บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

การพัฒนาระบบรหัสผ่านแบบใช้ครั้งเดียวสำหรับระบบยูนิกซ์ เป็นอีกวิธีหนึ่งที่สามารถป้องกันผลร้ายอันเกิดจากการลักลอบเข้าใช้บริการในระบบจากผู้ไม่มีสิทธิที่ทราบรหัสผ่านของผู้ใช้คนอื่น เนื่องจากระบบที่พัฒนาขึ้นนี้ทำให้รหัสผ่านที่ผู้ไม่มีสิทธิได้รับไปนั้นไม่สามารถนำไปใช้ในครั้งต่อไปได้เพราะระบบจะทำการเปลี่ยนรหัสผ่านของผู้ใช้ทุกครั้งที่ผ่านเข้าใช้ระบบสำเร็จ โดยที่รหัสผ่านใหม่นี้จะมีเพียงผู้ใช้ที่มีสิทธิและระบบเท่านั้นที่ทราบ รหัสผ่านถูกเก็บไว้เป็นฐานข้อมูลบนเครื่องคอมพิวเตอร์ระดับพีซีและถูกพิมพ์ส่งให้แก่ผู้ใช้ การตรวจสอบรหัสผ่านของผู้ใช้จะเกิดบนเครื่องระดับพีซี

เนื่องจากงานวิจัยชิ้นนี้ได้กำหนดให้รหัสผ่านถูกเก็บไว้เป็นฐานข้อมูลบนเครื่องคอมพิวเตอร์ระดับพีซี จึงต้องมีการติดต่อสื่อสารเพื่อขอตรวจสอบรหัสผ่านและบริการอื่นระหว่างโปรแกรมล็อกอินที่ทำงานภายใต้ระบบปฏิบัติการยูนิกซ์และโปรแกรมให้บริการตรวจสอบรหัสผ่านที่ทำงานภายใต้ระบบปฏิบัติการดอส โดยใช้โพรโตคอลทีซีพี/ไอพีในการติดต่อสื่อสาร ลักษณะการติดต่อระหว่างโปรแกรมเป็นแบบผู้ขอรับและผู้ให้บริการ โดยที่โปรแกรมล็อกอินเป็นส่วนขอรับบริการและโปรแกรมที่ทำงานภายใต้ระบบปฏิบัติการดอสจะเป็นส่วนของให้บริการ

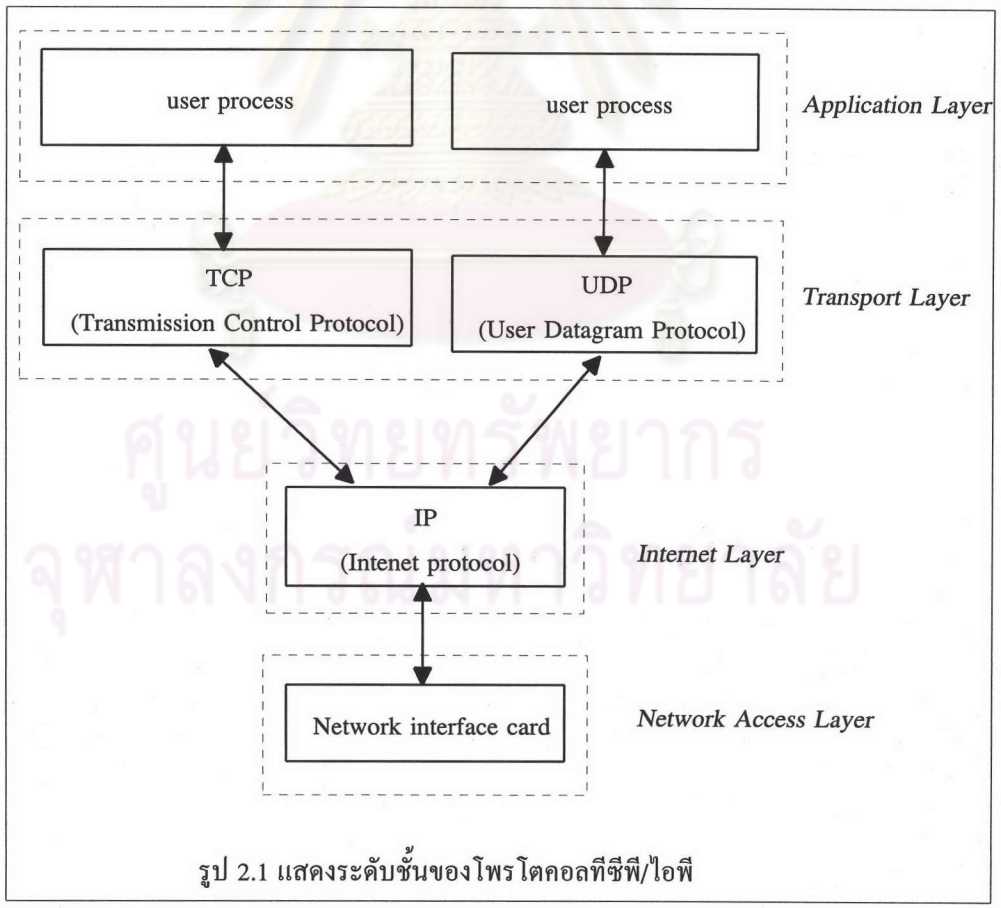
ในส่วนของผู้ให้บริการประกอบด้วยการทำงานหลายประเภทที่ถูกออกแบบให้ทำงานพร้อมกัน ทฤษฎีสำคัญที่เกี่ยวข้องคือ การเขียนโปรแกรมเรสซิเดนต์ (resident program) หรือโปรแกรมประเภทฝังตัวในหน่วยความจำ (TSR : Terminate and Stay Resident)

ในบทนี้กล่าวถึง โพรโตคอลทีซีพี/ไอพี โปรแกรมล็อกอิน ขั้นตอนการทำงานของโปรแกรมล็อกอิน และโปรแกรมเรสซิเดนต์

โพรโตรคอลทีซีพี/ไอพี

โพรโตคอลทีซีพี/ไอพี (TCP/IP, Transmission Control Protocol / Internet Protocol) เป็นโพรโตรคอลที่ถูกพัฒนาขึ้นโดย Advance Research Project Agency (ARPA) ได้รับการยอมรับให้เป็นมาตราฐานในระบบการสื่อสารข้อมูล เครือข่ายอินเตอร์เน็ต (Internet) ได้ใช้โพรโตคอลนี้เป็นหลักในการติดต่อสื่อสารเนื่องจากเป็นโพรโตคอลที่สามารถเชื่อมต่อระหว่างคอมพิวเตอร์ต่างยี่ห้อได้

โพรโตคอลทีซีพี/ไอพี หมายถึง ชุดของโปรแกรมที่ให้บริการในเครือข่าย เช่น วิธีในการรับส่งข้อมูล การจัดการกับสิ่งผิดพลาดระหว่างการติดต่อสื่อสาร การควบคุมให้เกิดความถูกต้องระหว่างการรับส่งข้อมูล เมื่อเรากล่าวถึง โพรโตคอลทีซีพี/ไอพี มิได้หมายถึงเฉพาะโพรโตคอลทีซีพีและโพรโตคอลไอพีเท่านั้น แต่ยังรวมถึงโพรโตคอลอื่น ๆ ที่เกี่ยวข้องด้วย



รูป 2.1 แสดงระดับชั้นของโพรโตคอลทีซีพี/ไอพี

โพรโตคอลทีซีพี (TCP, Transmission Control Protocol) ซึ่งเทียบได้กับระดับชั้นทรานสปอร์ต (transport layer) ของมาตราฐานการสื่อสารข้อมูลโอเอสไอ (OSI model) ทีซีพีจะเป็นตัวจัดการให้บริการส่งข้อมูลแบบรับส่งได้ทั้งไปและกลับพร้อมกันอย่างเชื่อถือได้แก่โพรโตคอลในระดับที่สูงขึ้นไป เอื้ออำนวยให้แต่ละโปรเซส (process) ที่อยู่บนเครื่องเดียวกันหรือต่างเครื่องกันสามารถส่งผ่านข้อมูลระหว่างกันได้ โดยที่ทีซีพีทำงานแบบ connection-oriented หมายถึง การที่คู่ของการสื่อสารข้อมูลต้องมีการตกลงจัดตั้งช่องทางการติดต่อสื่อสาร (port) ระหว่างกันให้เรียบร้อยก่อนที่จะเริ่มทำการสื่อสารข้อมูล

โพรโตคอลยูดีพี (UDP, User Datagram Protocol) เทียบกับระดับชั้นทรานสปอร์ตของมาตราฐานการสื่อสารข้อมูลโอเอสไอ โพรโตคอลยูดีพีต่างจากทีซีพีที่ข้อมูลที่ทำการส่งออกไปอาจมีการสูญหายไม่สามารถไปถึงจุดหมายที่ต้องการ หรือ กล่าวได้ว่าเป็นการทำงานแบบคอนเนคชั่นเลส (connectionless)

งานวิจัยชิ้นนี้ได้ใช้โพรโตคอลยูดีพีเป็นโพรโตคอลในการติดต่อสื่อสารในระดับชั้นทรานสปอร์ต เนื่องจากมีรูปแบบการติดต่อสื่อสารระหว่างกันเป็นแบบสอบถามตอบสนอง (query-response) คือ เมื่อมีฝ่ายหนึ่งส่งกลุ่มข้อมูล (packet) ออกไปจะต้องทำการรอจนกว่ามีกลุ่มข้อมูลตอบกลับมาจากอีกฝ่ายหนึ่ง ถ้าทำการรอนานกว่าช่วงเวลาหนึ่งที่กำหนดไว้ต้องทำการส่งกลุ่มข้อมูลนั้นซ้ำออกไปอีกครั้ง

อีกเหตุผลหนึ่งที่ใช้โพรโตคอลยูดีพี คือ กลุ่มข้อมูลที่ทำการส่งมีขนาดเล็ก เนื่องจากโพรโตคอลยูดีพีมีการทำงานแบบคอนเนคชั่นเลส หมายถึง การที่คู่ของการสื่อสารข้อมูลไม่ต้องมีการตกลงจัดตั้งช่องทางการติดต่อสื่อสารระหว่างกันให้เรียบร้อยก่อนที่จะเริ่มทำการสื่อสารข้อมูลจึงไม่จำเป็นต้องเสียเวลาในการส่งข้อมูลที่ใช้ในการตกลงช่องทางการสื่อสารออกไป โพรโตคอลยูดีพีถือว่าข้อมูลส่วนนี้เป็นข้อมูลที่สูญเปล่าไม่จำเป็นต้องส่ง (overhead) ดังนั้นเมื่อมีข้อมูลสูญหายระหว่างการส่งจึงต้องมีการส่งซ้ำข้อมูลและการส่งซ้ำกลุ่มข้อมูลที่มีขนาดเล็กจะกินเวลาน้อยกว่าการส่งข้อมูลใช้ในการตกลงช่องทางการสื่อสาร

โพรโตคอลไอพี (IP, Internat Protocol) เทียบได้กับระดับชั้นเน็ตเวิร์ค (network layer) ของมาตราฐานสื่อสารข้อมูลโอเอสไอ ทำการสนับสนุนการส่งผ่านข้อมูลในแบบคอนเนคชั่นเลส โดยการแบ่งข้อมูลออกเป็นกลุ่มเล็ก ๆ เรียกว่า ดาต้าแกรม (datagram) จำนวนมากแล้วทยอยส่ง

ออกไป โดยแต่ละกลุ่มข้อมูลอาจถูกส่งไปถึงยังจุดหมายปลายทางโดยใช้เส้นทางที่ไม่ซ้ำกัน จึงอาจทำให้มีการสูญหายของข้อมูลระหว่างทางได้ หรือมีการซ้ำซ้อน หรือได้รับข้อมูลผิดลำดับ ปัญหาเหล่านี้จะถูกควบคุมโดยโพรโตคอลที่อยู่ในระดับเหนือกว่า

เน็ตเวิร์คอินเตอร์เฟสการ์ด (network interface card) การนำเอาเครื่องในระดับไมโครคอมพิวเตอร์มาดัดแปลงให้ทำการติดต่อเข้ากับระบบเครือข่ายที่มีโพรโตคอลทีซีพี/ไอพีนั้น จะต้องเพิ่มอุปกรณ์พิเศษคือ เน็ตเวิร์คอินเตอร์เฟสการ์ดเป็นอุปกรณ์สำหรับทำงานในระดับชั้นฟิสิคอล โปรแกรมที่ใช้ติดต่อสื่อสารกับเน็ตเวิร์คอินเตอร์เฟสการ์ด คือ แพ็กเก็ตไดรเวอร์ (packet driver) ซึ่งเป็นข้อกำหนดที่ใช้โดยทั่วไปสำหรับโปรแกรมประยุกต์ที่ใช้โพรโตคอลทีซีพี/ไอพี โดยจะเป็นโปรแกรมประเภทฝังตัวในหน่วยความจำในระบบปฏิบัติการดอส ซึ่งโปรแกรมประยุกต์ต่าง ๆ จะสามารถขอใช้บริการจากแพ็กเก็ตไดรเวอร์

โปรแกรมล็อกอิน

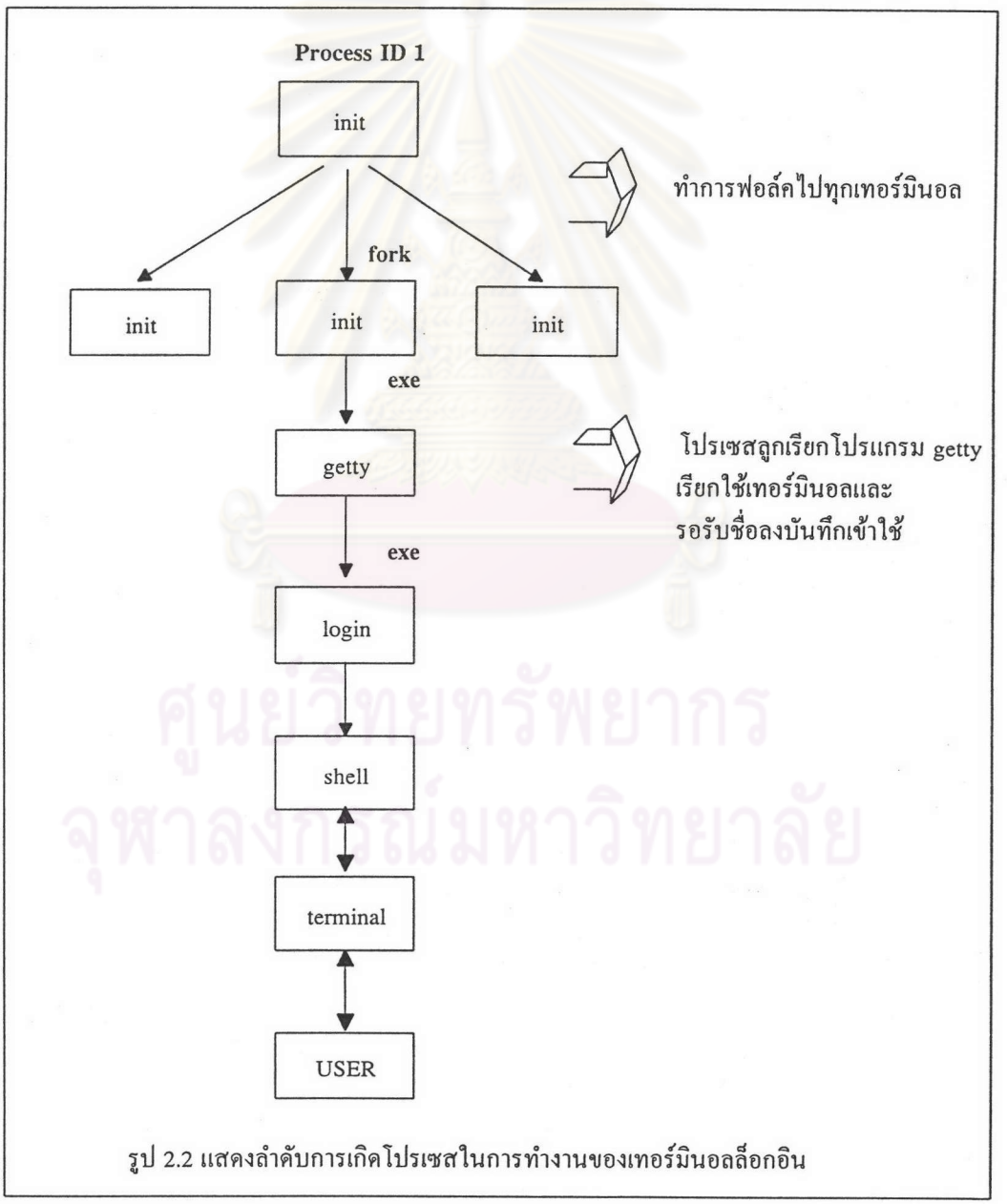
สามารถแบ่งออกได้เป็น 2 ประเภทคือ เทอร์มินอลล็อกอิน และเน็ตเวิร์คล็อกอิน

1. เทอร์มินอลล็อกอิน (terminal login) เมื่อระบบยูนิกซ์เริ่มทำงาน เคอร์เนล (kernel) จะทำการเรียกโปรแกรม /etc/init เพื่อทำการสร้างโปรเซส init โดยโปรเซสนี้จะอนุญาตให้ผู้ใช้หลาย ๆ คนสามารถเข้ามาใช้งานในระบบได้ในเวลาเดียวกัน ขั้นตอนการทำงานของโปรเซส init ขึ้นกับชนิดของระบบยูนิกซ์

ก. ในระบบยูนิกซ์ BSD 4.3 โปรเซส init เรียกใช้งานแฟ้มข้อมูล /etc/rc เพื่อเริ่มการทำงานของโปรเซสเดมอนต่าง ๆ (daemon process) จากนั้นโปรเซส init ทำการอ่านแฟ้มข้อมูล /etc/ttys ที่แต่ละบรรทัดของแฟ้มข้อมูลนี้แสดงค่าพารามิเตอร์ของเทอร์มินอล (terminal) แต่ละเครื่องที่อนุญาตให้ต่อเข้ามาในระบบได้

ข. ในระบบยูนิกซ์ System V โปรเซส init อ่านแฟ้มข้อมูล /etc/inittab ซึ่งระบุค่าเริ่มต้นต่าง ๆ ให้แก่ระบบ จากนั้นเรียกใช้งานแฟ้มข้อมูล /etc/rc เพื่อเริ่มการทำงานของโปรเซสเดมอนต่าง ๆ โปรเซส inittab เป็นแฟ้มข้อมูลที่ระบุว่าเทอร์มินอลเครื่องใดพร้อมใช้งานได้

ต่อจากนั้นโปรเซส init จะทำการฟอล์ก (fork) ตัวเองตามจำนวนของเทอร์มินอลที่ได้กำหนดไว้ และแต่ละโปรเซสที่ถูกฟอล์กออกมาจะทำการเรียกใช้โปรแกรม getty เพื่อทำการเรียกใช้เทอร์มินอล เนื่องจากระบบปฏิบัติการยูนิกซ์ถือว่าอุปกรณ์ทุกชนิดในระบบเป็นไฟล์การเรียกใช้เทอร์มินอลจึงต้องทำการเปิดก่อน เพื่อทำให้สามารถเขียนหรืออ่านจากเทอร์มินอลได้จากนั้นโปรแกรม getty จะขึ้นข้อความให้รู้ว่าพร้อมจะทำการเข้าสู่ระบบและรอจนกว่าจะมีผู้ใช้ป้อนชื่อลงบันทึกเข้าใช้ เพื่อขอเข้าใช้บริการต่อระบบแล้วจึงทำการเรียกใช้โปรแกรมล็อกอิน ส่วนขั้นตอนการทำงานของโปรแกรมล็อกอินจะกล่าวในภายหลังของบทนี้

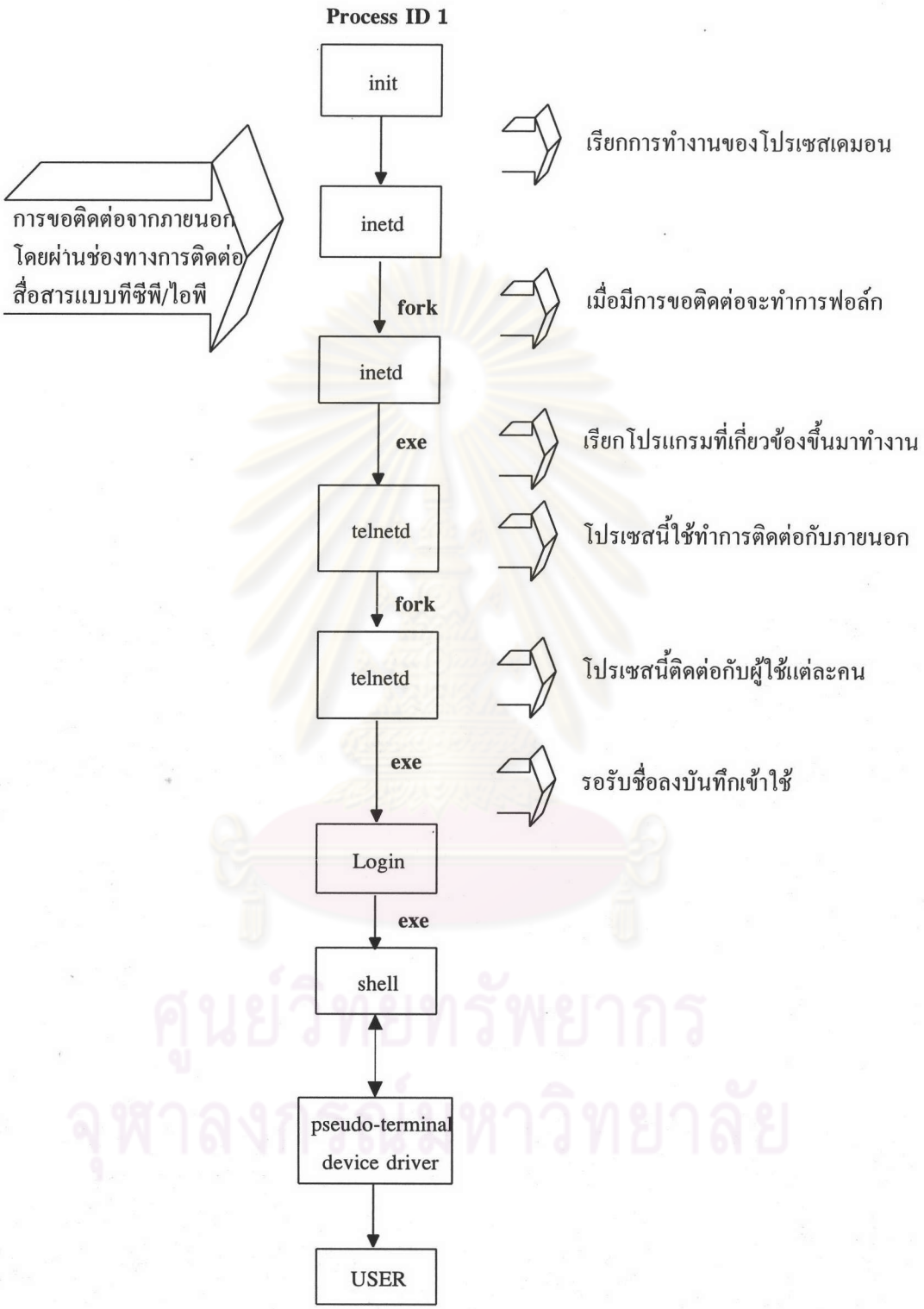


รูป 2.2 แสดงลำดับการเกิดโปรเซสในการทำงานของเทอร์มินอลล็อกอิน

2. เน็ตเวิร์คล็อกอิน (network login) ในกรณีของเทอร์มินอลล็อกอินทำให้เราทราบว่ามีอุปกรณ์หรือเทอร์มินอลเครื่องใดบ้างที่ถูกระบุให้พร้อมใช้ในการเข้าสู่ระบบ แต่ในกรณีของการใช้โปรแกรมล็อกอินผ่านเครือข่ายนั้นเป็นการขอเข้าใช้ระบบจากอุปกรณ์ภายนอก ซึ่งจะไม่มีทางทราบได้เลยว่าเมื่อใดจะมีการขอเข้าใช้ระบบ แนวทางแก้ไขที่ใช้อยู่ในปัจจุบัน คือ แทนที่จะมีโปรเซสที่รอให้ผู้ใช้ป้อนชื่อลงบันทึกเข้าใช้และทำงานขั้นต่อไป แต่เปลี่ยนเป็นโปรเซสที่รอการติดต่อมาจากภายนอกซึ่งโปรเซสนี้ คือ โปรเซส inetd

ในรูปที่ 2.3 แสดงลำดับการเกิดโปรเซสในการทำงานของเน็ตเวิร์คล็อกอิน เมื่อระบบยูนิกซ์เริ่มการทำงาน โปรเซส init เรียกใช้งานแฟ้มข้อมูล /etc/rc เพื่อเริ่มการทำงานของโปรเซสเดมอน ซึ่งหนึ่งในนั้นคือโปรเซส inetd มีหน้าที่คอยการติดต่อโดยมีโพรโตคอลทีซีพี/ไอพีเป็นโพรโตคอลในการติดต่อสื่อสาร เมื่อมีการขอติดต่อจากภายนอกเข้ามาโปรเซส inetd ทำการฟอล์กและเรียกใช้โปรแกรมที่เกี่ยวข้องตามที่ระบุจากช่องทางการสื่อสาร เช่น โปรแกรมฟิงเกอร์ (finger) ใช้ช่องทางการสื่อสารที่ 79 โปรแกรมเทลเนท (telnet) ใช้ช่องทางการสื่อสารที่ 23

โปรเซส telnetd ทำการเปิด pseudo-terminal และทำการฟอล์กออกเป็น 2 โปรเซส โดยโปรเซสหนึ่งจัดการเรื่องการติดต่อระหว่างเครื่องแม่ข่ายกับภายนอก ส่วนอีกโปรเซสทำการรอจนกว่าจะมีผู้ใช้ป้อนชื่อลงบันทึกเข้าใช้แล้วจึงทำการเรียกใช้โปรแกรมล็อกอิน



รูป 2.3 แสดงลำดับการเกิดโปรเซสในการทำงานของเน็ตเวิร์คล็อกอิน

ขั้นตอนการทำงานของโปรแกรมล็อกอิน

Richard (1990) ได้อธิบายขั้นตอนการทำงานของโปรแกรมล็อกอินไว้ดังนี้

1. โปรแกรมล็อกอินทำการเรียกใช้ซิสเต็มคอล (system call) getpwnam( ) ซึ่งเป็นบริการหนึ่งของระบบยูนิกซ์ เพื่อดึงข้อมูลของผู้ใช้จากแฟ้มข้อมูล /etc/passwd ซึ่งประกอบด้วย ชื่อลงบันทึกเข้าใช้ (login name) รหัสผ่านที่ถูกเข้ารหัสไว้ (encrypted password) หมายเลขของผู้ใช้ (user id) หมายเลขกลุ่มผู้ใช้ (group id) ไดเร็คทรอรี่ที่เข้าใช้งาน (login directory) และโปรแกรมเชลล์ (shell) เป็นต้น

2. ตรวจสอบรหัสผ่านของผู้ใช้โดยทำการเรียกใช้ซิสเต็มคอล getpass( ) เพื่อให้ผู้ใช้ป้อนรหัสผ่านและใช้ฟังก์ชั่น crypt( ) ทำการเข้ารหัสรหัสผ่านที่ผู้ใช้ป้อนเข้ามา จากนั้นนำผลลัพท์ที่ได้จากการเข้ารหัสมาเปรียบเทียบกับฟิลด์ pw-passwd ในแฟ้มข้อมูล /etc/passwd

3. กรณีที่ผู้ใช้ป้อนรหัสผ่านที่ผิดแก่ระบบหลายครั้ง โปรแกรมล็อกอินจะทำการเรียกใช้ซิสเต็มคอล exit() ทำให้โปรเซส init ของเทอร์มินอลเครื่องนี้จบการทำงานลง จากนั้นโปรเซส init ตัวแรกของระบบที่มีหมายเลขโปรเซส (process id) เท่ากับ 1 จะฟอล์กตัวเองมายังเทอร์มินอลเครื่องนี้และทำการเรียกโปรเซสต่าง ๆ ตามลำดับเพื่อให้เทอร์มินอลเครื่องนี้พร้อมทำงานต่อไป

4. กรณีที่ผู้ใช้ป้อนรหัสผ่านที่ถูกต้อง โปรแกรมล็อกอินเรียกใช้ซิสเต็มคอล chdir( ) เพื่อติดตั้งไดเร็คทรอรี่ที่เข้าใช้งานให้เท่ากับฟิลด์ login directory จากนั้นตั้งค่าหมายเลขของกลุ่มผู้ใช้และหมายเลขของผู้ใช้และทำการเรียกใช้โปรแกรมเชลล์ ต่อจากนั้นการทำงานจะอยู่ภายใต้โปรเซสเชลล์ที่กำหนดไว้ โดยมีโปรเซสพ่อ (parent process) คือ โปรเซส init ที่มีหมายเลขโปรเซสเท่ากับ 1 ถ้ามีการออกจากระบบโดยจบการทำงานของโปรเซสเชลล์ทำการส่งสัญญาน (signal) SIGCHLD ออกไป โปรเซส init ตัวแรกของระบบจะทำการฟอล์กอีกครั้งและทำการเรียกโปรเซสต่าง ๆ เพื่อให้เทอร์มินอลเครื่องนี้พร้อมทำงานต่อไป

โปรแกรมเรสซิเดนต์

ในระบบปฏิบัติการดอสอนุญาตให้มีเพียงหนึ่งโปรเซสที่สามารถทำงานหรือขอใช้บริการจากซีพียูในขณะใดขณะหนึ่ง เมื่อออกแบบให้มีการทำงานหลายงานพร้อมกันจึงต้องมีการสลับการทำงานระหว่างสองโปรเซส โดยโปรเซสที่ทำงานด้านหน้าหรือ โฟว์กราวน์โปรเซส (foreground process) เป็นโปรเซสที่ทำงานอยู่ตลอดเวลา และโปรเซสทำงานเบื้องหลัง หรือ แบ็คกราวน์โปรเซส (backgroud process) เป็นโปรเซสที่จะเริ่มทำงานต่อเมื่อมีการส่งอินเตอร์รัพต์เข้ามายังซีพียู โดยต้องเป็นอินเตอร์รัพต์ที่เรากำหนดให้มาปลุกการทำงานของแบ็คกราวน์โปรเซส

อินเตอร์รัพต์ (interrupt) คือ สัญญานที่ใช้รับส่งระหว่างอุปกรณ์กับซีพียูเมื่อต้องการสั่งให้อุปกรณ์ทำงาน หรือแจ้งข้อมูลต่าง ๆ ที่เกิดจากอุปกรณ์มายังซีพียู ในระบบปฏิบัติการดอสซีพียูมีหน้าที่คอยให้บริการแก่อุปกรณ์ที่ร้องขอเข้ามา เมื่ออุปกรณ์เกิดการทำงานจะส่งอินเตอร์รัพต์ไปยังซีพียูและที่ซีพียูจะมีตัวชี้บ่งโปรแกรมที่รองรับการทำงานนั้นไว้ในตารางอินเตอร์รัพต์ (interrupt vector) ซีพียูจะทำงานตามหน้าที่ของอินเตอร์รัพต์ที่ส่งเข้ามา

การที่เรียกโปรแกรมประเภทนี้ว่า โปรแกรมประเภทฝังตัวในหน่วยความจำ เพราะเมื่อเลิกใช้งานแล้วจะยังคงฝังตัวอยู่ในหน่วยความจำไม่ได้คืนหน่วยความจำที่โปรแกรมใช้งานอยู่ และเมื่อได้รับอินเตอร์รัพต์ที่กำหนดไว้จะปลุกการทำงานของโปรแกรมขึ้นมาใหม่ ลักษณะเป็นแบบเดียวกับเดมอนโปรเซสในระบบยูนิกซ์

ในบทนี้ได้นำเสนอทฤษฎีสำคัญที่เกี่ยวข้องต่องานวิจัย บทต่อไปกล่าวถึงการออกแบบการทำงานของระบบการให้บริการรหัสผ่านแบบใช้ครั้งเดียวสำหรับระบบยูนิกซ์