโปรแกรม SQL Plus โดยแสดงค่าที่อยู่ในฐานข้อมูล Oracle เพื่อเป็นการยืนยันการทำงานของระบบว่ามีความถูกต้องของระบบในส่วน Export

5.1.3 ผลจากการทดสอบ

จากการทดสอบการใช้งานโปรแกรมตามขั้นตอนที่กำหนดได้ผลลัพธ์ดังนี้

- สามารถทำการดึงข้อมูลจากฐานข้อมูล Oracle เข้าไปในระบบ Directory Service ได้อย่างถูกต้อง ตามตัวเลือกทั้งหมดที่มี และตามค่าที่กำหนดเพิ่มเติมในส่วนของ User Account
- สามารถทำการปรับปรุงข้อมูลในฐานข้อมูล Oracle จากระบบ Directory Service ได้อย่างถูกต้อง ตามตัวเลือกทั้งหมดที่มี
- สามารถให้ผู้ใช้งานที่ถูกเพิ่มเข้าไปใน Directory ใช้งานระบบ Unix โดยใช้การรับรองสิทธิของผู้ใช้งานผ่านทาง LDAP Directory ได้ แต่มีปัญหาของไม่มี Home Directory ของผู้ใช้ เนื่องไม่มีการสร้างไว้ในระบบ ทำให้ผู้ใช้งานไม่สามารถใช้งานระบบได้

5.2 ข้อเสนอแนะ

- ระบบที่พัฒนาควรจะเพิ่มความหยืดหยุ่นในส่วนของ User ID ให้มากขึ้น สามารถกำหนดรูปแบบให้เหมาะกับการใช้งานได้ เช่นมีการเพิ่มตัวอักษร หรือตัวคำบางส่วนมาจากชื่อของผู้ใช้งาน
- ในการใช้งานจริงข้อมูลอยู่ในฐานข้อมูลเดิมอาจมีบางส่วนไม่สมบูรณ์ ทำให้โปรแกรมไม่สามารถทำงานได้ ซึ่งควรมีการพัฒนาและเพิ่มการตรวจสอบในส่วนนี้ของโปรแกรมในสามารถใช้งานได้แม้ข้อมูลจะไม่สมบูรณ์
- ระบบที่พัฒนาขึ้นมีการใช้งานอาจจะพัฒนาให้มี GUI ที่ดีขึ้นมีการเก็บข้อมูลไว้ใช้งานในภายหลังได้ หรือมีค่าที่มีการใช้ประจำเพื่อให้สามารถทำงานได้สะดวกรวดเร็วยิ่งขึ้น
- ค่าที่กำหนดบางส่วนอาจยังคงมีความยืดหยุ่นน้อยเกินไป และมีการทำงานที่ยังไม่มีประสิทธิภาพที่ดีนัก ซึ่งสามารถปรับปรุงให้ดีเพิ่มขึ้นได้

บรรณานุกรม

สำนักวิจัยและบริการคอมพิวเตอร์. 2002. **Computer Research Service Center, KMITL** [Online]. Available: http://www.kmitl.ac.th/crsc/index.html

Howard, L. 1998. **Request for Comments 2307.** [Online]. Available: http://www.ietf.org/rfc/rfc2307.txt

Howes, T. and Smith, M. 1997. **LDAP Programming Directory-Enabled Applications with Lightweight Directory Access Protocol.** Indianapolis. Macmillan Technical.

Timothy, A. et al. 1999. **Understanding and Deploying LDAP Directory Sercices.** USA.Macmilan Computer.

ภาคผนวก

ภาคผนวก ก

รายละเอียดการติดตั้ง Program

ขั้นตอนการติดตั้งโปรแกรม Berkelay DB

1. export CC=gcc
2. cd /root/src/db-4.1.25.NC/build_unix
3. ../dist/configure --prefix=/opt/db_4.1.25.NC
4. make
5. make install

ขั้นตอนการติดตั้งโปรแกรม Openldap

1. export CC=gcc
2. export CPPFLAGS="-I/opt/db_4.1.25.NC/include"
3. export LDFLAGS="-L/opt/db_4.1.25.NC/lib"
4. cd /root/src/openldap-2.1.12
5. ./configure --prefix=/opt/openldap_2.1.12
6. make depend
7. make
8. make install
9. ทำการกำหนดค่าต่าง ๆ ใน File /opt/openldap_2.1.12/etc/openldap/slapd.conf

```
include     /opt/openldap_2.1.12/etc/openldap/schema/core.schema
include     /opt/openldap_2.1.12/etc/openldap/schema/cosine.schema
include     /opt/openldap_2.1.12/etc/openldap/schema/nis.schema
pidfile     /opt/openldap_2.1.12/var/slapd.pid
argsfile    /opt/openldap_2.1.12/var/slapd.args
backend     bdb
database    bdb
suffix      "dc=mydomain,dc=com"
rootdn      "cn=Manager,dc=mydomain,dc=com"
rootpw      secret
directory   /opt/openldap_2.1.12/var/openldap-data
index objectClass eq
```

ขั้นตอนการติดตั้งโปรแกรม APACHE

1. /root/src/apache_1.3.27
2. ./configure --prefix=/opt/apache_1.3.27 --enable-module=so
3. make
4. make install
5. ถ้าเป็นบนให้รัน Window apache_1.3.27-win32-x86-no_src.exe โปรแกรมจะติดตั้งโดยอัตโนมัติ
6. แก้ไข File httpd.conf

ServerType standalone

ServerRoot "/opt/apache_1.3.27"

PidFile /opt/apache_1.3.27/logs/httpd.pid

ScoreBoardFile /opt/apache_1.3.27/logs/httpd.scoreboard

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15

MinSpareServers 5

MaxSpareServers 10

StartServers 5

MaxClients 150

MaxRequestsPerChild 0

LoadModule php4_module libexec/libphp4.so

Port 80

User nobody

Group nobody

ServerAdmin root@proxy.tp.co.th

DocumentRoot "/opt/web"

```
<Directory "/opt/web">
   Options Indexes FollowSymLinks MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

<IfModule mod_dir.c>
   DirectoryIndex index.php index.html
</IfModule>
AddType application/x-httpd-php .php
HostnameLookups Off
ErrorLog /opt/apache_1.3.27/logs/error_log
```

ขั้นตอนการติดตั้งโปรแกรม PHP

1. ./configure --prefix=/opt/php_4.3.1 --with-apxs=/opt/apache_1.3.27/bin/apxs --with-ldap=/opt/openldap_2.1.12 --with-oci8=/opt/oracle
2. make
3. make install
4. สำหรับ Window ให้แตกไฟล์ php-4.3.1-Win32.zip ที่ c:\php
5. คัดลอก php.ini-dist ไปไว้ใน c:\winnt แล้วเปลี่ยนชื่อเป็น php.ini
6. แก้ไข File php.ini

```
extension_dir = c:\php
extension=php_dbx.dll
extension=php_ldap.dll
extension=php_oci8.dll
```

ขั้นตอนการติดตั้งโปรแกรม Pam_ldap

1. ./configure -prefix=/opt/pam_ldap_161 --with_ldap_lib=openldap
2. make
3. make install

ประวัติผู้เขียน

ชื่อผู้เขียน	นายจิรวัฒน์ สันติมิตร
วันเกิด	28 พฤษภาคม 2521
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตร์บัณฑิต (ไฟฟ้า)
สถานที่สำเร็จการศึกษา	คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษาที่สำเร็จการศึกษา	2541