การพิสูจน์ตนด้วยชีวมาตร

BIOMETRIC AUTHENTICATION

นายณรงค์ฤทธิ์ อินทสันตา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

BIOMETRIC AUTHENTICATION

BY

MR.NARONGRIT INTASUNTA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIRED FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEER

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

หัวข้อปริญญานิพนธ์ การพิสูจน์ตนด้วยชีวมาตร

Biometric Authentication

จัดทำโดย นายณรงค์ฤทธิ์ อินทสันตา เลขประจำตัว 45010218

อาจารย์ที่ปรึกษา ดร.พิทักษ์ ธรรมวาริน

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

(ดร.พิทักษ์ ธรรมวาริน)

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์ การพิสูจน์ตนด้วยชีวมาตร

Biometric Authentication

จัดทำโดย นายณรงค์ฤทธิ์ อินทสันตา เลขประจำตัว 45010218

อาจารย์ที่ปรึกษา ดร.พิทักษ์ ธรรมวาริน

บทคัดย่อ

ในปัจจุบันนี้ระบบปฏิบัติการลินุกซ์เป็นระบบปฏิบัติการหนึ่งที่มีการใช้งานอย่างแพร่หลาย อีกทั้งยังมีการปรับปรุงประสิทธิภาพมาโดยตลอด เพื่อเพิ่มประสิทธิภาพของระบบปฏิบัติการ ดังนั้นในปริญญานิพนธ์เล่มนี้จึงได้นำเสนอการปรับปรุงวิธีการพิสูจน์ตนเพื่อเข้าใช้ระบบของผู้ใช้บนระบบปฏิบัติการลินุกซ์ จากเดิมที่ใช้การป้อนยูสเซอร์เนมและรหัสผ่าน ซึ่งมักเกิดปัญหามีผู้อื่นล่วงรู้รหัสผ่าน หรือผู้ใช้เองลืมรหัสผ่าน มาเป็นการใช้ใบหน้าแทนการใช้รหัสผ่านเดิม เพื่อเพิ่มความปลอดภัยให้มากขึ้น อีกทั้งเพื่อความสะดวกในการเข้าใช้งานของผู้ใช้งานในระบบอีกด้วย

Thesis Title	Biometric Authentication
Student	Mr.Narongrit Intasunta ID. 45010218
Advisor	Prof. Pitak Thumwarin
Graduate Level	Bachelor Degree of Information Engineering
Department	Information engineering
Academic Year	2005

ABSTRACT

At this moment, using Linux operating system is more popular. For system security, user authentication is needed in order to access into system.

The authentication is mainly use username and password. But username and password are insufficient to secure the system, because of using password may be guessed or may be known by others and sometimes users forget their password. Therefore in order to overcome the above problems, we developed the Linux login system by using face recognition.

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ เนื่องด้วยคำแนะนำ สนับสนุน และคำปรึกษาจากอาจารย์ที่ปรึกษา ดร.พิทักษ์ ธรรมวาริน ที่คอยแนะนำ และเอาใจใส่กับการทำโครงงานนี้เป็นอย่างดี ซึ่งต้องขอขอบพระคุณเป็นอย่างสูงยิ่ง

นอกเหนือจากนี้ ขอขอบพระคุณคณาจารย์ทุกท่าน ในภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นอย่างยิ่งที่ได้ช่วยประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้จัดทำโครงงาน ทำให้ผู้จัดทำมีความรู้ ความสามารถในการจัดทำปริญญานิพนธ์ฉบับนี้

ขอขอบคุณเพื่อน ๆ พี่ ๆ และน้อง ๆ และคนอื่น ๆ ที่ไม่ได้กล่าวถึง ซึ่งได้ช่วยเหลือในการทำงานและแก้ไขปัญหา อุปสรรคต่าง ๆ ให้ผ่านพ้นไปได้ด้วยดี

สุดท้ายนี้ ต้องขอขอบพระคุณบุคคลที่สำคัญเป็นที่สุด คือ บิดา มารดาที่เคารพและเป็นที่รักยิ่ง ผู้ที่ให้กำเนิด คอยสั่งสอน ดูแล ให้การศึกษา พร้อมทั้งสนับสนุนกิจกรรมต่าง ๆ นับเป็นพระคุณอย่างสูงสุดหาที่เปรียบมิได้ ผู้จัดทำขอระลึกพระคุณอันยิ่งใหญ่สุดประมาณนี้ไว้กว่าชีวิตจะหาไม่ และกราบขอบพระคุณทุกท่านไว้ ณ ที่นี้ด้วย

ณรงค์ฤทธิ์ อินทสันตา

สารบัญ

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันการใช้งานระบบปฏิบัติการลินุกซ์ได้รับความนิยมเพิ่มมากขึ้นอย่างกว้างขวางทั้งในระดับผู้ใช้งานทั่วไป หรือองค์กร สำหรับการเข้าใช้งานบนระบบปฏิบัติการลินุกซ์นั้นก่อนที่จะสามารถเข้าใช้งานในระบบได้ จำเป็นต้องมีการตรวจสอบและยืนยันผู้เข้าใช้งานก่อน ทั้งนี้เพื่อความปลอดภัยต่อระบบปฏิบัติการ ความปลอดภัยต่อข้อมูลภายใน และเป็นการกำหนดสิทธิการเข้าใช้งานของผู้ใช้งานในระดับต่าง ๆ

วิธีการในการตรวจสอบผู้ใช้และยืนยันสิทธิผู้ใช้ โดยปกติแล้วจะนิยมใช้ชื่อบัญชีผู้ใช้และรหัสผ่านเป็นสำคัญ แต่การใช้วิธีการป้อนข้อมูลบัญชีผู้ใช้และรหัสผ่าน อาจไม่เพียงพอต่อความปลอดภัยของระบบ เพราะผู้ใช้อาจลืมรหัสผ่าน ทำรหัสผ่านหาย หรืออาจมีผู้อื่นล่วงรู้รหัสผ่านนั้นได้ ฉะนั้นวิธีการที่นำเอาระบบรักษาความปลอดภัยด้วยการใช้การรู้จำใบหน้า (Face recognition) ผู้ใช้ เพื่อใช้ในการตรวจสอบและยืนยันผู้เข้าใช้งานบนระบบปฏิบัติการลินุกซ์ แทนการใช้รหัสผ่านจึงเป็นอีกวิธีการที่มีความปลอดภัยมากกว่า เนื่องจากใบหน้าแต่ละคน จะแตกต่างกันออกไป อีกทั้งยังเป็นการยากต่อการลอกเลียนแบบได้

ด้วยเหตุผลดังกล่าว ในปริญญานิพนธ์ฉบับนี้กล่าวถึงระบบรักษาความปลอดภัยด้วยการรู้จำใบหน้ามาใช้ในการตรวจสอบและยืนยันผู้เข้าใช้งานในระบบปฏิบัติการลินุกซ์แทนการใช้รหัส-ผ่าน โดยเมื่อผู้ใช้ต้องการล็อกอินเข้าในบัญชีผู้ใช้ (user account) ของตน สามารถทำได้โดยการสแกนใบหน้า ถ้าการตรวจสอบถูกต้อง ผู้ใช้จึงจะสามารถเข้าใช้งานระบบผ่านบัญชีผู้ใช้ของตนได้ตามปกติ แต่ถ้ามีความผิดพลาดเกิดขึ้น จะไม่สามารถใช้งานได้ จะต้องทำการล็อกอินใหม่อีกครั้ง

สำหรับกระบวนการการรู้จำใบหน้านั้น ในปริญญานิพนธ์ฉบับนี้ได้นำเฟรมเวิร์ค OpenCV [4] ซึ่งเป็นเฟรมเวิร์คในการพัฒนาด้านกระบวนการวิเคราะห์ภาพ ซึ่งจะกล่าวถึงในบทถัดไป

สำหรับกระบวนการตรวจสอบการเข้าใช้งานในระบบปฏิบัติการลินุกซ์นั้น ระบบต้นแบบจะทำงานร่วมกับ PAM API (Pluggable Authentication Modules Application Programming Interface) [5] ซึ่งเป็นมอดูลที่สนับสนุนการพิสูจน์ตนของผู้ใช้งานในระบบปฏิบัติการลินุกซ์ ซึ่งสนับสนุนระบบการพิสูจน์ตนหลาย ๆ ระบบ สนับสนุนกลไกและนโยบายที่เฉพาะเจาะจงในแต่ละแอพพลิเคชันได้ ในการทำงานร่วมกับ PAM นี้จะทำการพัฒนาโปรแกรมในส่วนของ pam_anubis ขึ้นมาใหม่ เพื่อรองรับระบบการล็อกอินด้วยใบหน้า ในระบบปฏิบัติการลินุกซ์ได้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อมุ่งเน้นเรื่องความปลอดภัยในการเข้าใช้งานในระบบปฏิบัติการลินุกซ์

1.2.2 เพื่อพัฒนาต้นแบบในการพิสูจน์ตนเข้าสู่ระบบ ด้วยวิธีการทางชีวมาตร

1.2.3 เพื่อพัฒนาวิธีการพิสูจน์ตนทางชีวมาตร (Biometrics) ให้ใช้งานร่วมกับระบบปฏิบัติการลินุกซ์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ได้รับความรู้ ความเข้าใจเกี่ยวกับกระบวนพิสูจน์ตนบนระบบปฏิบัติการลินุกซ์

1.3.2 สามารถประยุกต์ใช้ระบบต้นแบบการพิสูจน์ตนให้มีรูปแบบที่แตกต่างออกไปได้

1.4 ขอบเขตของโครงงาน

1.4.1 ผู้ใช้สามารถล็อกอินเข้าใช้งานในระบบปฏิบัติการลินุกซ์ได้ โดยใช้ใบหน้าของผู้ใช้โดยผ่านโปรแกรมล็อกอินที่พัฒนาขึ้น

1.4.2 ระบบต้นแบบที่พัฒนาขึ้น ภายใต้ระบบปฏิบัติการลินุกซ์ (GNU/Linux Debian 3.1) เคอร์เนล 2.6.x

บทที่ 2
หลักการ และทฤษฎีที่เกี่ยวข้อง

2.1 ระบบพิสูจน์ตน

ในปัจจุบันนี้ กระบวนการพิสูจน์ตนถือได้ว่ามีความสำคัญต่อความปลอดภัยของระบบมาก กล่าวคือ หากระบบใด ๆ ที่ไม่มีการตรวจสอบการเข้าใช้งานของผู้ใช้งานระบบแล้ว ย่อมมีความเสี่ยงต่อความเสียหายที่อาจเกิดขึ้นกับระบบ เนื่องจากอาจมีการแอบอ้างเป็นบุคคลอื่น เพื่อให้ได้มาซึ่งสิทธิ์ในการเข้าถึงระบบ ไม่ว่าจะด้วยจุดประสงค์ใด ด้วยเหตุดังกล่าวการนำวิธีการพิสูจน์ตน ที่เหมาะสม ปลอดภัยมาประยุกต์ใช้จึงเป็นสิ่งที่ควรให้ความสำคัญ

2.1.1 ระบบรักษาความปลอดภัยบนคอมพิวเตอร์

ระบบรักษาความปลอดภัยบนคอมพิวเตอร์ คือ สิ่งที่คอยปกป้องคุ้มครองคอมพิวเตอร์ และสิ่งที่เกี่ยวข้องให้พ้นอันตราย และการสูญหาย ทุกสิ่งที่เกี่ยวข้องกับคอมพิวเตอร์ จะได้รับการคุ้มครองจากระบบรักษาความปลอดภัย ตามทฤษฎีระบบรักษาความปลอดภัยมีสิ่งที่ต้องคำนึงถึงดังต่อไปนี้

1. ความมั่นคงและถูกต้อง (Integrity & Accuracy) ข้อมูลที่อยู่บนคอมพิวเตอร์ต้องปลอดภัย ไม่สูญหาย ไม่เสียหาย ไม่ถูกเปลี่ยนแปลง โดยอุบัติเหตุหรือเจตนาจากผู้ที่ไม่ได้รับอนุญาต ในการส่งผ่านข้อมูลต้องมีการรับรองอย่างถูกต้อง มีบันทึกเกี่ยวกับการรับส่ง

2. ความมั่นใจ (Confidentiality) คอมพิวเตอร์ต้องเก็บรักษาความลับได้ จำแนกได้ว่าใครเป็นผู้มีสิทธ์ และใครคือผู้ไม่มีสิทธิ์จัดการข้อมูล

3. ความสามารถเข้าถึงข้อมูลได้ (Availability) ข้อมูลที่ควรเข้าถึงได้ ต้องเข้าถึงได้ง่าย สะดวกต่อการนำมาใช้ อยู่ในที่ที่สามารถนำมาใช้ได้ตลอดเวลา หาเกิดอุบัติเหตุ หรือข้อบกพร่องขึ้น ต้องสามารถกู้คืนได้

2.1.2 ส่วนประกอบของระบบความปลอดภัย

ส่วนที่ประกอบขึ้นมาเป็นระบบรักษาความปลอดภัยมีด้วยกัน 3 ข้อ

1. การออกแบบระบบ การออกแบบระบบที่ดีทำให้สามารถรักษาความปลอดภัยได้อย่างมีประสิทธิภาพ การใช้สถาปัตยกรรมเข้ามาจัดการระบบเป็นตัวอย่างหนึ่งของการออกแบบระบบที่ดี ทำให้สามารถจัดแบ่งหน่วยความจำ และแยกอภิสิทธิ์ออกจากสิทธิ์ทั่วไปได้

2. การควบคุมการจัดข้อมูล การควบคุมข้อมูลหมายถึง การกำหนดว่าให้ใครสามารถจัดการข้อมูลได้บ้าง และต้องกำหนดด้วยว่าจัดการข้อมูลไปเพื่อจุดประสงค์ใด

3. การควบคุมการจัดการในระบบ การควบคุมการจัดการในระบบทำให้สามารถกำหนดได้ว่าใครมีสิทธิ์ใช้ข้อมูลได้ถึงระดับไหน นอกจากนี้ยังทำให้แน่ใจด้วยผู้ที่ไม่ได้รับอนุญาต จะไม่มีสิทธิ์จัดการข้อมูล

2.1.3 การควบคุมการจัดการ

เป้าหมายหลักของการรักษาความปลอดภัย ต้องสามารถจำกัดได้ว่า ใครสามารถเข้าถึงข้อมูลได้มากขนาดไหน ซึ่งเรียกว่าการควบคุมการจัดการ (Access control) เหตุผลที่ต้องควบคุมจัดการ คือ

1. เพื่อสนับสนุนให้การเข้าถึงข้อมูลของผู้ที่ได้รับอนุญาตเป็นไปอย่างถูกต้องและง่ายดาย
2. เพื่อส่งเสริมให้เกิดความมั่นคงของข้อมูล
3. เพื่อปกป้องความเป็นส่วนตัวในข้อมูลส่วนบุคคล

นอกจากนี้แล้ว การควบคุมการจัดการยังมีขอบเขตไปถึงการจำกัดการใช้โปรแกรมต่าง ๆ เพื่อลบ เขียนทับ และทำสำเนาด้วย ในการควบคุมการจัดการเราต้องคำนึงถึงสิ่งต่อไปนี้

1. ใครที่สามารถใช้ได้บ้าง (Authentication)
2. ผู้ที่ได้รับอนุญาต มีสิทธิ์ใช้ส่วนใดและระดับใดได้บ้าง (Authorization)
3. ต้องบันทึกการกระทำต่าง ๆ ของผู้ที่ได้รับอนุญาต (Accounting)

เมื่อผู้ใช้ต้องการใช้บริการจากระบบ ผู้ใช้ต้องระบบว่าเป็นใคร และระบบจะตรวจสอบว่าผู้ใช้เป็นคนนั้นจริงหรือไม่ ทั้งสองขั้นตอนนี้เรียกว่าการแสดงตน (Identification) และการพิสูจน์ตน (Authentication)

2.1.4 การพิสูจน์ตน

การพิสูจน์ตน คือการที่ระบบตรวจสอบว่าผู้ใช้เป็นคนเดียวกับบุคคลที่อ้าง โดยวิธีการในการตรวจสอบมี 3 ลักษณะ

2.1.1.1 สิ่งที่คุณรู้ (What you know)

ใช้ในการพิสูจน์ตน โดยผู้ใช้ต้องรู้ในสิ่งระบบต้องการรู้ เพื่อใช้ตรวจสอบโดยปกติแล้ว คือ ชื่อผู้ใช้ (user name) และรหัสผ่าน (password) ซึ่งหมายความว่าหากทราบชื่อผู้ใช้ และรหัสผ่าน จะถือว่าเป็นผู้ใช้นั้นจริง วิธีนี้เป็นวิธีที่มีการใช้งานกันอย่างแพร่หลาย เนื่องจากสามารถใช้งานง่าย แต่การพิสูจน์ตนด้วยวิธีนี้ ยังเป็นวิธีการที่มีความปลอดภัยไม่สูงมาก เนื่องจากหากมีการขโมยรหัสผ่าน หรือมีผู้อื่นล่วงรู้รหัสผ่าน ซึ่งสามารถทำได้ไม่ยาก ทำให้สามารถเข้าใช้งานระบบได้

2.1.1.2 สิ่งที่คุณมี (What you have)

ใช้ในการพิสูจน์ตน โดยผู้ใช้ต้องมีในสิ่งที่ระบบต้องการให้แสดงเพื่อการตรวจสอบ เช่น บัตรสมาร์ตการ์ด บัตรแถบแม่เหล็ก RSA Token หรือกุญแจ ซึ่งแม้ว่าจะเป็นวิธีการที่ใช้เทคโนโลยีสูงกว่าแบบแรก แต่อาจจะมีปัญหาได้ หากสิ่งดังกล่าวถูกขโมย หรือแม้กระทั่งมีการทำการคัดลอก ซึ่งอาจจะยากกว่าแบบแรก ซึ่งจำเป็นต้องอาศัยเทคโนโลยีบางแขนงเข้ามาช่วย ซึ่งสามารถทำได้เช่นกัน

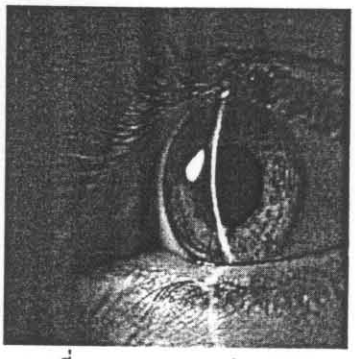
2.1.1.3 สิ่งที่คุณเป็น (What you are)

ใช้ในการพิสูจน์ตน โดยผู้ใช้ต้องแสดงในสิ่งซึ่งเป็นลักษณะทางกายภาพของผู้ใช้ นั่นคือการนำเทคโนโลยีทางทางชีวมาตร (Biometrics) มาใช้ในการตรวจสอบตัวตน โดยอาศัยอวัยวะที่เรามีอยู่ หรือสิ่งที่มีลักษณะเป็นเอกลักษณ์ เช่น ลายนิ้วมือ เสียง ใบหน้า ม่านตา เป็นต้น

วิธีการนี้ถือได้ว่าเป็นวิธีการที่มีประสิทธิภาพสูง การปลอมแปลง หรือคัดลอกทำได้ยากมาก แต่มีค่าใช้จ่ายสูงขึ้นเช่นกัน ตัวอย่างของวิธีการทางชีวมาตร

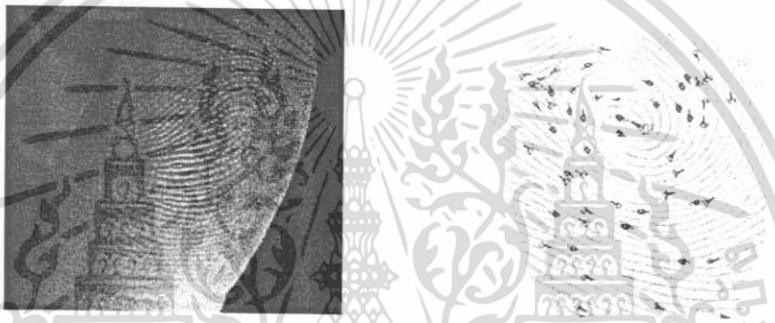
เรตินา เป็นวิธีที่ใช้การตรวจสอบเส้นเลือด บริเวณด้านหลังของดวงตา ซึ่งลักษณะที่ได้มานั้นจะมีความเป็นเอกภาพ สำหรับแต่ละบุคคลเช่นเดียวกับลายนิ้วมือ แต่อุปกรณ์มีราคาแพง

ม่านตา เป็นวิธีการซึ่งคล้ายกับการใช้ เรตินาจะทำการตรวจสอบลักษณะของม่านตา ซึ่งวิธีนี้จะมีส่วนติดต่อผู้ใช้ที่นับได้ว่าเป็นจุดเด่น คือ อุปกรณ์สแกนม่านตา สามารถทำการสแกนม่านตาได้แม้ระยะห่างระหว่างดวงตากับกล้องจะห่างกันพอสมควร



ภาพที่ 2-1 การสแกนม่านตา [11]

ลายนิ้วมือ ใช้วิธีตรวจสอบลายเส้นของนิ้วมือ ซึ่งแต่ละคนจะมีลักษณะที่แตกต่างกัน ซึ่งวิธีนี้มีการนำไปใช้งานกันเป็นเวลานานพอสมควร

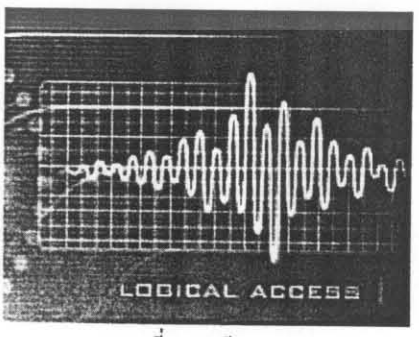


ภาพที่ 2-2 ลายนิ้วมือ และการตรวจสอบนิ้วมือ [12]

ใบหน้า เป็นวิธีการซึ่งใช้ใบหน้าของบุคคล วิธีนี้คอมพิวเตอร์สามารถวัดขนาดของใบหน้าและยังมีความสามารถในการจดจำใบหน้าบุคคลได้เป็นอย่างดี

ลายมือ เป็นวิธี ที่มีการใช้งานกันอย่างกว้างขวางมากกว่าวิธีการตรวจสอบลายนิ้วมือเพียงอย่างเดียว เนื่องจากวิธีนี้จะวัดขนาดของมือ ความยาวนิ้ว ความกว้างและส่วนอื่น ๆ ทั้งยังมีราคาถูกกว่าอุปกรณ์ของการตรวจสอบลายนิ้วมืออีกด้วย

เสียง เป็นวิธีที่จะใช้การพิจารณาลักษณะของสเปคตรัมความถี่ของเสียงของคน แต่วิธีนี้อาจมีปัญหาได้เช่นกัน กล่าวคือหากในการพิสูจน์ตนผู้ที่ทำการพิสูจน์ตนเป็นหวัดจะส่งผลต่อเสียงได้



ภาพที่ 2-3 เสียง [13]

จังหวะการพิมพ์ เป็นวิธีที่ใช้จังหวะการพิมพ์ของแต่ละบุคคลซึ่งมีลักษะที่แตกต่างกันอย่างชัดเจน

ภาพที่ 2-4 จังหวะการพิมพ์ [14]

ลายเซ็น เป็นวิธีการที่ใช้ตรวจสอบจากการเซ็นของบุคคลซึ่งยากที่จะลอกเลียนแบบให้มีระยะเวลา จังหวะการเคลื่อนไหวที่เหมือนกันได้ ในบางระบบได้นำวิธีการนี้ไปใช้โดยให้ผู้ใช้งานทำการเซ็นชื่อบนอุปกรณ์อิเล็คทรอนิคส์

2.2 Pluggable Authentication Modules (PAM)

Pluggable Authentication Modules (PAM) [6] ให้บริการแอพลิเคชันในระบบ ด้วยกระบวน การพิสูจน์ตน และบริการด้านความปลอดภัยอื่นที่เกี่ยวข้อง ซึ่งถูกออกแบบมาในลักษณะไลบรารีร่วม (shared library) เพื่อให้แอพลิเคชันอื่น ๆ สามารถเรียกใช้งานได้และรวมถึง ผู้ดูแลระบบเองสามารถเลือกวิธีการในการพิสูจน์ตนที่เหมาะกับความต้องการ และกับบริการที่มีอยู่ได้ และผู้พัฒนาแอพลิเคชันเองไม่จำเป็นต้องแก้ไข ในส่วนของซอร์สโค้ด (Sourcecode) โปรแกรมเมื่อต้องการวิธีการพิสูจน์ตนที่แตกต่างจากแบบเดิม

เฟรมเวิร์ก PAM เป็นเทคโนโลยีในการพิสูจน์ตนเพื่อเข้าใช้งานของผู้ใช้ในระบบ โดยใช้แนวคิดแบบปลั๊กอินโดยไม่ต้องทำการเปลี่ยนแปลงคำสั่งใด ๆ เช่น login, ftp, telnet เป็นต้น ภายในเฟรมเวิร์กนี้ยังประกอบด้วยกลไกสำหรับการจัดการบัญชีรายชื่อ (account), การจัดการเซสชัน (session) และการจัดการรหัสผ่าน (password) อีกด้วย

PAM ทำให้ผู้ดูแลระบบสามารถเลือกใช้บริการต่าง ๆ (service) ในการทำการพิสูจน์ตน ซึ่งมีประโยชน์ต่อผู้ดูแลระบบดังนี้

- นโยบายในการปรับตั้งค่าระบบที่ยืดหยุ่น
  - นโยบายการพิสูจน์ตนในแต่ละแอพพลิเคชัน
  - สามารถเลือกกลไกการพิสูจน์ตนพื้นฐานสำหรับแอพพลิเคชันที่ไม่มีการเฉพาะเจาะจงไว้
  - สามารถใช้รหัสผ่านหลายรหัสบนระบบที่มีความปลอดภัยสูง

อีกทั้งยังง่ายต่อการใช้งานสำหรับผู้ใช้งานระบบทั่วไปอีกด้วย ดังนี้

- ไม่ต้องทำการป้อนรหัสผ่านใหม่ ในกรณีที่เป็นผู้ใช้งานเดียวกัน
- ใช้เพียงรหัสผ่านเดียวเท่านั้น แม้รหัสผ่านนั้นเกี่ยวข้องกับวิธีการพิสูจน์ตนที่แตกต่างกันออกไป โดยใช้การเปรียบเทียบรหัสผ่าน

2.2.1 เฟรมเวิร์ก PAM

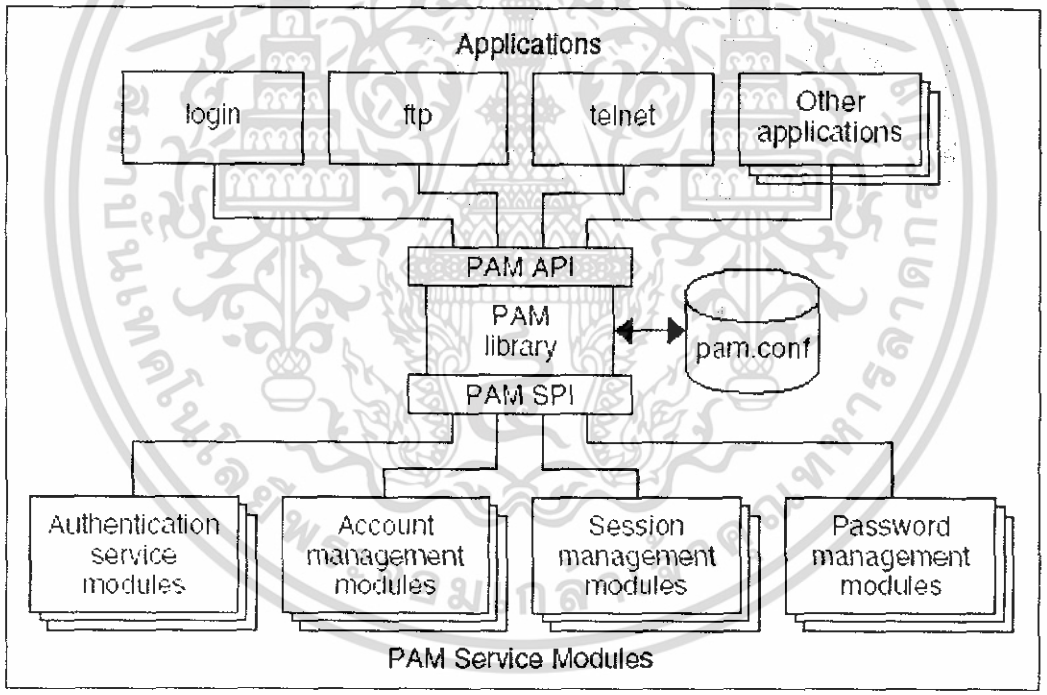
เฟรมเวิร์ก PAM ประกอบด้วย 4 ส่วน คือ

- PAM application
- PAM library
- PAM configuration file
- PAM service modules

เฟรมเวิร์กเปรียบเสมือนแบบแผน หรือแนวทางสำหรับกระบวนการที่เกี่ยวการพิสูจน์ตน แนวทางดังกล่าวส่งผลให้ ผู้พัฒนาแอพลิเคชันสามารถใช้บริการ PAM ได้โดยไม่จำเป็นต้องทราบถึงแนวทางหรือ วิธีการหรือกฎเกณฑ์

แนวทางในการพิสูจน์ตนนั้นสามารถดัดแปลงแก้ไขได้อย่างเป็นอิสระจาก แอพลิเคชัน และด้วยการทำงานของ PAM ส่งผลให้ผู้ดูแลระบบสามารถเพิ่มส่วนของการพิสูจน์ตนที่ต้องการกับระบบโดยไม่ต้องเปลี่ยนแปลงแอพลิเคชันใด ๆ การปรับเปลี่ยนสามารถทำได้โดยแก้ไขไฟล์ปรับแต่งค่าของ PAM

ภาพที่ 2-5 แสดงให้เห็นโครงสร้างเฟรมเวิร์ก PAM กล่าวคือ แอพลิเคชันติดต่อกับไลบรารี PAM ผ่าน PAM Application Programming Interface (API) ในขณะที่มอดูล PAM ติดต่อกับไลบรารี PAM ผ่าน PAM service provider interface (SPI) แสดงให้เห็นว่าไลบรารี PAM อนุญาตให้ แอพลิเคชัน และมอดูลสามารถติดต่อซึ่งกันและกันได้



ภาพที่ 2-5 แสดงความสัมพันธ์ ของ PAM Framework

แอพลิเคชัน จะมีการเรียกใช้ไลบรารี PAM เพื่อใช้งานมอดูลที่เกี่ยวข้องโดยเมื่อมีการเรียกใช้ไลบรารี PAM แล้วไลบรารี PAM จะดูค่าปรับตั้งค่า (configuration) ซึ่งมีการระบุไว้ที่ /etc/pam.conf หรือในไดเร็คทอรี /etc/pam.d/ ซึ่งไฟล์ดังกล่าวนี้จะทำหน้าที่บอกว่ามอดูลใดจะถูกใช้งานโดย แอพลิเคชันหนึ่ง ๆ และมอดูลจะตอบสนองต่อการเรียกใช้งานของแอพลิเคชันผ่านไลบรารี PAM

2.2.2 PAM Application

แอพลิเคชันซึ่งใช้ PAM ต้องมีการอ้างถึงไลบรารี libpam และก่อนที่แอพลิเคชันใด ๆ จะใช้บริการซึ่งมีให้โดยมอดูลแล้ว จำเป็นต้องเรียกฟังก์ชั่น pam_start() เพื่อสร้าง handle ในการส่งผ่าน การเรียกใช้ PAM ทุกกรณี และเมื่อแอพลิเคชันเสร็จสิ้นแล้วต้องเรียกฟังก์ชั่น pam_end() เพื่อทำการลบข้อมูลต่าง ๆ ที่มีการเรียกใช้โดย ไลบรารี PAM

2.2.3 PAM Library

ไลบรารี PAM เก็บอยู่ใน /usr/lib/libpam ถือเป็นศูนย์กลางในโครงสร้างของ PAM

libpam ส่ง API ออกไป ทำให้แอพลิเคชันสามารถเรียก API ดังกล่าวสำหรับการพิสูจน์ตน (authentication) การจัดการแอคเคาท์ (account management) การออกใบรับรอง (credential establishment) การจัดการเซสชัน (session management) และการเปลี่ยนรหัสผ่าน (password changes)

libpam นำไฟล์หลักในการปรับแต่งค่า (pam.conf) มาใช้ในการระบุเจาะจงว่า มอดูล PAM ใดที่จะถูกเรียกใช้สำหรับแต่ละบริการ ซึ่งผู้ดูแลระบบจะเป็นผู้จัดการไฟล์ปรับแต่งค่าของ PAM

libpam นำ pam_sm SPI มาใช้ซึ่งถูกใช้ใน service มอดูล

กระบวนการประยุกต์ใช้ไลบรารี PAM ในการพิสูจน์ตน

วิธีการซึ่งแอพลิเคชันเรียกใช้ไลบรารี PAM สำหรับการพิสูจน์ตน พิจารณาได้จากตัวอย่างในการพิสูจน์ตนของผู้ใช้โดยโปรแกรม login

1. แอพลิเคชัน เริ่มการเรียกใช้งาน PAM โดยเรียกฟังก์ชั่น pam_start() และระบุว่าเป็นบริการ login

2. แอพลิเคชันเรียกฟังก์ชั่น pam_authenticate() ซึ่งเป็นส่วนหนึ่งของ PAM API ที่ได้รับจาก PAM ไลบรารี

3. ไลบรารีจะค้นหาส่วนที่ได้มีการระบุค่า login ว่ามีอยู่ใน pam.conf หรือไม่

4. ในแต่ละมอดูล ใน pam.conf ซึ่งได้มีการปรับแต่งค่าสำหรับบริการ login และ PAM ไลบรารีจะเรียกใช้ฟังก์ชั่น pam_sm_authenticate() ซึ่ง pam_sm_authenticate() เป็นส่วนหนึ่งของ PAM SPI

การทำงานในลักษณะ Stack

เมื่อแอพลิเคชันเรียกใช้ฟังก์ชั่นใด ๆ ต่อไปนี้ libpam จะอ่านค่าจาก ไฟล์ปรับแต่งค่าเพื่อใช้ในการพิจารณาว่ามอดูลไหนจะถูกเรียกใช้งานสำหรับบริการ หรือแอพลิเคชัน

- pam_authenticate()
- pam_acct_mgmt()
- pam_setcred()
- pam_open_session()
- pam_close_session()
- pam_chauthok()

ภายในไฟล์ปรับแต่งค่าของ PAM นั้นจะมีค่าเพียง 1 มอดูล สำหรับแต่ละโอเปอเรชันของมอดูลนั้น เช่น authentication หรือ account management ผลที่ได้จากมอดูลดังกล่าวจะมีผลต่อโอเปอเรชันที่อยู่ถัดไป สำหรับตัวอย่าง เห็นได้ว่ากระบวนการ authentication สำหรับแอพลิเคชัน passwd จะมีเพียง 1 มอดูล คือ pam_passwd_auth.so.1

```
passwd          auth     required        pam_passwd_auth.so.1
```

ในทางกลับกันนั้น สามารถทำการกำหนดให้มีการเรียกใช้งานได้หลาย ๆ มอดูลซึ่งลักษณะนี้เรียกว่าเป็น stacked ซึ่ง PAM สนับสนุนการทำงานแบบ stack ด้วย

```
login           auth     requisite       pam_authtok_get.so.1
login           auth     required        pam_dhkeys.so.1
login           auth     required        pam_unix_cred.so.1
login           auth     required        pam_unix_auth.so.1
login           auth     required        pam_dial_auth.so.1
```

จากตัวอย่างแสดงให้เห็น stack ของการ authenticate ของบริการ login อย่างง่าย ซึ่งในการพิจารณาผลที่ได้จาก stack ดังกล่าว โค้ดผลลัพธ์ของแต่ละมอดูล จำเป็นต้องใช้ integration process ซึ่งใน integration process นั้น มอดูลจะถูก executed ตามลำดับที่ได้มีการระบุไว้ในไฟล์ปรับแต่งค่า โค้ดผลลัพธ์ที่ได้ทั้งสำเร็จและไม่สำเร็จจะถูกนำไปรวมไว้ในผลลัพธ์รวมซึ่งจะขึ้นกับค่าแฟลกคอนโทรลของมอดูล ค่าแฟลกคอนโทรลสามารถทำให้เกิดการสิ้นสุดของ stack ได้ หลังจากการประมวลผล stack เสร็จสิ้น ผลลัพธ์ที่ได้แต่ละค่าจะถูกนำมารวมกันเป็นค่าเดียวและค่าผลลัพธ์ ที่ได้นี้จะถูกส่งให้แอพลิเคชัน

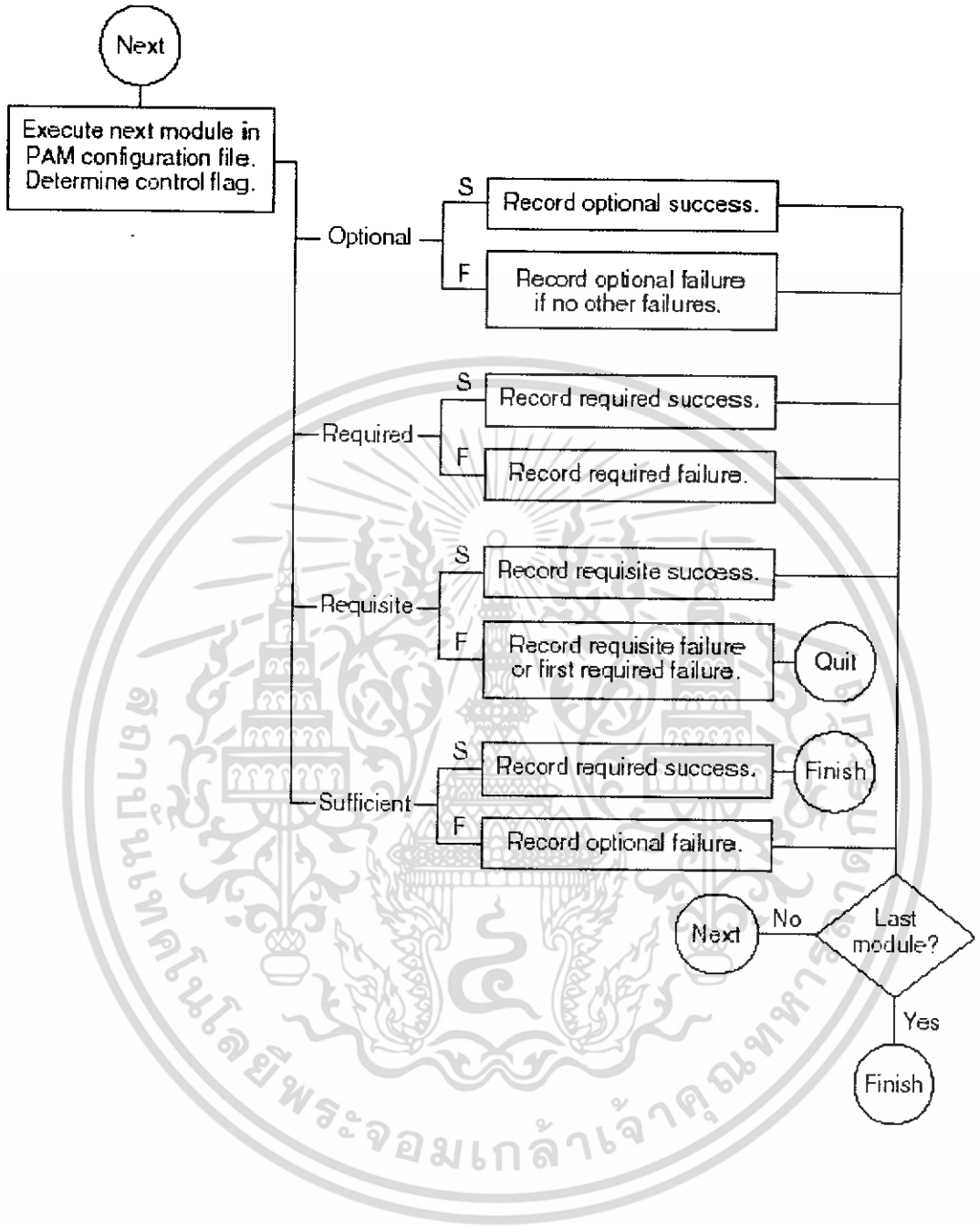
ค่าแฟลกคอนโทรล (control flag) ระบุกฎซึ่ง PAM มอดูล ใช้ในการพิจารณาตัดสินใจการเข้าถึง บริการซึ่งค่าแฟลกคอนโทรล มีดังนี้

1. *required* ทุกไลบรารีที่อยู่ในกลุ่ม module-type เดียวกันที่มีค่าแฟลกคอนโทรลเป็น required จะต้องคืนค่าเป็น success ทั้งหมด และการคืนค่ากลับเป็น success นั้นจะเกิดขึ้นเมื่อไม่มีมอดูลที่เป็น binding หรือมอดูลที่เป็น required return ค่า unsuccess

2. *requisite* หากไลบรารีใดที่มีค่าแฟลกคอนโทรลเป็น requisite เมื่อมีการ execute แล้วคืนค่า ออกมาเป็น unsuccess การ Authenticate จะถูกยกเลิก และคืนการทำงานไปยังแอพพลิเคชันที่เรียกใช้ PAM แต่จะมีการคืนค่ากลับเป็น success เมื่อทุก ๆ มอดูลที่เป็น requisite คืนค่ากลับเป็น success ทั้งหมด

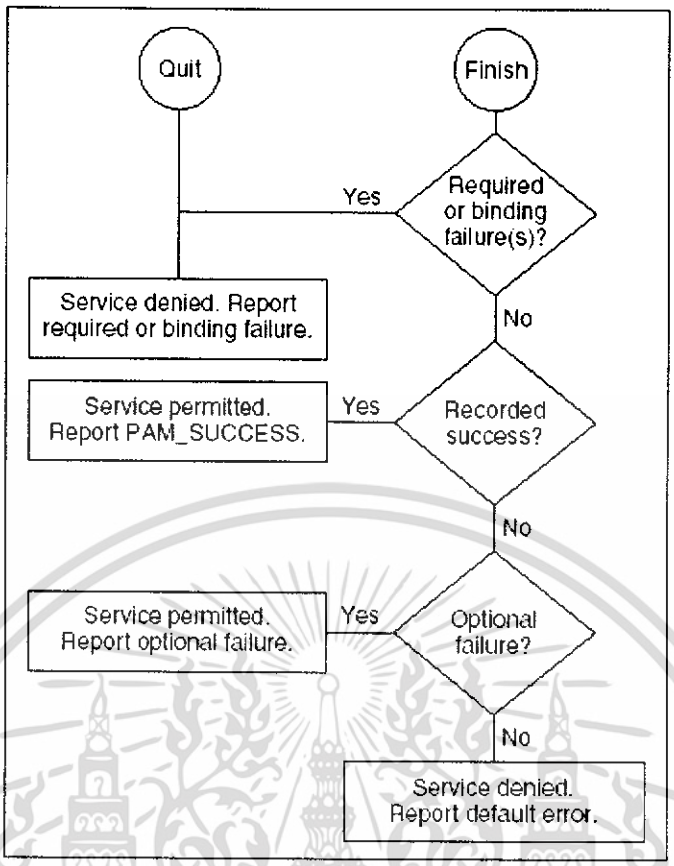
3. *sufficient* ถ้าไลบรารีใดมีค่าแฟลกคอนโทรลเป็น sufficient เมื่อมีการ execute แล้วคืนค่า มาเป็น success ระบบจะไม่สนใจการทำงานของไลบรารีอื่น ๆ ที่อยู่ในกลุ่ม module-type เดียวกัน และจะส่งค่า success คืนกลับไปให้แอพลิเคชันหากไม่มีมอดูลที่เป็น required ก่อนหน้ามีการคืนค่าเป็น unsuccess

4. *optional* ไม่ว่าค่าที่คืนกลับมาจะเป็น success หรือ unsuccess จะไม่มีผลต่อการ Authenticate ของระบบโดยรวม แต่จะมีผลเมื่อไฟล์ ปรับแต่งค่ามีการกำหนดค่าแฟลกคอนโทรลของเซอร์วิสชนิดนั้นเป็น optional เพียงอย่างเดียว



ภาพที่ 2-6 ผลที่เกิดจากค่าแฟลกคอนโทรล

แสดงการบันทึกค่า success หรือ failure ของแต่ละชนิดของแฟลกคอนโทรล



ภาพที่ 2-7 การรวมค่าผลลัพธ์

แสดงการรวมค่าผลลัพธ์ที่เก็บไว้ ว่ามีแนวทางในการพิจารณาผลลัพธ์สุดท้ายอย่างไร

2.2.4 PAM Modules

แต่ละมอดูลจะมีการกำหนดกระบวนการ และวิธีการเฉพาะ ในบางมอดูลอาจมีการกำหนดแค่ ชนิดของมอดูลเพียงชนิดเดียว ในขณะที่บางมอดูลอาจจะมีการกำหนดให้สามารถทำงานได้หลาย module-type ทั้งนี้ทั้งนั้นในแต่ละมอดูล จะต้องมีการกำหนดให้สามารถใช้งานได้ 1 module-type ตัวอย่างเช่น /usr/lib/security/pam_unix.so สนับสนุนทั้ง 4 module-type คือ การพิสูจน์ตน (Authentication) , การจัดการแอคเคาน์ (Account Management), การจัดการเซสชั่น (Session Management) และการจัดการรหัสผ่าน (Password Management)

จะเห็นได้ว่า PAM จะมีการแบ่งงาน เป็น 4 ส่วน ซึ่งจะเป็นอิสระต่อกัน และแต่ละส่วนจะทำงานแตกต่างกันออกไป

2.3.4.1 Authentication management (auth) ส่วนนี้จะดูแลในเรื่องการพิสูจน์สิทธิ์โดยตรงซึ่งตามปกติแล้ว การพิสูจน์สิทธิ์นั้น จะกระทำโดยมีการตรวจสอบ ชื่อบัญชีผู้ใช้และรหัสผ่าน ถ้าการตรวจสอบผ่าน จึงจะสามารถเข้าใช้บริการได้แต่ในบางครั้งการพิสูจน์สิทธิ์ อาจทำ

โดยรูปแบบอื่น เช่น ผ่านทาง สมาร์ตการ์ด ลายนิ้วมือ หรือใบหน้า ดังนั้นจึงเป็นสิ่งจำเป็นที่ผู้ดูแลระบบ จะต้องเลือกใช้มอดูลให้ตรงกับกระบวนการในการพิสูจน์ตน

2.3.4.2 Account management (account) ส่วนนี้จะดูแลในลักษณะการจัดการการใช้บริการ หรืออนุญาตให้ใช้บริการนั้นได้ เช่น บัญชีผู้ใช้ นี้มีสิทธิ์ที่จะเข้าใช้บริการหรือไม่

2.3.4.3 Password management (password) ส่วนนี้ถูกใช้ในการกำหนดรหัสผ่านของบัญชีผู้ใช้ หรือทำการกำหนดรหัสผ่านใหม่เมื่อมีการเปลี่ยนแปลงเกิดขึ้น

2.3.4.4 Session management (session) ส่วนนี้ จะระบุว่าจะมีการทำอะไรบ้างในช่วงที่ ผู้ใช้บริการเริ่มใช้ และช่วงหลังจากที่ใช้บริการเสร็จแล้ว เช่น ช่วงเริ่มใช้บริการอาจมีการบันทึกข้อมูลลง log และหลังจากที่ ใช้บริการเสร็จแล้วอาจมีการส่งข้อความไปแจ้งให้ระบบได้รับรู้

2.2.5 PAM Configuration file

ไฟล์สำหรับการปรับแต่งค่า (configuration file) ของ PAM จะมีสองรูปแบบ คือ แบบ configuration file เดียว (/etc/pam.conf) วิธีการพิสูจน์สิทธิ์ ของทุก ๆ บริการจะถูกเก็บไว้ภายในไฟล์นี้เท่านั้น และแบบที่ 2 คือ แบบแยกเป็นแต่ละเซอร์วิสที่ให้บริการ ซึ่งจะไฟล์คอนฟิกกูเรชันสำหรับแต่ละเซอร์วิสจะถูกเก็บไว้ที่ตำแหน่ง /etc/pam.d เรียกรูปแบบการปรับตั้งค่าของในลักษณะนี้ว่า Directory based configuration

2.3.5.1 กรณี *configuration file เป็นแบบไฟล์เดียว* (/etc/pam.conf) ไฟล์นี้จะเก็บค่าที่ระบุว่า แอพพลิเคชันใดในระบบจะเรียกใช้วิธีการ หรือกระบวนการในการพิสูจน์ตนใด โดยการระบุค่าในไฟล์มีลักษณะ ดังนี้

service_name module_type control_flag module_path module_options

service_Name ระบุชื่อของบริการ เช่น ftpd , login
module_type ระบุ module-type เช่น auth , account , password ,session
control_flag ระบุค่าแฟลกคอนโทรล
module_path ระบุตำแหน่งของ Library Object
module_options ระบุค่า options ที่สามารถส่งไปยังมอดูล

ทั้ง 5 ส่วนที่กล่าวมานั้น มีเพียงส่วน module_options เท่านั้นที่สามารถละเว้นได้ นอกจากนั้นแล้วทั้ง 4 ส่วนแรก เป็นส่วนที่จำเป็นซึ่งต้องระบุลงไป

ตัวอย่างไฟล์ pam.conf ซึ่งมีการระบุการใช้งานทุก ๆ เซอร์วิสในไฟล์เพียงไฟล์เดียว

```
# PAM configuration
# Authentication management
#
login       auth       Required       /usr/lib/security/pam_unix.so.1
login       auth       Required       /usr/lib/security/pam_dial_auth.so.1
rlogin      auth       Sufficient     /usr/lib/security/pam_rhost_auth.so.1
rlogin      auth       Required       /usr/lib/security/pam_unix.so.1
dtlogin     auth       Required       /usr/lib/security/pam_unix.so.1
telnet      auth       Required       /usr/lib/security/pam_unix.so.1
su          auth       Required       /usr/lib/security/pam_unix.so.1
ftp         auth       Required       /usr/lib/security/pam_unix.so.1
uucp        auth       Required       /usr/lib/security/pam_unix.so.1
rsh         auth       Required       /usr/lib/security/pam_rhost_auth.so.1
OTHER       auth       Required       /usr/lib/security/pam_unix.so.1
#
# Account management
login       account    Required       /usr/lib/security/pam_unix.so.11
rlogin      account    Required       /usr/lib/security/pam_unix.so.1
dtlogin     account    Required       /usr/lib/security/pam_unix.so.1
telnet      account    Required       /usr/lib/security/pam_unix.so.1
ftp         account    Required       /usr/lib/security/pam_unix.so.1
#
# Session management
#
login       session    Required       /usr/lib/security/pam_unix.so.1
rlogin      session    Required       /usr/lib/security/pam_unix.so.1
dtlogin     session    Required       /usr/lib/security/pam_unix.so.1
telnet      session    Required       /usr/lib/security/pam_unix.so.1
uucp        session    required       /usr/lib/security/pam_unix.so.1
```

```
OTHER     session     Required     /usr/lib/security/pam_unix.so.1
#
# Password management
#
passwd    password    Required     /usr/lib/security/pam_unix.so.1
OTHER     password    Required     /usr/lib/security/pam_unix.so.1
```

2.3.5.2 *กรณี configuration ไฟล์ แยก* สำหรับแต่ละเซอร์วิส ซึ่งจะคล้าย ๆ กับการใช้คอนฟิกกูเรชันไฟล์เพียงไฟล์เดียว หากแต่จะต่างกันที่แบบหลังนี้จะใช้ ชื่อไฟล์เป็นชื่อของบริการนั้น ๆ เลย จึงทำให้ภายในไฟล์ไม่การคำหนดชื่อเซอร์วิส จึงทำให้การระบุค่าภายในไฟล์เป็นดังนี้

module_type control_flag module_path arguments

ตัวอย่างไฟล์ /etc/pam.d/su

```
Auth        sufficient    /lib/security/pam_rootok.so
Auth        required      /lib/security/pam_stack.so      service=system-auth
Account     required      /lib/security/pam_stack.so      service=system-auth
password    required      /lib/security/pam_stack.so      service=system-auth
Session     required      /lib/security/pam_stack.so      service=system-auth
Session     required      /lib/security/pam_xauth.so
```

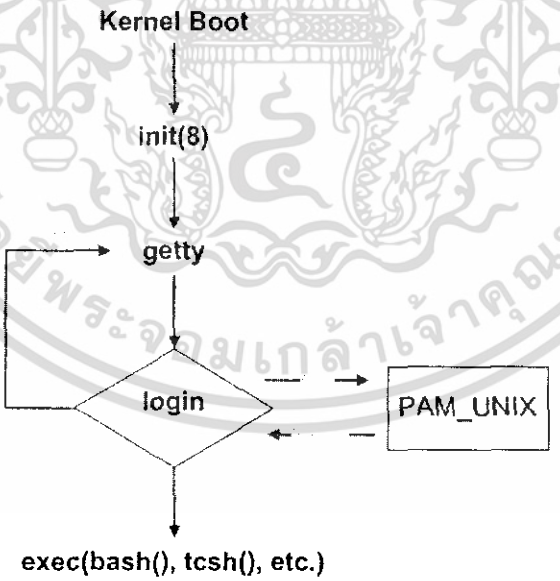
บทที่ 3
ขั้นตอนการออกแบบ การทำงาน และการวิเคราะห์

ในปริญญานิพนธ์ฉบับนี้ การออกแบบระบบต้นแบบการพิสูจน์ตนด้วยชีวมาตรโดยใช้ใบหน้าในการล็อกอินเข้าใช้งานในระบบปฏิบัติการลินุกซ์นั้น จะแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ ส่วนของระบบพิสูจน์ตน และส่วนรู้จำใบหน้า

3.1 แนวทางการออกแบบ

3.1.1 การออกแบบโปรแกรมส่วนการพิสูจน์ตน

ในการพัฒนาระบบต้นแบบการพิสูจน์ตนด้วยชีวมาตร จำเป็นต้องศึกษาทฤษฎี PAM และอาศัยหลักการการรู้จำใบหน้า โดยอาศัยหลักการและแนวคิดในการเปลี่ยนแปลงกลไกการล็อกอินแบบเดิม ที่ใช้การป้อนข้อมูล ซึ่งได้แก่ ชื่อบัญชีผู้ใช้งานในระบบ และรหัสผ่าน มาเป็นการใช้การรับภาพจากกล้อง Web camera เพื่ออาศัยกระบวนการรู้จำใบหน้าแทน โดยวิธีการล็อกอินเพื่อใช้งานในระบบแบบเดิมนั้น มีกระบวนการดังนี้

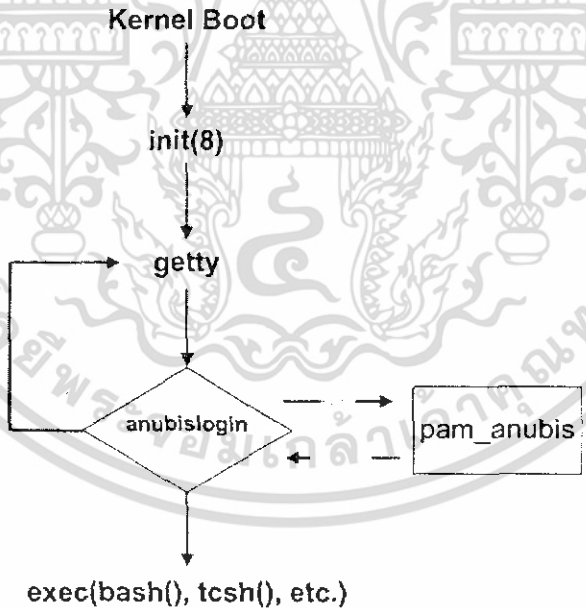


ภาพที่ 3-1 แสดงโครงสร้างการล็อกอินแบบดั้งเดิม

โดยมีกระบวนการการทำงานดังนี้

- หลังจากเปิดเครื่องแล้ว ระบบเรียกใช้งานเคอร์เนลบูต (kernel boots)
- จากนั้นทำการเรียก init ซึ่งภายในประกอบด้วยไฟล์ /etc/inittab ซึ่งใช้เก็บข้อมูลการเรียกใช้งาน getty ไว้
- ซึ่งเมื่อเรียกใช้งาน getty เพื่อทำ opentty() คือ ทำการเปิด /dev/tty และทำ do_prompt() คือ การแสดงรายละเอียดจาก /etc/issue
- การล็อกอิน จะรอรับข้อมูลยูสเซอร์เนม และรหัสผ่านจากผู้ใช้งานระบบ
- เมื่อได้รับข้อมูลยูสเซอร์เนมและรหัสผ่านจากผู้ใช้งานในระบบแล้ว จะทำการเรียกใช้งานมอดูล pam_unix เพื่อตรวจสอบการพิสูจน์ตนของผู้เข้าใช้งานในระบบ
- ถ้าการตรวจสอบถูกต้อง ผู้ใช้งานสามารถเข้าใช้งานในระบบได้ แต่ถ้าเกิดความผิดพลาดของข้อมูลที่ป้อนเข้ามา ผู้ใช้งานจะไม่สามารถเข้าใช้งานในระบบได้

แต่สำหรับระบบต้นแบบการพิสูจน์ตนด้วยชีวมาตร โดยอาศัยการล็อกอินด้วยใบหน้านั้น ทำการเปลี่ยนแปลงขั้นตอนของโครงสร้างการล็อกอิน ดังนี้



ภาพที่ 3-2 แสดงโครงสร้างการล็อกอินที่พัฒนาขึ้นใหม่

โดยมีกระบวนการการทำงานดังนี้

- หลังจากเปิดเครื่องแล้ว ระบบเรียกใช้งานเคอร์เนลบูต (kernel boots)
- จากนั้นทำการเรียก init ซึ่งภายในประกอบด้วยไฟล์ /etc/inittab ซึ่งใช้เก็บข้อมูลการเรียกใช้งาน getty ไว้
- ซึ่งเมื่อเรียกใช้งาน getty เพื่อทำ opentty() คือ ทำการเปิด /dev/tty
- การล็อกอิน จะใช้งาน anubislogin ที่พัฒนาขึ้นใหม่ ซึ่งจะรอรับข้อมูลยูสเซอร์เนมจากผู้ใช้งานระบบ
- เมื่อได้รับข้อมูลยูสเซอร์เนมจากผู้ใช้งานในระบบแล้ว จะทำการเรียกใช้งานมอดูล pam_anubis ที่พัฒนาขึ้น เพื่อตรวจสอบการพิสูจน์ตนของผู้เข้าใช้งานในระบบ โดยจะรอรับข้อมูลใบหน้าของผู้เข้าใช้งานในระบบผ่านทาง Web camera ซึ่งจะการเปรียบเทียบใบหน้าที่ได้จาก Web camera กับข้อมูลที่มีอยู่ในระบบ
- ถ้าการตรวจสอบถูกต้อง ผู้ใช้งานสามารถเข้าใช้งานในระบบได้ แต่ถ้าเกิดความผิดพลาดของข้อมูลที่ป้อนเข้ามา ผู้ใช้งานจะไม่สามารถเข้าใช้งานในระบบได้

3.1.2 การวางแผนสำหรับ PAM

เมื่อพิจารณาที่จะใช้งาน PAM ในระบบแล้ว ควรมุ่งเน้นการด้านต่าง ๆ ดังนี้

- ตัดสินใจว่าความต้องการของระบบ คือ อะไร โดยเฉพาะอย่างยิ่งมอดูลไหนที่ควรเลือกใช้งาน
- เจาะจงเซอร์วิสที่ต้องการใช้งานเป็นพิเศษ
- พิจารณาคำสั่งที่จะทำให้มอดูลนั้นทำงานได้
- เลือกแฟลกควบคุมสำหรับมอดูลนั้น ๆ
- เลือกตัวเลือกที่จำเป็นสำหรับมอดูลนั้น ๆ

สำหรับการเปลี่ยนแปลงคอนฟิกกูเรชันไฟล์ควรปฏิบัติดังนี้

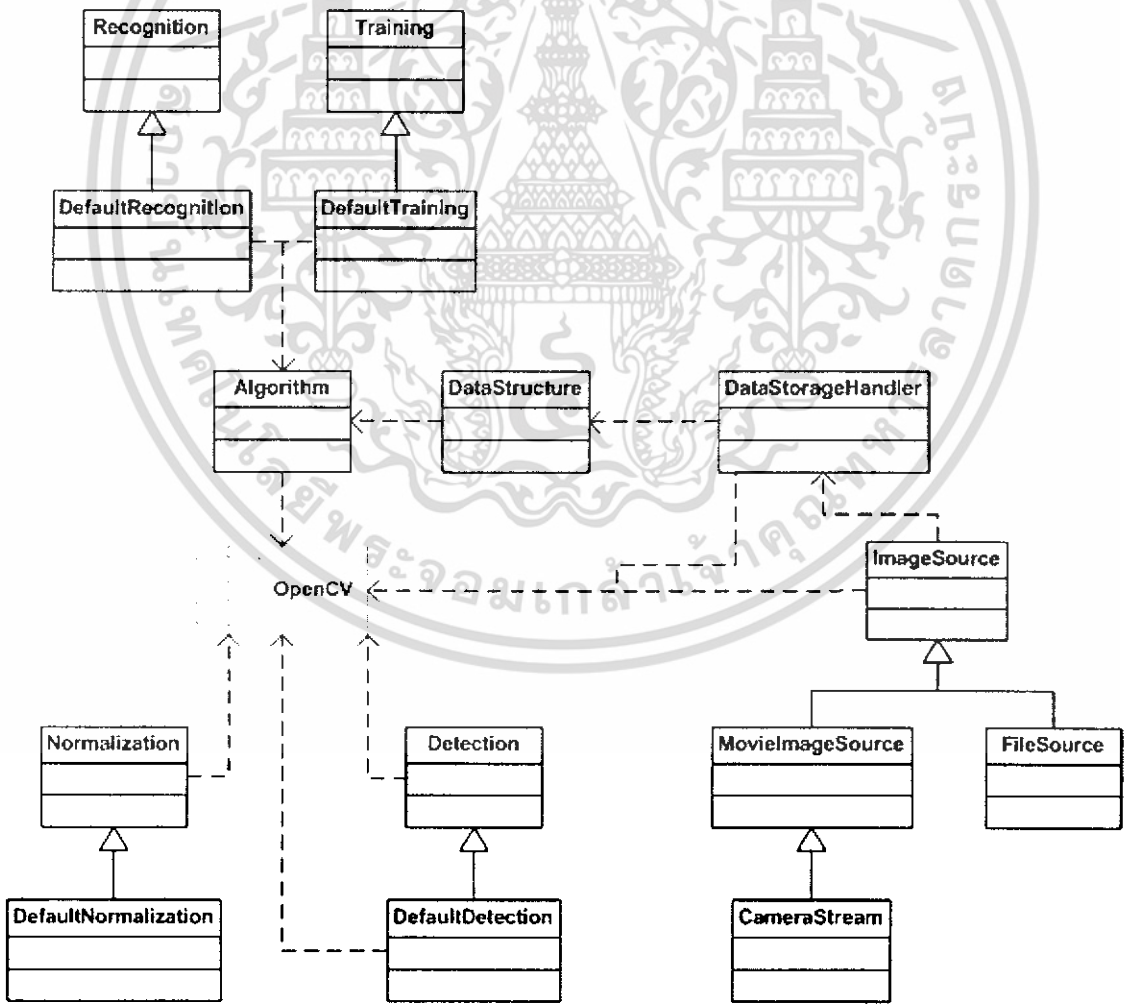
- ใช้เอนทรี OTHER สำหรับแต่ละรูปแบบมอดูลเพื่อที่แต่ละแอพพลิเคชันได้ต้องถูกรวมเข้าไปด้วย
- ทำให้แน่ใจว่ามีการพิจารณาความปลอดภัยของแฟลกควบคุม *sufficient* และ *optional*
- ทบทวน man pages ที่เกี่ยวข้องกับโมดูลเพื่อทำความเข้าใจว่ามีหน้าที่อย่างไร และมีตัวเลือกอะไรบ้าง
- ทบทวน man pages เพื่อศึกษาการตอบสนองระหว่างมอดูลสแต็กต่าง ๆ

ทั้งนี้ ระบบที่ได้พัฒนาในส่วนของ pam_anubis ซึ่งเป็นมอดูล PAM ที่ใช้เพื่อทำการพิสูจน์ตนที่ทำหน้าที่ในการรับข้อมูลใบหน้าผู้ใช้งานระบบ จากกล้อง Web camera และนำมาเปรียบเทียบกับข้อมูลใบหน้าที่มีในระบบ ด้วยวิธีการ Verification คือ การใช้รหัสผู้ใช้ (UID) ในการดึงข้อมูลใบหน้าที่มีอยู่มาเปรียบเทียบกับข้อมูลที่ได้รับมาใหม่ จากกล้อง Web camera และส่งกลับค่าผลลัพธ์ที่ได้กลับไปยัง anubislogin()

3.1.2 การออกแบบโปรแกรม ส่วนรู้จำใบหน้า

ในการจัดทำปริญญานิพนธ์ฉบับนี้ ได้นำเฟรมเวิร์ค OpenCV (Open Source Computer Vision Library) [14] ที่พัฒนาโดยบริษัทอินเทล คอเปอร์เรชันจำกัด (Intel Corporation) มาช่วยในการพัฒนาโครงงาน โดยเฟรมเวิร์ค OpenCV นั้น พัฒนาด้วยโปรแกรมภาษา C/C++ ซึ่งมี Data Structure และมีฟังก์ชันที่คลอบคลุมกระบวนการต่าง ๆ ในด้าน Image processing เช่น กระบวนการ Detection, Normalization, Training, และ Recognition เป็นต้นดังนั้น จึงนำเอาเฟรมเวิร์ค OpenCV มาช่วยในกระบวนการรู้จำใบหน้า

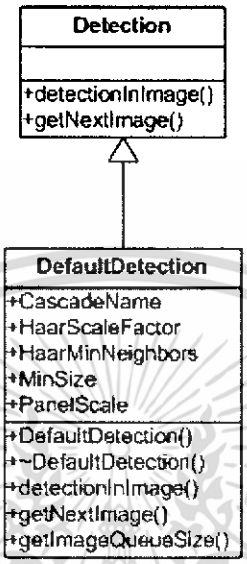
3.1.2.1 Class diagram



ภาพที่ 3-3 คลาสไดอะแกรม แสดงภาพรวมของเฟรมเวิร์ค OpenCV

3.1.2.1.1 Face detection class

ในกระบวนการหาใบหน้าบุคคลนั้น จะทำการ call ไปที่เมธอด detectInImage() เพื่อค้นหาใบหน้าจากรูปที่รับเข้ามาทาง Web camera



ภาพที่ 3-4 คลาส DefaultDetection จะ Implement method ต่าง ๆ ที่กำหนดโดยคลาส Detection

ในส่วนของคลาส Detection ประกอบด้วย

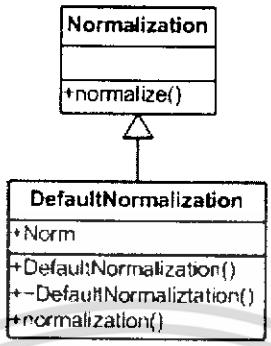
- *detectionInImage()*: หาใบหน้าจากภาพที่ได้จาก Web camera
- *getNextImage()*: รับภาพจาก Web camera เพื่อให้สามารถหาใบหน้าอย่างต่อเนื่อง

ในส่วนของคลาส DefaultDetection จะ Implement มาจาก face detection algorithm ของเฟรมเวิร์ค OpenCV โดยประกอบไปด้วยเมธอด

- *DefaultDetection()*: คอนสทรัคเตอร์
- *~DefaultDetection()*: ดิสทรัคเตอร์
- *detectionInImage()*: หาใบหน้าจากภาพที่ได้จาก Web camera
- *getNextImage()*: รับภาพจาก Web camera เพื่อให้สามารถหาใบหน้าอย่างต่อเนื่อง
- *getImageQueueSize()*: รับค่าขนาดที่ได้จากการหาใบหน้าปัจจุบัน

3.1.2.1.2 Normalization class

กระบวนการทำ Normalization รูปภาพ เพื่อให้ได้รูปแบบที่ต้องการ



ภาพที่ 3-5 เมธอดที่ถูกใช้โดยคลาส Normalization

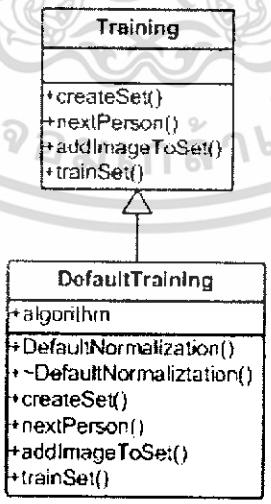
ในส่วนของคลาส Detection ประกอบด้วย

- *normalize()*: กระบวนการ normalization รูปภาพ

ในส่วนของคลาส DefaultDetection ประกอบด้วย

- *DefaultNormalization()*: คอนสทรัคเตอร์
- *~DefalutNormalization()*: ดิสทรัคเตอร์
- *normalize()*: กระบวนการ normalization รูปภาพ

3.1.2.1.3 Training class



ภาพที่ 3-6 เมธอดที่ถูกใช้โดยคลาส Training

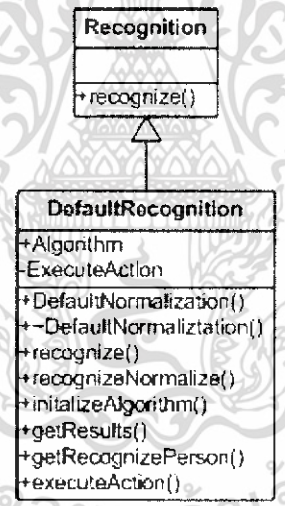
ในส่วนของคลาส Training ประกอบด้วย

- *createSet()*: ทำการสร้างกลุ่มข้อมูลขึ้นมาใหม่
- *nextPerson()*: เพิ่มผู้ใช้ในข้อมูลส่วน Training (Training set)
- *trainSet()*: ทำกระบวนการ Training

ในส่วนของคลาส DefaultTraining ประกอบด้วย

- *DefaultNormalization()*: คอนสทรัคเตอร์
- *~DefalutNormalization()*: ดิสทรัคเตอร์
- *createSet()*: ทำการสร้างกลุ่มข้อมูลขึ้นมาใหม่
- *nextPerson()*: เพิ่มผู้ใช้ในข้อมูลส่วน Training (Training set)
- *trainSet()*: ทำกระบวนการ Training

3.1.2.1.4 Recognition class



ภาพที่ 3-7 เมธอดที่ถูกใช้โดยคลาส Recognition

ในส่วนของคลาส Recognition ประกอบด้วย

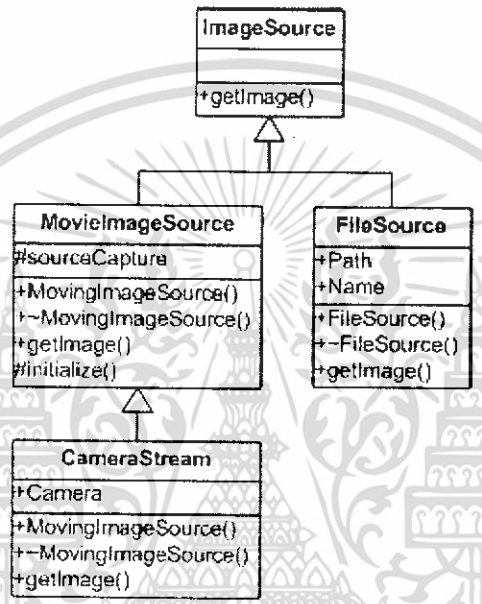
- *recognize()*: กระบวนการรู้จำใบหน้า

ในส่วนของคลาส DefaultRecognition ประกอบด้วย

- *DefaultNormalization()*: คอนสทรัคเตอร์
- *~DefalutNormalization()*: ดิสทรัคเตอร์
- *recognize()*: กระบวนการรู้จำใบหน้า

- *recognizeNormalization()*: กระบวนการรู้จำใบหน้าและการทำ Normalize
- *initializeAlgorithm()*:
- *getResult()*: ผลลัพธ์ที่ได้จากกระบวนการรู้จำใบหน้า
- *getRecognizePerson()*: ผลลัพธ์ที่ได้จากกระบวนการรู้จำเพื่อระบุถึงตัวบุคคล

3.1.2.1.5 Image acquiring class



ภาพที่ 3-8 เมธอดที่ถูกใช้โดยคลาส Image acquiring

ในส่วนของคลาส ImageSource ประกอบด้วย

- *getImage()*: รับภาพใบหน้าจาก Web camera

ในส่วนของคลาส MovieImageSource ประกอบด้วย

- *MovingImageSource()*: คอนสทรัคเตอร์
- *~ MovingImageSource()*: ดิสทรัคเตอร์
- *getImage()*: รับภาพเข้ามาจาก web camera
- *initialize()*: เรียกใช้ CvCapture()

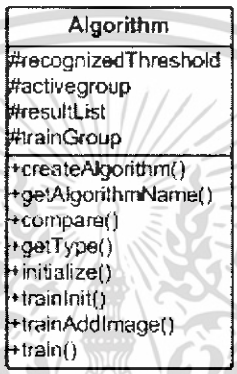
ในส่วนของคลาส FileSource ประกอบด้วย

- *FileSource()*: คอนสทรัคเตอร์
- *~FileSource()*: ดิสทรัคเตอร์
- *getImage()*: รับภาพเข้ามาจาก web camera

ในส่วนของคลาส CameraStream ประกอบด้วย

- *MovingImageSource()*: คอนสทรัคเตอร์
- *~ MovingImageSource()*: ดิสทรัคเตอร์
- *getImage()*: รับภาพเข้ามาจาก web camera
- *initialize()*: เรียกใช้ CvCapture()

3.1.2.1.6 Algorithm class

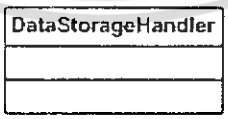


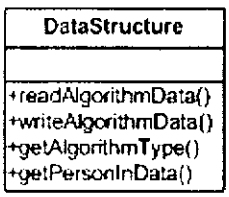
ภาพที่ 3-9 คลาส Algorithm

ในส่วนของคลาส Algorithm ประกอบด้วย

- *createAlgorithm()*:
- *getAlgorithmName()*:
- *compare()*: กระบวนการเปรียบเทียบ

3.1.2.1.7 File handling





ภาพที่ 3-10 คลาส DataStorageHandler และคลาส DataStructure

ในส่วนของคลาส DataStorageHandler อาศัยการ Implement จากเฟรมเวิร์ค OpenCV ซึ่งในการเข้าถึงแฟ้มข้อมูลนั้น แฟ้มข้อมูลจะถูกจัดเก็บในรูปของ XML ซึ่งจะมี parser คอยจัดการ (XML parser)

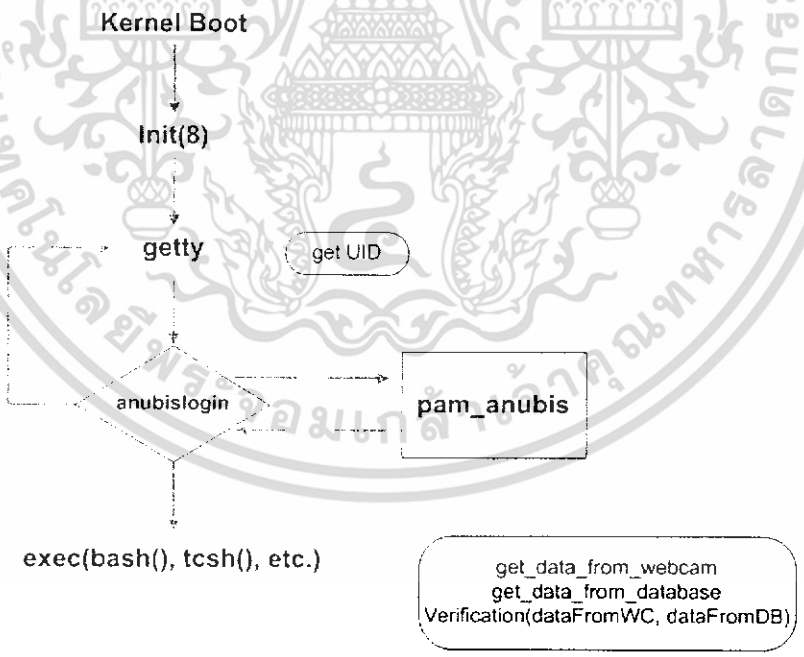
ในส่วนของคลาส DataStructure ประกอบด้วย

- *readAlgorithmData()*: อ่านค่าข้อมูลจากไฟล์ XML
- *writeAlgorithmData()*: บันทึกข้อมูลลงไฟล์ XML
- *getAlgorithmType()*: รับค่าในส่วนของ Algorithm ที่ใช้
- *getPersonInData()*: รับค่าผลลัพธ์ในส่วนของชื่อผู้ใช้ (Person ID)

นอกจากนี้ยังได้พัฒนาโปรแกรม anubisadd และโปรแกรม anubisdel เพื่อรองรับการใช้งานในการเพิ่มผู้ใช้งานในระบบ และการลบผู้ใช้งานออกจากระบบตามลำดับ

3.2 รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค

3.2.1 ส่วนของการล็อกอิน (login)



ภาพที่ 3-11 แสดงโครงสร้างการล็อกอินที่พัฒนาขึ้น

3.2.1.1 Input/Output Specification

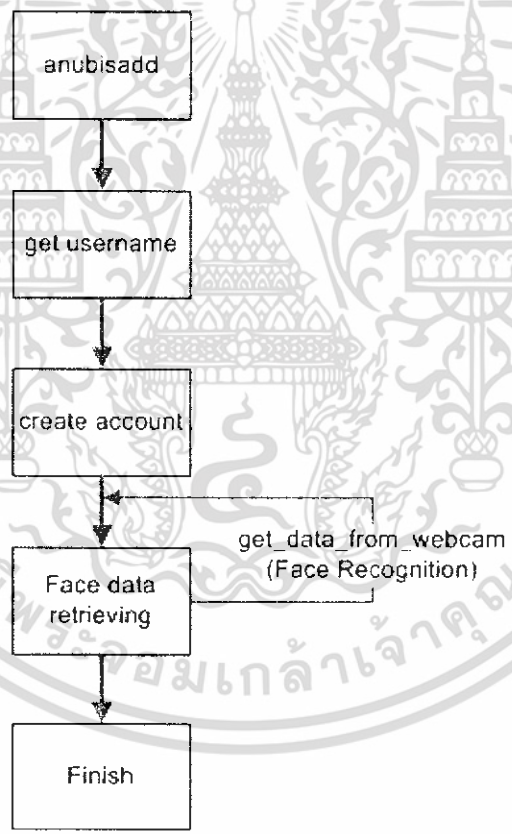
Input Specification ข้อมูลรับเข้า คือ ชื่อบัญชีผู้ใช้ที่จะเข้าใช้งานระบบ และภาพใบหน้าของผู้ใช้คนนั้นซึ่งได้จากกล้องเว็บแคม

Output Specification ข้อมูลออก คือ X-Window

3.2.1.2 Functional Specification

pam_anubis	มอดูลที่ใช้ในการพิสูจน์ตน
pam_sm_authenticate()	ฟังก์ชั่นหลักสำหรับการพิสูจน์ตนของมอดูล PAM
pam_get_user()	ดึงค่าข้อมูลผู้ใช้ สำหรับการประมวลผล
pam_set_item()	ตั้งค่าข้อมูลสำหรับการพิสูจน์ตน

3.2.2 ส่วนของการเพิ่มบัญชีผู้ใช้



ภาพที่ 3-12 แสดงโครงสร้างซอฟแวร์ส่วนการเพิ่มบัญชีผู้ใช้

3.2.2.1 Input/Output Specification

Input Specification ข้อมูลรับเข้า คือ ชื่อบัญชีผู้ใช้ที่จะเพิ่มเข้าสู่ระบบ และภาพใบหน้าของผู้ใช้คนนั้นซึ่งได้จากกล้องเว็บแคม

Output Specification ข้อมูลออก คือ ชื่อบัญชีผู้ใช้คนใหม่ถูกเพิ่มเข้ามาในระบบ และข้อมูล ใบหน้าของผู้ใช้ถูกเก็บในฐานข้อมูล

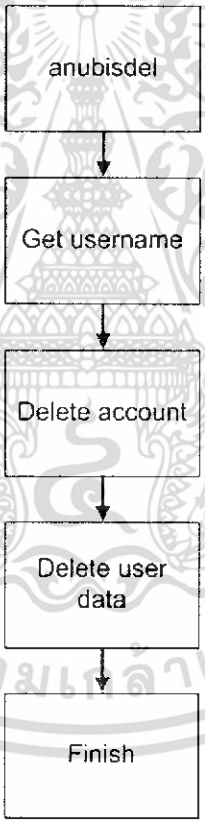
3.2.2.2 Functional Specification

anubisadd

execl() เพื่อรันคำสั่ง adduser เพื่อเพิ่มผู้ใช้ในระบบและสร้าง home directory

getpwnam() เพื่อดึงค่าข้อมูลในไฟล์ /etc/passwd ของผู้ใช้คนดังกล่าว

3.2.3 ส่วนของการลบบัญชีผู้ใช้



ภาพที่ 3-13 แสดงโครงสร้างซอฟแวร์ส่วนการลบบัญชีผู้ใช้

3.2.3.1 Input/Output Specification

Input Specification ข้อมูลรับเข้า คือ ชื่อบัญชีผู้ใช้ที่จะลบออกจากระบบ

Output Specification ข้อมูลออก คือ ชื่อบัญชีผู้ใช้คนนั้น รวมถึงข้อมูลของผู้นั้นถูกลบจากระบบ

3.2.3.2 Functional Specification

anubisdel

execl() เพื่อเรียกคำสั่ง deluser เพื่อลบผู้ใช้งานในระบบ

getpwnam() ดึงข้อมูลจากไฟล์ /etc/passwd เพื่อใช้ในการลบผู้ใช้ออกจากระบบ

3.3 รายละเอียดของการพัฒนา

อุปกรณ์ที่ใช้ในการพัฒนา

- ระบบปฏิบัติการ GNU/Linux (Debian 3.1) kernel version 2.6
- Logitech Quick Zoom web camera
- OpenCV (Open source Computer Vision)
- GNU C/C++ compiler
- VIM Editor
- libglade, libgtk+2.0

3.4 รายละเอียดของกล้องถ่ายภาพ

ในปริญญานิพนธ์ฉบับนี้ ใช้กล้องถ่ายภาพ ยี่ห้อ Logitech QuickCam Zoom Silver

สามารถถ่ายวิดีโอได้ ที่ความละเอียดสูงสุด 640 * 480 พิกเซล

อัตราการถ่ายวิดีโอต่อเนื่อง 30 เฟรม ต่อวินาที

สามารถถ่ายภาพนิ่งได้ที่ความละเอียดสูงสุด 640 * 480 พิกเซล

บทที่ 4

การทดลอง และผลการทดลอง

การทดสอบแบ่งออกเป็น 3 ขั้นตอนหลัก ได้แก่ การล็อกอิน (anubislogin) การเพิ่มผู้ใช้งานในระบบ (anubisadd) และการลบผู้ใช้งานในระบบ (anubisdel) โดยมีรายละเอียดในแต่ละขั้นตอนดังต่อไปนี้

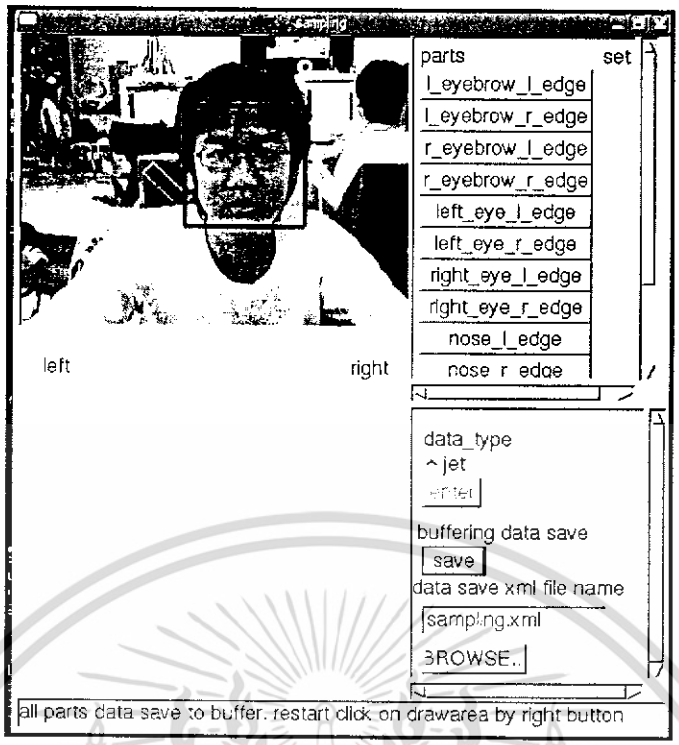
4.1 การเพิ่มชื่อบัญชีผู้ใช้ในระบบ

สำหรับการเพิ่มชื่อบัญชีผู้ใช้ในระบบนั้น ผู้ที่มีสิทธิ์ในการเพิ่มชื่อบัญชีผู้ใช้ในระบบนั้นจะมีเพียงผู้ดูแลระบบหรือ Administrator หรือ root ในระบบปฏิบัติการลินุกซ์เท่านั้น โดยระบบจะสแกนใบหน้าของผู้ใช้ ผ่านทางกล้อง Web Camera

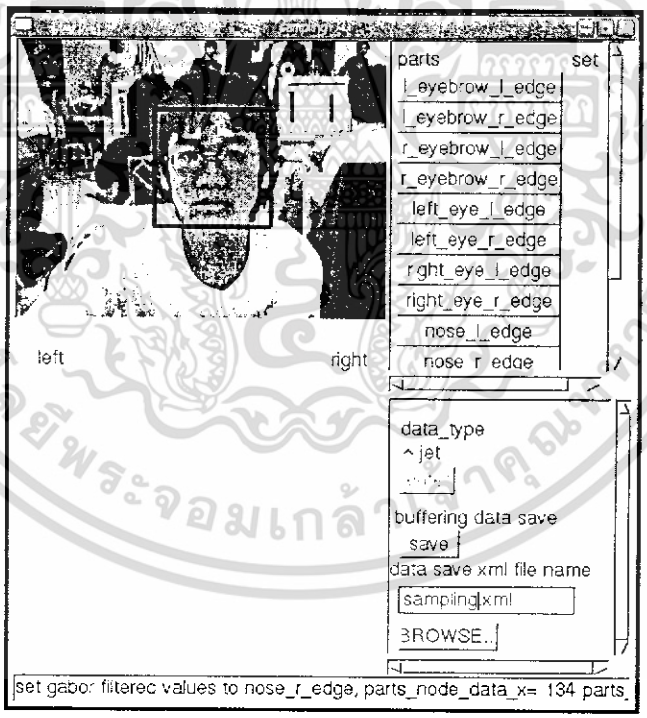
ภาพที่ 4-1 แสดงการใช้คำสั่งเพิ่มชื่อบัญชีผู้ใช้

ดังภาพที่ 4-1 เป็นการใช้งานคำสั่งเพิ่มชื่อบัญชีผู้ใช้ในระบบ โดยพิมพ์คำสั่ง "sudo anubisadd" ตามด้วยชื่อผู้ใช้คนใหม่ที่จะเพิ่มเข้าไปในระบบ ซึ่งระบบจะแสดงไดอะล็อกเพื่อให้กำหนดตำแหน่งของจุดอ้างอิงที่กำหนดให้ เพื่อใช้อ้างอิงในการล็อกอินครั้งต่อ ๆ ไป โดยตำแหน่งจุดอ้างอิงบนใบหน้า มีทั้งหมด 12 จุด ดังนี้

- ขอบหางคิ้วซ้ายของคิ้วซ้าย
- ขอบหางคิ้วขวาของคิ้วซ้าย
- ขอบหางคิ้วซ้ายของคิ้วขวา
- ขอบหางคิ้วขวาของคิ้วขวา
- ขอบตาซ้ายของตาซ้าย
- ขอบตาขวาของตาซ้าย
- ขอบตาซ้ายของตาขวา
- ขอบตาขวาของตาขวา
- ขอบตีนจมูกด้านซ้าย
- ขอบตีนจมูกด้านขวา
- ขอบริมฝีปากซ้าย
- ขอบริมฝีปากขวา



ภาพที่ 4-2 ไดอะล็อกก่อนทำการกำหนดตำแหน่งของจุดอ้างอิง



ภาพที่ 4-3 แสดงไดอะล็อกหลังทำการกำหนดตำแหน่งของจุดอ้างอิง

เมื่อทำการกำหนดตำแหน่งของจุดอ้างอิงเรียบร้อยแล้ว จากนั้นทำการบันทึกข้อมูลของผู้ใช้คนใหม่ลงในฐานข้อมูล ซึ่งอยู่ในรูปแบบไฟล์ชนิด XML

จากนั้นขั้นตอนต่อไป จะต้องทำการระบุชื่อผู้ใช้ เพื่อกำหนดเป็นชื่อบัญชีผู้ใช้ในระบบ ดังภาพที่ 4-4

ภาพที่ 4-4 แสดงไดอะล็อกสำหรับการระบุชื่อผู้ใช้

4.2 การลบชื่อบัญชีผู้ใช้ในระบบ

การลบชื่อบัญชีผู้ใช้ในระบบ ผู้ที่มีสิทธิ์กระทำการดังกล่าว จะมีเพียงผู้ดูแลระบบเท่านั้น ซึ่งระบบจะรับชื่อผู้ใช้ที่ต้องการลบและทำการลบแฟ้มข้อมูลต่าง ๆ ของผู้ใช้คนนั้น

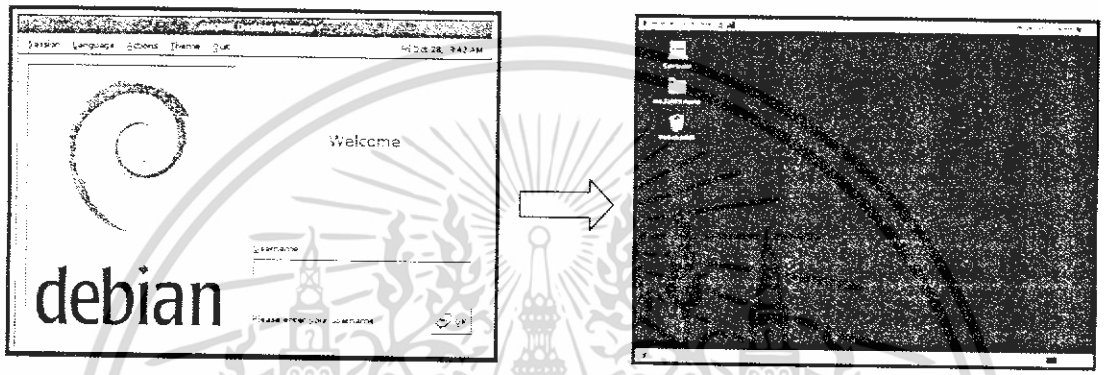
ภาพที่ 4-5 แสดงการใช้คำสั่งลบชื่อบัญชีผู้ใช้

4.3 การตรวจสอบผู้ใช้

ในการตรวจสอบผู้ใช้ มีการทดลองดังต่อไปนี้

- ผู้ใช้งานพิมพ์ชื่อบัญชีผู้ใช้ของตนเอง
- กล้องจะทำการบันทึกใบหน้าของผู้ใช้และนำไปเปรียบเทียบกับข้อมูลใบหน้าของชื่อบัญชีผู้ใช้งานที่ได้รับ ในฐานข้อมูล

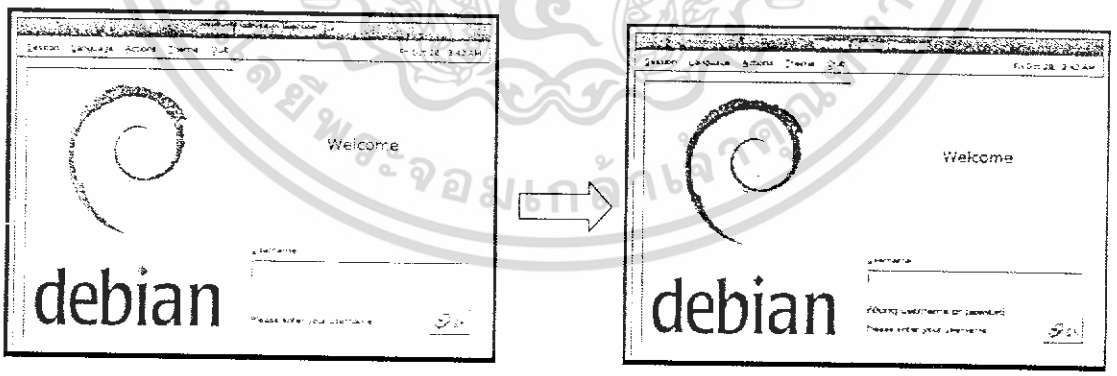
4.3.1 การตรวจสอบใบหน้าสำเร็จ



ภาพที่ 4-7 แสดงผลการตรวจสอบใบหน้าสำเร็จ

เมื่อผลการตรวจสอบใบหน้าสำเร็จแล้ว ระบบจะให้เชลล์ (shell) กับผู้ใช้คนดังกล่าวเพื่อใช้ในการใช้งานต่อไป

4.3.2 การตรวจสอบใบหน้าไม่สำเร็จ



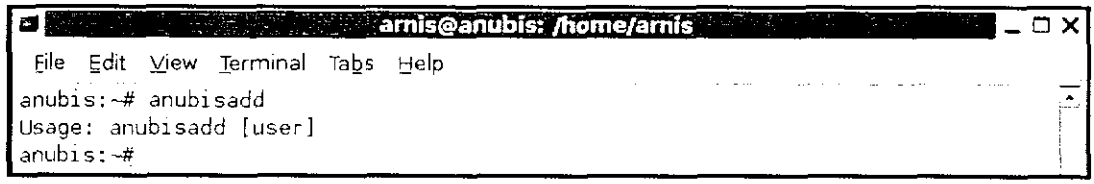
ภาพที่ 4-8 แสดงผลการตรวจสอบใบหน้าไม่สำเร็จ

หากกระบวนการตรวจสอบใบหน้าไม่สำเร็จ ผู้ใช้คนดังกล่าว จะไม่สามารถเข้าใช้งานในระบบได้

4.4 การใช้งานที่ไม่ถูกต้อง

การใช้งานของผู้ดูแลระบบที่ไม่ถูกต้อง จะมีโอกาสเกิดขึ้นได้สองกรณีคือ

4.4.1 เพิ่มชื่อบัญชีผู้ใช้งานในระบบไม่ถูกต้อง แสดงผลดังนี้



ภาพที่ 4-9 แสดงผลของคำสั่งเพิ่มผู้ใช้งานที่ไม่ถูกต้อง

4.4.2 ลบชื่อบัญชีผู้ใช้งานในระบบไม่ถูกต้อง แสดงผลดังนี้

ภาพที่ 4-10 แสดงผลของคำสั่งลบผู้ใช้งานที่ไม่ถูกต้อง

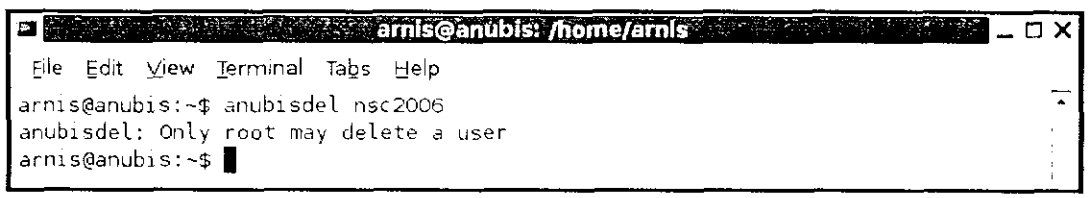
4.5 การเพิ่มหรือลบชื่อบัญชีผู้ใช้โดยไม่มีสิทธิ์ของผู้ดูแลระบบ

ในการเพิ่มหรือลบชื่อบัญชีผู้ใช้ในระบบโดยไม่มีสิทธิ์ของผู้ดูแลระบบในการกระทำดังกล่าวระบบจะแสดงผลลัพธ์ ดังนี้

4.5.1 การใช้คำสั่งเพิ่มชื่อผู้ใช้โดยไม่มีสิทธิ์ของผู้ดูแลระบบ

ภาพที่ 4-11 แสดงการเพิ่มชื่อบัญชีผู้ใช้โดยไม่มีสิทธิ์ของผู้ดูแลระบบ

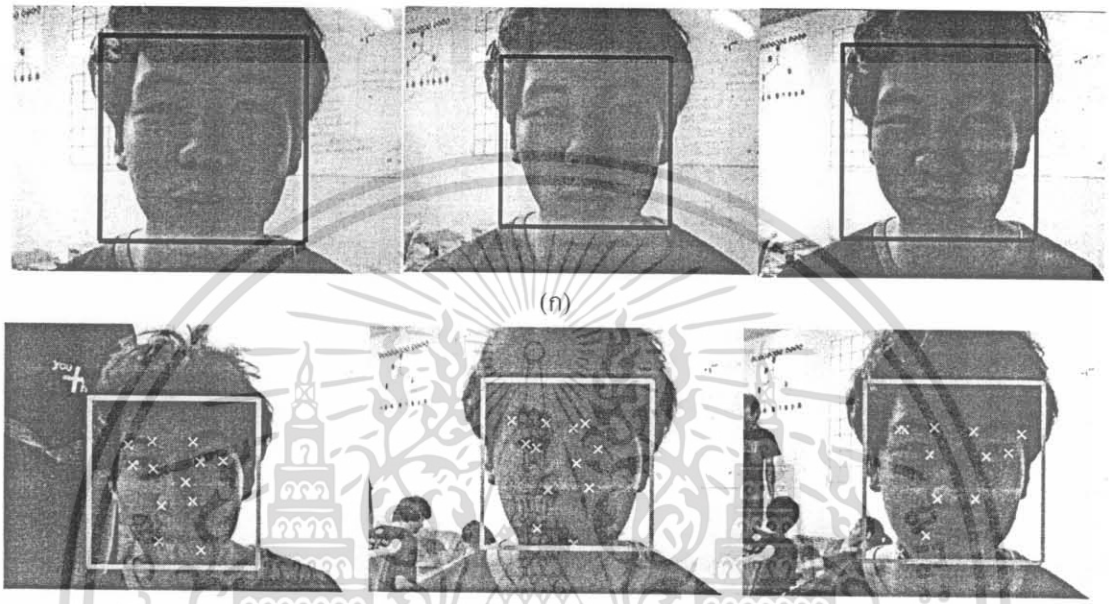
4.5.2 การใช้คำสั่งลบชื่อผู้ใช้โดยไม่มีสิทธิ์ของผู้ดูแลระบบ



ภาพที่ 4-12 แสดงการลบชื่อบัญชีผู้ใช้โดยไม่มีสิทธิ์ของผู้ดูแลระบบ

4.5 ผลการทดลอง

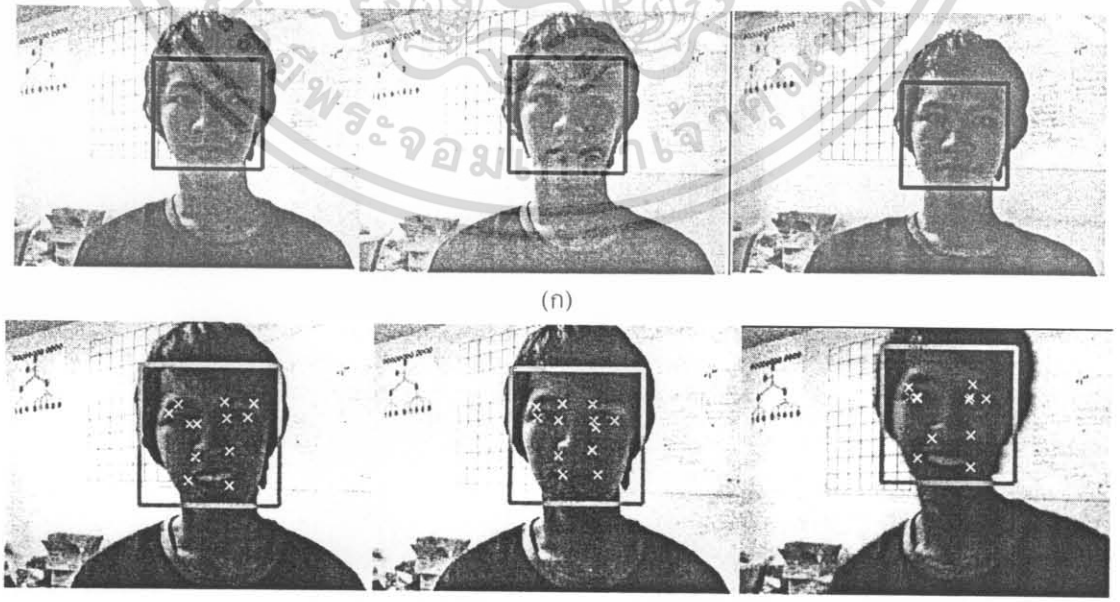
ในการทดลองเพื่อใช้วิเคราะห์ประสิทธิภาพระบบที่ได้พัฒนาขึ้น โดยใช้ข้อมูลใบหน้าของบุคคลที่ลักษณะที่แตกต่างกัน จำนวน 10 คน คนละ 10 ภาพ ในลักษณะท่าทางที่แตกต่างกันออกไป ซึ่งประกอบด้วยข้อมูลที่ได้จากใบหน้าของผู้ชาย 7 คน และผู้หญิง 3 คน ดังแสดงได้ ต่อไปนี้



(ก)

(ข)

ภาพที่ 4-13 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ ก
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ ก ที่สามารถตรวจจับได้

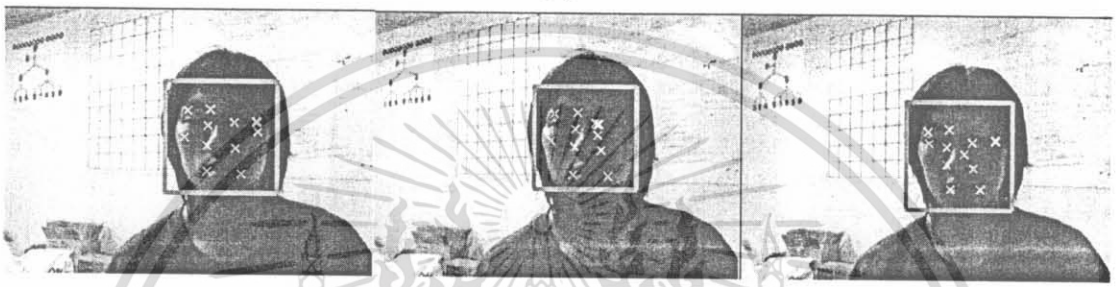


(ก)

(ข)

ภาพที่ 4-14 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ ข
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ ข ที่สามารถตรวจจับได้

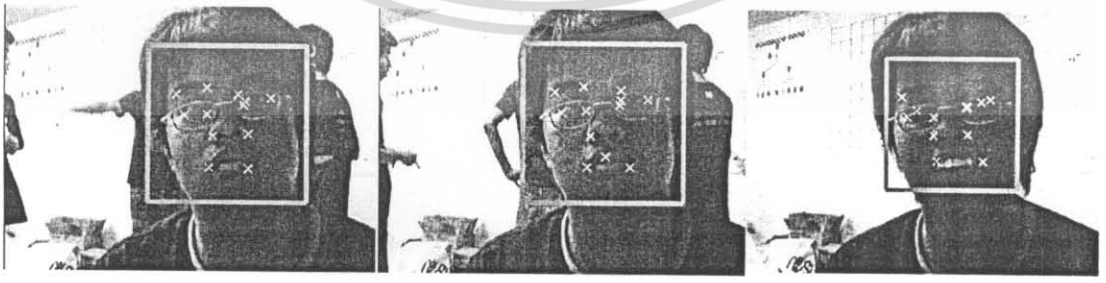
(ก)



(ข)

ภาพที่ 4-15 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ ค
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ ค ที่สามารถตรวจจับได้

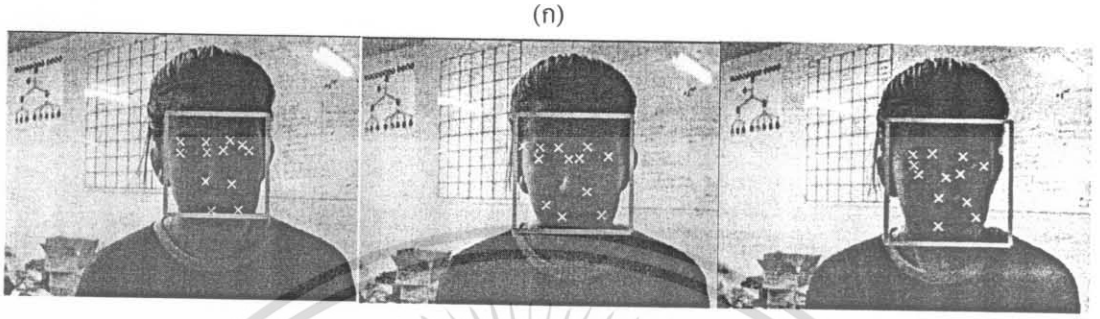
(ก)



(ข)

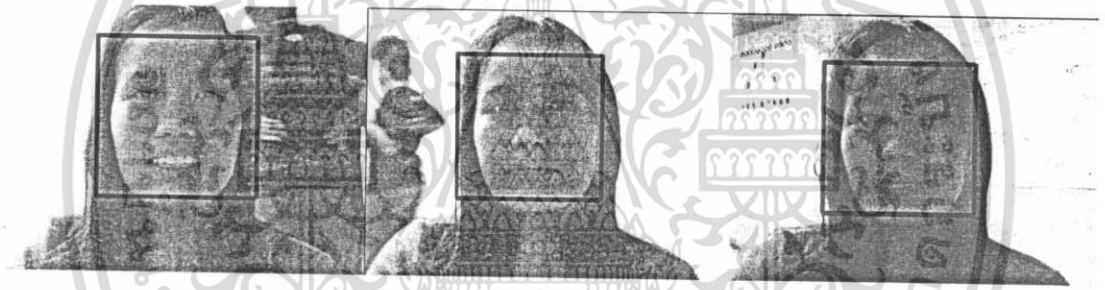
ภาพที่ 4-16 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ ง
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ ง ที่สามารถตรวจจับได้

(ก)



(ข)

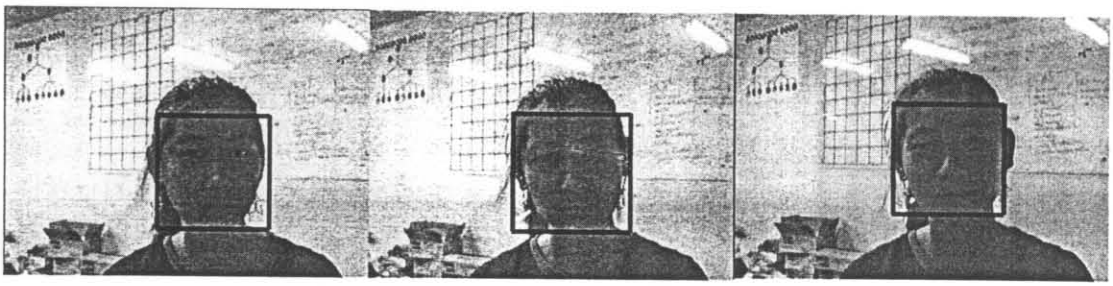
ภาพที่ 4-17 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ จ
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ จ ที่สามารถตรวจจับได้



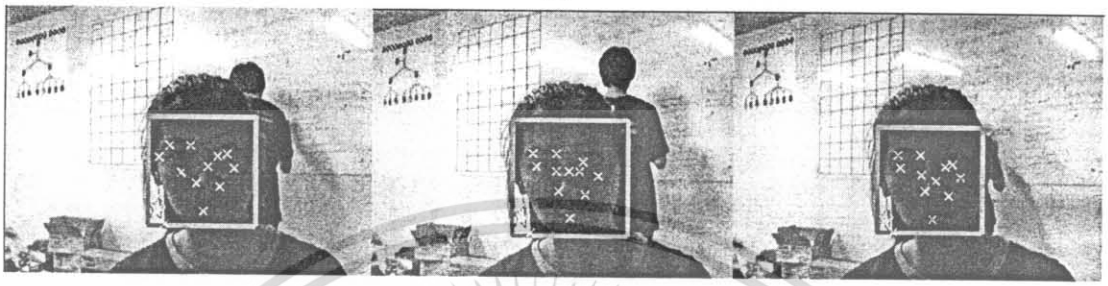
(ก)

(ข)

ภาพที่ 4-18 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ ช
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ ช ที่สามารถตรวจจับได้



(ก)

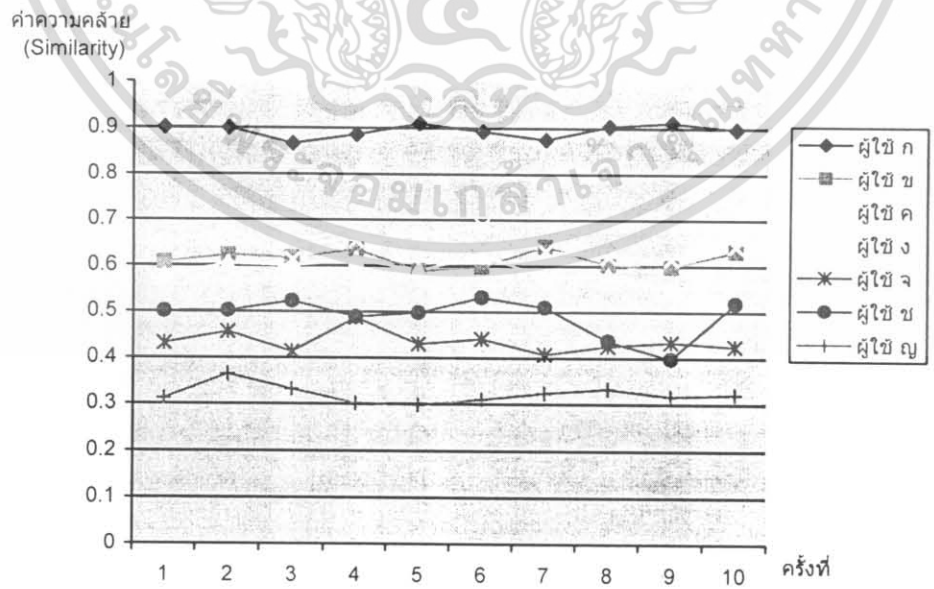


(ข)

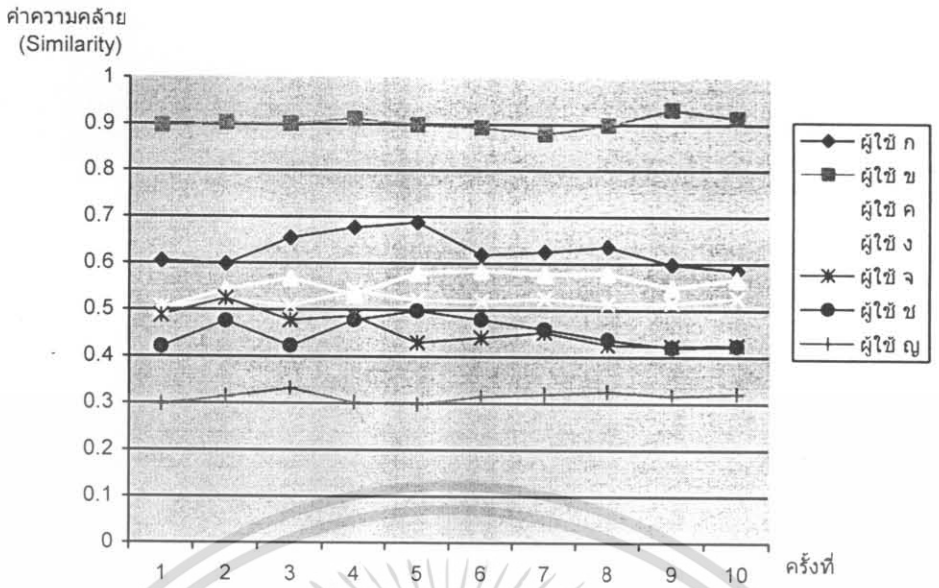
ภาพที่ 4-19 (ก) แสดงส่วนของใบหน้าที่ตรวจจับได้ และการกำหนดจุดอ้างอิงบนใบหน้าผู้ใช้ ญ
(ข) แสดงตำแหน่งจุดอ้างอิงบนใบหน้าผู้ใช้ ญ ที่สามารถตรวจจับได้

โดยอาศัยอัลกอรึธึมการรู้จำใบหน้าจากเฟรมเวิร์ค OpenCV [4] ทำให้เราสามารถหาค่าความคล้ายคลึงได้ (Similarity) โดยที่ค่าความคล้ายคลึงจะอยู่ในช่วง 0-1 ซึ่งถ้าค่าที่ได้เข้าใกล้ 1 มากเท่าใด ความคล้ายคลึงจะมากตามไปด้วย

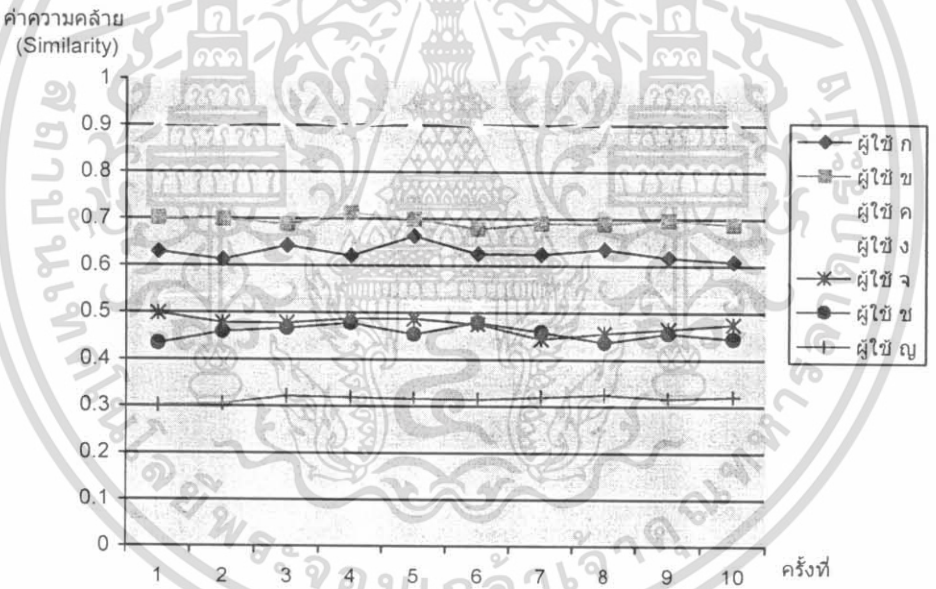
เมื่อนำข้อมูลจากฐานข้อมูลที่เก็บข้อมูลใบหน้าผู้ใช้แต่ละคนมาวิเคราะห์ จะได้กราฟ ดังแสดงต่อไปนี้



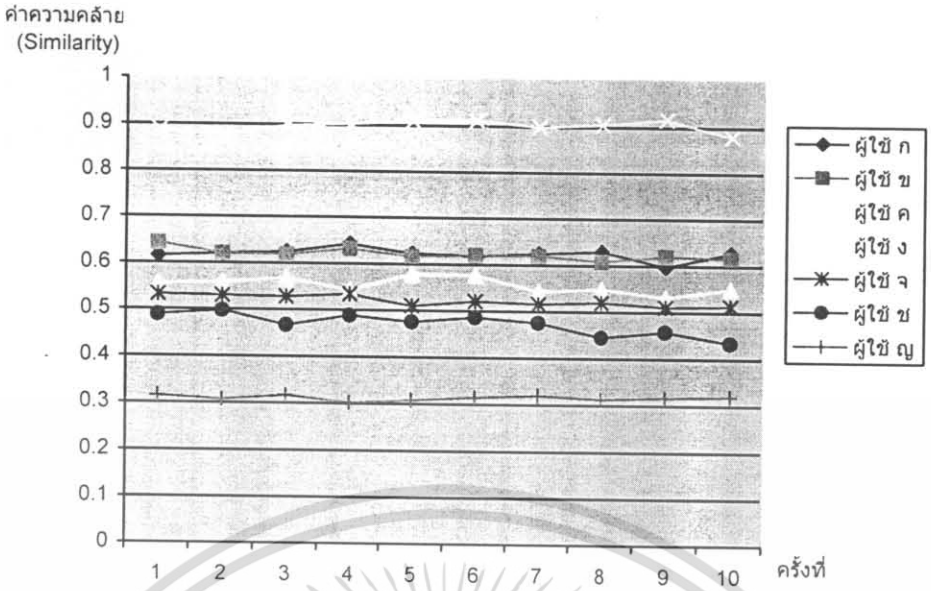
ภาพที่ 4-20 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ ก เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ



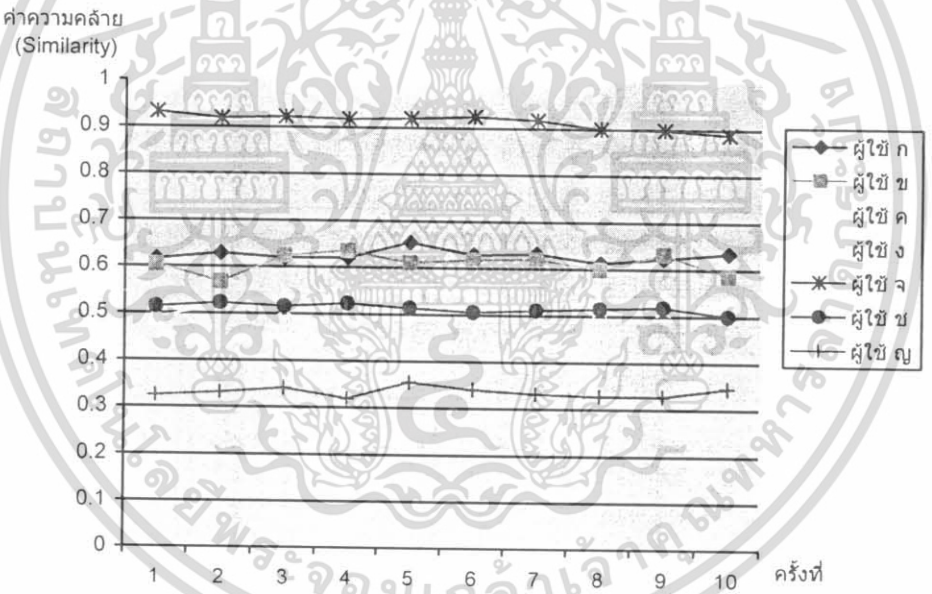
ภาพที่ 4-21 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ ข เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ



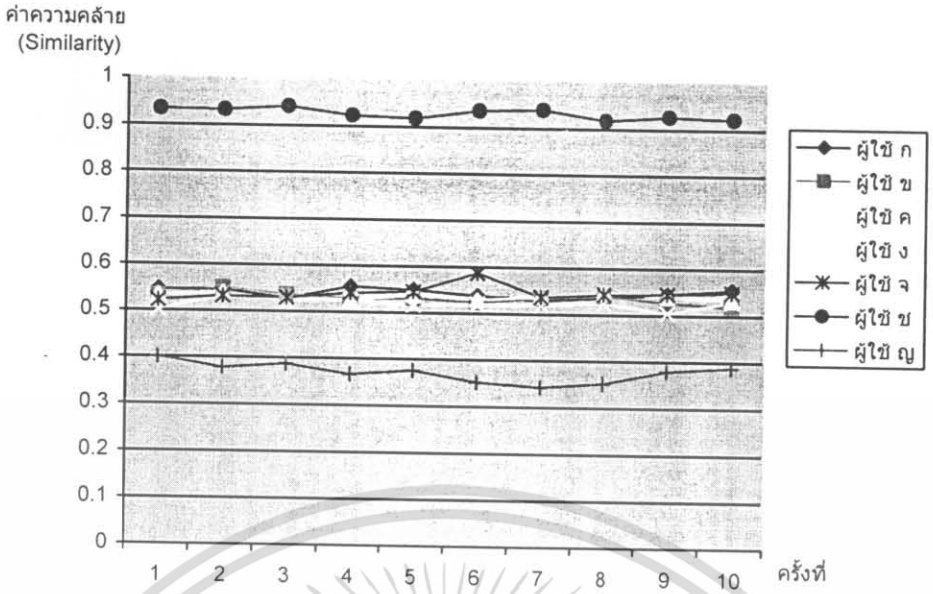
ภาพที่ 4-22 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ ค เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ



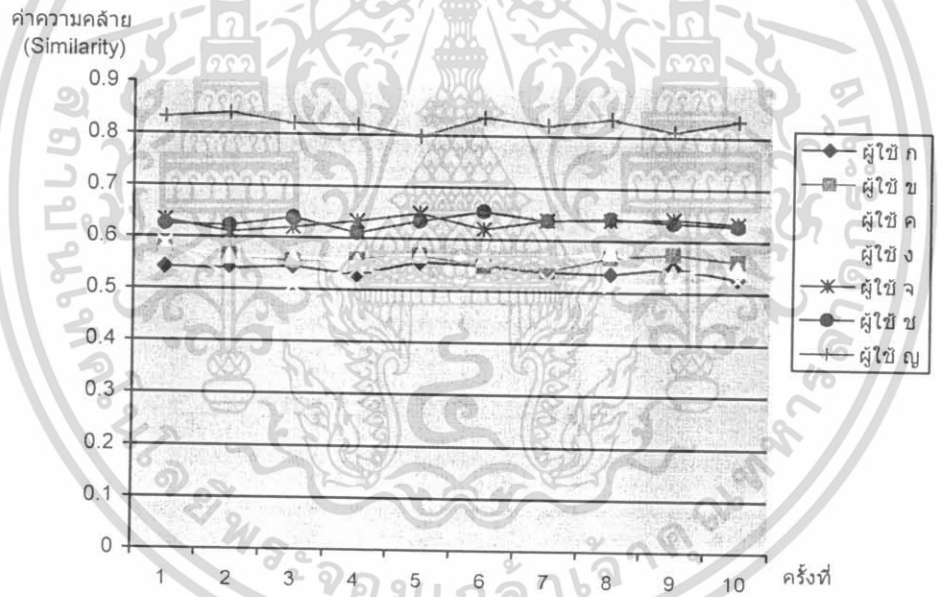
ภาพที่ 4-23 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ ง เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ



ภาพที่ 4-24 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ จ เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ



ภาพที่ 4-25 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ ช เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ



ภาพที่ 4-26 กราฟเส้นแสดงความคล้ายคลึงของใบหน้าผู้ใช้ ญ เมื่อเปรียบเทียบกับใบหน้าผู้ใช้คนอื่น ๆ

จากกราฟจะเห็นได้ว่าแนวโน้มของค่าความคล้ายคลึงของใบหน้าผู้ใช้แต่ละคน จะอยู่ในช่วง 0.3 – 0.95 และแนวโน้มของค่าความคล้ายคลึงของใบหน้าผู้ใช้เอง จะอยู่ในช่วง 0.8 – 0.95

บทที่ 5
สรุปและวิจารณ์

5.1 สรุปผล

จากกราฟรูปที่ จะเห็นได้ว่าค่าความคล้ายคลึง (Similarity) ของใบหน้าของผู้ใช้เองจะมีค่าแตกต่าง และมากกว่าค่าความคล้ายคลึงเมื่อเปรียบเทียบกับใบหน้าของผู้อื่นอย่างชัดเจน

ในปริญญานิพนธ์ฉบับนี้ได้นำเสนอการพัฒนาระบบล็อกอินในการเข้าใช้งานในระบบปฏิบัติการลินุกซ์ โดยใช้ใบหน้าแทนการใช้ยูสเซอร์เนม และรหัสผ่าน ซึ่งในส่วนของการพัฒนาในส่วนระบบล็อกอิน ได้ทำการพัฒนาในส่วนเฟรมเวิร์ค PAM เพื่อให้สามารถรองรับการพิสูจน์ตนด้วยใบหน้าได้ และในส่วนรู้จำใบหน้า พัฒนาด้วยเฟรมเวิร์ค OpenCV และจากผลที่ได้จากการนำไปประยุกต์ใช้งานจริง จะเห็นได้ว่าระบบที่ได้พัฒนาขึ้นสามารถใช้งานได้จริง มีประสิทธิภาพความปลอดภัยในการเข้าใช้งานในระบบเพิ่มขึ้น และการใช้งานของผู้สะดวกสบายเพิ่มมากขึ้นด้วย

5.2 บทวิจารณ์

ปริญญานิพนธ์ฉบับนี้ เมื่อนำมาใช้งานจริงแล้วมีข้อจำกัด ดังต่อไปนี้

1. ระยะของการถ่ายภาพนั้นจะต้องถ่ายให้อยู่ห่างจากกล้องเท่ากับระยะที่กำหนด ซึ่งหากเกินระยะที่กำหนดไว้ จะทำให้รายละเอียดของภาพที่ได้ไม่ดี ทำให้ผลลัพธ์ที่ได้มีความผิดพลาด

2. ต้องถ่ายภาพในสภาวะแสงที่กำหนด คือ ต้องไม่มากหรือน้อยจนเกินไป เพราะปริมาณแสงจะมีผลต่อภาพที่ได้เป็นอย่างมาก

3. การประมวลผลภาพเป็นแบบ 2 มิติ จึงไม่สามารถแยกความแตกต่างระหว่างการถ่ายภาพจริง ๆ กับการนำภาพถ่ายมาวางหน้ากล้อง

4. การถ่ายภาพผู้ใช้แต่ละครั้ง ต้องใช้การกดเมาส์ ซึ่งไม่สะดวกต่อการใช้งานมากนัก

5.3 ปัญหาและอุปสรรคในการพัฒนาโครงงาน

1. เนื่องจากการพัฒนาโครงงานนี้เป็นเทคโนโลยีใหม่ โดยเฉพาะอย่างยิ่งการพัฒนาวิธีชีวมาตร ให้ใช้งานร่วมกับระบบปฏิบัติการลินุกซ์ ยังไม่แพร่หลาย ส่งผลให้ไม่มีเอกสารอ้างอิงที่อธิบายเกี่ยวกับ API ให้ศึกษามากเท่าใดนัก

2. ระบบนี้พัฒนาขึ้นโดยใช้ภาษาซี บนระบบปฏิบัติการลินุกซ์ จึงพบปัญหาในการเขียนโปรแกรมในส่วนที่เกี่ยวข้องกับระบบจดจำใบหน้า โมดูล PAM โปรแกรมการล็อกอิน การเพิ่มและการลบบัญชีผู้ใช้งานในระบบ

3. ไลบรารีที่ใช้ในการพัฒนาโครงงานมีจำนวนมาก เช่น OpenCV GTK IEEE1394 และ Video4linux เป็นต้น ทำให้ต้องใช้เวลาในการศึกษาการพัฒนาโปรแกรมจากไลบรารีข้างต้น

5.4 ข้อเสนอแนะ และแนวทางในการพัฒนาต่อ

1. พัฒนาปรับปรุงอัลกอรึธึมที่ใช้ในการรู้จำใบหน้า เพื่อที่จะสามารถรู้จำใบหน้าได้ดียิ่งขึ้น

2. เพิ่มความสามารถในการปรับขนาดของภาพ ซึ่งจะช่วยเพิ่มความสามารถในการใช้งานซึ่งไม่จำเป็นต้องกำหนดระยะห่างในการถ่ายภาพคงที่ไว้

3. พัฒนาระบบสั่งงานด้วยเสียง หรือใช้ระบบอินฟราเรดในการสั่งการ เพื่อความสะดวกในการใช้งานมากยิ่งขึ้น

บรรณานุกรม

เอกสารอ้างอิงที่เป็นปริญญานิพนธ์

[1] ภรายุทธ วังวรัญญูและสุจิตรา ไพบูลย์วาณิชย์ " **ระบบพิสูจน์ตน**" ปริญญานิพนธ์ วิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2543

เอกสารอ้างอิงที่เป็นหนังสือ

[2] Charlie Kaufman, Radia Perlman and Mike Speciner 2002. "**Network Security Private Communication in a Public world**", Prentice Hall PTR

[3] Paul Reid, "**Biometrics for Network Security**", Prentice Hall PTR.

เอกสารอ้างอิงที่เป็นเว็บไซต์

[4] **"OpenCV – Open Source Computer Vision Library"** [Online].Available: http://www.intel.com/research/mrl/research/opencv/

[5] **"Linux-PAM"** [Online].Available:http://www.kernel.org/pub/linux/libs/pam/.

[6] **"User Authentication HOWTO"** [Online].Available: http://www.linux.com/howtos/User-Authentication-HOWTO/x115.shtml

[7] **"FOCUS on Sun and Linux: Pluggable Authentication Modules, Part II"** [Online].Available: http://www.securityfocus.com/infocus/1390

[8] **"Pluggable Authentication Modules"** [Online].Available: http://en.wikipedia.org/wiki/Pluggable_authentication_modules

[9] **"Sample PAM Application"** [Online].Available: http://www.freebsd.org/doc/en_US.ISO8859-1/articles/pam/pam-sample-appl.html

[10] **"pam_sm_authenticate"** [Online].Available: http://www.opengroup.org/onlinepubs/008329799/pam_sm_authenticate.htm

[11] http://www.questbiometrics.com/images/iris-scanning-biometrics.jpg

[12] http://upload.wikimedia.org/wikipedia/en/6/60/Fingerprintonfinger.JPG

[13] http://www.biobex.com/images/home_13.jpg

[14] http://www.westernconnect.com/images/keystroke.jpg

ภาคผนวก

ขั้นตอนการติดตั้ง

1. ขั้นตอนการกล้อง Web camera

1. ติดตั้ง pwc-source (Philips web camera) โดยสามารถดาวน์โหลดได้ที่ http://www.vanheusden.com/setpwc/setpwc-1.1.tgz หรือจาก Debian package pool (testing) โดยใช้คำสั่ง

```
#apt-get install pwc-source
```

2. เมื่อทำการติดตั้งเรียบร้อยแล้ว จากนั้นต้องทำการ patch version ของ pwc-source โดยสามารถดาวน์โหลดได้จาก http://www.saillard.org/linux/pwc/files/ (ปัจจุบัน คือ เวอร์ชัน pwc-10.0.10-to-10.0.11.patch.bz2)

3. คลายไฟล์ archive ออก โดยใช้คำสั่ง

```
$tar xjvf pwc-10.0.10-to-10.0.11.patch.bz2
$cd pwc-10.0.10-to-10.0.11
```

4. ทำการ patch version ของ pwc-source โดยก่อนที่จะทำการติดตั้ง จำเป็นต้องมี kernel header ของ kernel ของระบบที่ใช้ปัจจุบันก่อน โดยสามารถติดตั้งผ่าน apt-get ของ Debian ได้โดย

```
#apt-get install kernel-headers-$(uname –r)
```

จากนั้น

```
#make
```

ทำการลบ library เก่าทิ้งก่อน

```
#find /lib/modules/`uname –r` /-name "pwc*.ko"
#rm /path/to/that/file
#make install
#cp pwc.ko /path/
```

5. ทำการอัพเดตมอดูลใหม่

```
#depmod –a
#rmmod pwc
#rmmod pwcx
#modprobe pwc power_save=1
```

2. ขั้นตอนการติดตั้งโปรแกรม

1. ต้องติดตั้งกล้อง Web camera เสร็จเรียบร้อยแล้ว

2. คัดลอกไฟล์ anubis.tar.bz2 มาไว้ในไดเร็คทอรีใด ๆ

$cp anubis.tar.bz2 /path/

3. คลายไฟล์ดังกล่าวออก ด้วยการใช้คำสั่ง

$tar xjvf anubis.tar.bz2

4. หลังจากคลายไฟล์ออกแล้ว ติดตั้งโปรแกรม

$cd anubis

$./configure

$make

#make install

5. คัดลอกไฟล์ pam_anubisgdm.so ไปไว้ใน /usr/include/security/

#cp pam_anubisgdm.so /usr/include/security/

6. ปรับแต่งค่าไฟล์ /etc/gdm/gdm.conf ดังภาพ

#vi /etc/gdm/gdm.conf

```
[servers]
# These are the standard servers.  You can add as many you want here
# and they will always be started.  Each line must start with a unique
# number and that will be the display number of that server.  Usually just
# the 0 server is used.
1=Standard vt08
0=Standard vt07
```

7. ปรับแต่งค่าไฟล์ /etc/gdm/Init/Default ดังภาพ

```
#!/bin/sh
# Stolen from the debian kdm setup, aren't I sneaky
# Plus a lot of fun stuff added
#  -George

arcbisrecog -j /root/gabor_jet.xml -f /root/face_jet.xml

PATH=/usr/X11R6/bin:$PATH
OLD_IFS=$IFS

gdmwhich () {
  COMMAND="$1"
  OUTPUT=
  IFS=:
  for dir in $PATH
  do
    if test -x "$dir/$COMMAND" ; then
      if test "x$OUTPUT" = "x" ; then
        OUTPUT="$dir/$COMMAND"
      fi
    fi
  done
  IFS=$OLD_IFS
```

ส่วนของซอฟท์แวร์มีวิธีการใช้งานดังนี้

การใช้งานโปรแกรมนั้นมีขั้นตอนในการใช้งาน แบ่งออกเป็น 3 ขั้นตอนหลักได้แก่

1. การล็อกอิน (login)
2. การเพิ่มผู้ใช้งานระบบ (add user)
3. การลบผู้ใช้งานระบบ (delete user)

ก่อนการใช้งานระบบต้นแบบในการล็อกอินเข้าลินุกซ์ด้วยใบหน้าจะต้องมีการแก้ไขข้อมูลในไฟล์ PAM Configuration ก่อนเพื่อเป็นการเปลี่ยนระบบจากการล็อกอินด้วยการป้อนข้อมูลยูเซอร์เนมและพาสเวิร์ด มาเป็นการตรวจสอบใบหน้า โดยมีขั้นตอนดังนี้ คือ

1. แก้ไขข้อมูลในไฟล์ /etc/pam.d/gdm
2. เพิ่มข้อมูล auth required pam_anubisgdm.so ลงไป
3. คอมเมนท์ที่ @include common-auth โดยทำการใส่ "#" หน้าประโยค ดังรูป

```
File  Edit  View  Terminal  Tabs  Help
#%PAM-1.0
auth     requisite      pam_nologin.so
auth     required       pam_env.so
#include common-auth
auth     required       pam_anubisgdm.so
@include common-account
session required        pam_limits.so
@include common-session
@include common-password
~
```