ตัวตรวจจับความปลอดภัยการร่วมใช้ทรัพยากร

Resource Sharing Security Scanner

นายยุทธพงษ์ มานุษยานนท์ 41013543

นายสุริยา จินตกานนท์ 41013558

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ตัวตรวจจับความปลอดภัยการร่วมใช้ทรัพยากร

Resource Sharing Security Scanner

โดย

นายยุทธพงษ์ มานุษยานนท์ 41013543

นายสุริยา จินตกานนท์ 41013558

อาจารย์ที่ปรึกษา

อาจารย์ ธนา หงษ์สุวรรณ

อาจารย์ อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตัวตรวจจับความปลอดภัยการร่วมใช้ทรัพยากร

Resource Sharing Security Scanner

ผู้จัดทำ

1. นายยุทธพงษ์ มานุษยานนท์ รหัส 41013543

2.นายสุริยา จินตกานนท์ รหัส 41013558

_____ อาจารย์ที่ปรึกษา

(อาจารย์ ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระภูพงษ์)

ตัวตรวจจับความปลอดภัยการร่วมใช้ทรัพยากร

นายยุทธพงษ์ มานุษยานนท์
นายสุริยา จินตกานนท์
อ. ธนา หงษ์สุวรรณ
อ. อัครเดช วัชระภูพงษ์
ปีการศึกษา 2543

บทคัดย่อ

การใช้ทรัพยากรร่วมกันในระบบเครือข่ายคอมพิวเตอร์มีข้อดีในด้านการแลกเปลี่ยนข้อมูลและความประหยัด แต่เมื่อมีการใช้ทรัพยากรร่วมกันก็ต้องมีการตรวจสอบดูแลความปลอดภัยของการใช้ทรัพยากรนั้น เพราะหากไม่มีการดูแลความปลอดภัยที่ดีพอ อาจมีผู้ไม่ประสงค์ดีหรือผู้ที่รู้เท่าไม่ถึงการณ์เข้ามาใช้ทรัพยากร และก่อให้เกิดความเสียหายแก่ทรัพยากรได้

ระบบเครือข่ายของวินโดวส์และระบบเครือข่ายของยูนิกซ์นับเป็นระบบเครือข่ายที่นิยมใช้งานกันอย่างแพร่หลายและมีการใช้ทรัพยากรร่วมกัน โดยสามารถแชร์ทรัพยากรได้ทั้งไดเร็กทอรีและเครื่องพิมพ์ ในระบบเครือข่ายทั้งสองมีระดับการรักษาความปลอดภัยของการใช้ทรัพยากรร่วมกันให้ผู้ใช้เลือกใช้ได้หลายรูปแบบ ขึ้นอยู่กับผู้ใช้ซึ่งมีหน้าที่ในการปรับแต่งและตรวจสอบระบบรักษาความปลอดภัยเพื่อป้องกันความสูญเสียที่อาจเกิดขึ้น

ในโครงงานนี้ได้ศึกษาเกี่ยวกับความปลอดภัยในการใช้ทรัพยากรร่วมกัน และสร้างเครื่องมือสำหรับผู้ดูแลระบบ เพื่อใช้ในการตรวจสอบการใช้ทรัพยากรร่วมกัน โดยเครื่องมือที่สร้างขึ้นนี้เป็นตัวตรวจสอบว่ามีการแชร์ทรัพยากรใดบ้าง และตรวจสอบระดับการรักษาความปลอดภัยของการแชร์ทรัพยากรนั้น เพื่อเป็นแนวทางให้ผู้ดูแลระบบใช้ในการปรับแต่งระบบรักษาความปลอดภัยต่อไป

Resource Sharing Security Scanner

Mr. Yutthapong Manutsayanont
Mr. Suriya Jintakanont
Mr. Thana Hongsuwan Advisor
Mr. Akkradach Watcharapong Advisor

ABSTRACT

The resource sharing in computer network uses for data exchanging and economizing. When using resource sharing the security is important. If they are not secure, it may be somebody who can destroy the resources or do not have knowledge about using resource sharing that can make the loss in systems.

Windows networking and Unix networking is most to used. There are using resource sharing with them. They can share both directories and printers. In the network systems have many levels of security for resource sharing. Administrator can choose the best security and setup them to prevent damage.

This project is study about resource sharing security and development of tools for administrator uses for inspects the resource sharing. The tool is uses for enumerate the sharing resource and make report about security level. The results of tool that can use to be a guideline for administrators to diagnose security of systems and modify them.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดีเนื่องจากได้รับความช่วยเหลือ และคำแนะนำจากบุคคลหลายท่านด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือท่าน อาจารย์ ธนา หงษ์สุวรรณ และท่าน อาจารย์ อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ ให้คำชี้แนะและช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งให้ความดูแลเอาใจใส่เป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และคอยเป็นกำลังใจให้เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายยุทธพงษ์ มานุษยานนท์
นายสุริยา จินตกานนท์

สารบัญ

หน้าที่

สารบัญภาพประกอบ

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความสำคัญ และที่มา

ในสภาพแวดล้อมการทำงานในองค์กรนั้นต้องมีการใช้ทรัพยากรร่วมกันด้วยเหตุผลหลายประการ เช่น เพื่อการแลกเปลี่ยนข้อมูลหรือเพื่อความประหยัด และเมื่อมีการใช้งานทรัพยากรร่วมกันก็ควรมีการจัดสรรการใช้ทรัพยากรและการดูแลความปลอดภัย

การใช้งานเครื่องคอมพิวเตอร์ก็เช่นกัน เมื่อมีการใช้งานทรัพยากรร่วมกันก็จำเป็นที่ต้องมีระบบรักษาความปลอดภัยที่ดี โดยในปัจจุบันเครื่องคอมพิวเตอร์ส่วนบุคคลส่วนมากใช้ระบบปฏิบัติการของไมโครซอฟต์ เช่น วินโดวส์ 95/98 หรือวินโดวส์เอ็นที โดยในระบบปฏิบัติการเหล่านี้มีส่วนของการจัดการระบบเครือข่ายพื้นฐานและการใช้ทรัพยากรร่วมกันมาให้พร้อมใช้งาน เรียกว่าระบบเครือข่ายของวินโดวส์ (Microsoft Network) ซึ่งใช้โพรโตคอลเอสเอ็มบีเป็นองค์ประกอบหลักสำคัญ

นอกเหนือจากระบบเครือข่ายของวินโดวส์ซึ่งเป็นที่นิยมใช้งานกันแล้ว ยังมีระบบเครือข่ายของยูนิกซ์ซึ่งมีการใช้ทรัพยากรร่วมกันได้เหมือนกับระบบเครือข่ายของวินโดวส์ โดยในระบบเครือข่ายของยูนิกซ์ใช้โพรโตคอลเอ็นเอฟเอสเป็นหัวใจสำคัญ โดยทั้งเอสเอ็มบีและเอ็นเอฟเอสมีลักษณะการทำงานที่คล้ายคลึงกันและมีหน่วยงานไม่น้อยที่ใช้งานระบบเครือข่ายทั้งสองร่วมกัน

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.2.1 เพื่อศึกษาหลักการของการใช้ทรัพยากรร่วมกันในระบบเครือข่าย

1.2.2 ศึกษาเกี่ยวกับความปลอดภัยในการใช้ทรัพยากรร่วมกัน

1.2.3 ตรวจสอบความปลอดภัยในการใช้ทรัพยากรร่วมกัน

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้คือการสร้างเครื่องมือในการตรวจสอบการใช้ทรัพยากรร่วมกัน โดยเน้นที่ระบบเครือข่ายของวินโดวส์และระบบเครือข่ายของยูนิกซ์เป็นหลัก ซึ่งระบบเครือข่ายทั้งสองดังกล่าวนั้นใช้งานโพรโตคอลเอสเอ็มบีและเอ็นเอฟเอส โดยการทำงานในโครงงานนี้เป็นการศึกษากลไกการทำงานและเขียนโปรแกรมตรวจสอบข้อมูลจากการใช้ทรัพยากรร่วมกันของทั้งสองระบบเครือข่าย

1.4 เป้าหมายของปริญญานิพนธ์

เขียนโปรแกรมตรวจสอบการใช้ทรัพยากรร่วมกันของระบบเครือข่ายของวินโดวส์ซึ่งใช้งานโพรโตคอลเอสเอ็มบีและระบบเครือข่ายยูนิกซ์ซึ่งใช้งานเอ็นเอฟเอส โดยตัวโปรแกรมทำการตรวจสอบและรายงานผลเกี่ยวกับการเปิดให้ใช้ทรัพยากรร่วมกันในระบบเครือข่ายทั้งในระบบเครือข่ายของวินโดวส์และระบบเครือข่ายของยูนิกซ์ โดยรายงานรายละเอียดเกี่ยวกับการเปิดให้ใช้ทรัพยากรร่วมกัน

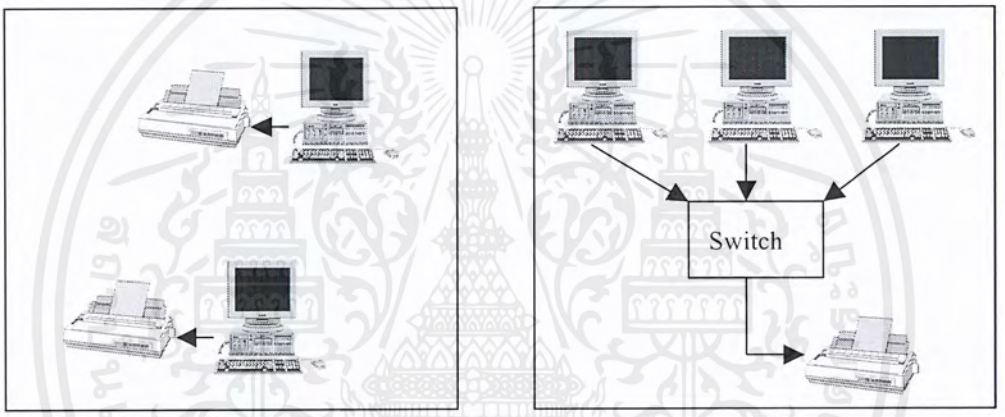
เช่น เปิดให้ใช้ทรัพยากรใดบ้าง, ทรัพยากรที่เปิดให้ใช้มีข้อมูลรายละเอียดปลีกย่อยอย่างไร, มีรหัสผ่านในการเข้าใช้หรือไม่และอื่นๆ

1.5 วิธีการดำเนินการ

การดำเนินงานในโครงงานนี้ เริ่มจากการศึกษาทฤษฎีพื้นฐานที่เกี่ยวข้องของโครงงาน ซึ่งได้แก่ การศึกษาเรื่องหลักๆ ดังนี้ คือ โพรโตคอลที่ใช้ในการใช้ทรัพยากรร่วมกันในระบบเครือข่ายของวินโดวส์ และในระบบเครือข่ายของยูนิกซ์, ศึกษาการเขียนโปรแกรมภาษาซีในระบบเครือข่ายและเขียนโปรแกรมตรวจสอบการใช้ทรัพยากรร่วมกันในระบบเครือข่าย

บทที่ 2
การใช้ทรัพยากรร่วมกัน

การทำงานในปัจจุบันนี้ ต้องการความสะดวกในการแลกเปลี่ยนข้อมูลข่าวสาร และมีการใช้ทรัพยากรร่วมกันเพื่อความประหยัด เช่น อุปกรณ์บางอย่างในองค์กรที่ไม่ได้ใช้งานเป็นประจำ ไม่จำเป็นต้องมีให้กับพนักงานทุกคน แต่มีเป็นของส่วนกลางเพียงชุดเดียว และให้พนักงานผลัดกันเข้ามาใช้อุปกรณ์นั้นทีละคนเพื่อความประหยัด คอมพิวเตอร์ก็เช่นกัน อุปกรณ์บางอย่างที่ไม่ได้ใช้งานตลอดเวลา เช่น เครื่องพิมพ์ ไม่จำเป็นต้องจัดให้มีเครื่องพิมพ์ที่เครื่องคอมพิวเตอร์ทุกเครื่อง เพราะไม่ได้เป็นอุปกรณ์ที่บุคคลใดบุคคลหนึ่งใช้งานตลอดเวลา แต่ทุกคนอาจมีความต้องการใช้งานได้ จึงควรจัดให้เป็นทรัพยากรที่ใช้งานร่วมกันได้ ดังรูปที่ 2.1



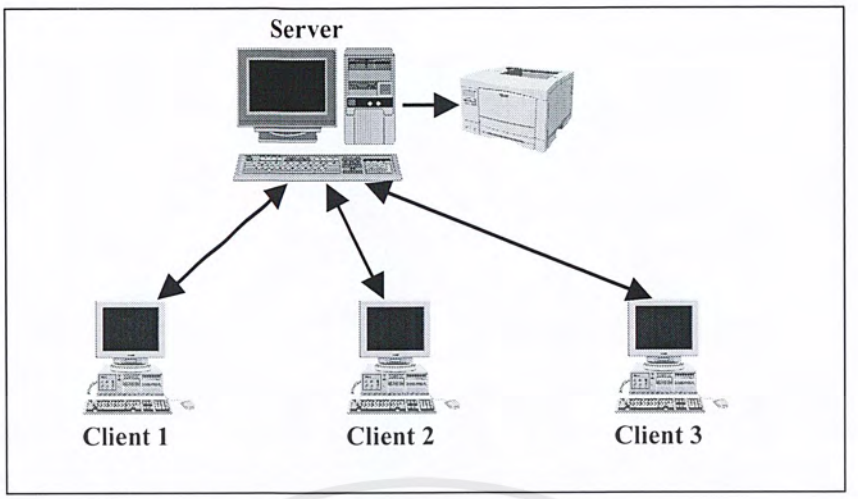
รูปที่ 2-1 (A) ไม่มีการใช้ทรัพยากรร่วมกัน (B) มีการใช้ทรัพยากรร่วมกัน

2.1 การเชื่อมต่อเครื่องคอมพิวเตอร์ในระบบเครือข่าย

เมื่อมีการพัฒนาระบบเครือข่ายสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล การใช้ทรัพยากรร่วมกันจึงทำได้ง่ายขึ้นโดยผ่านระบบเครือข่าย การเชื่อมต่อเครื่องคอมพิวเตอร์ในระบบเครือข่ายมี 2 ลักษณะ คือ

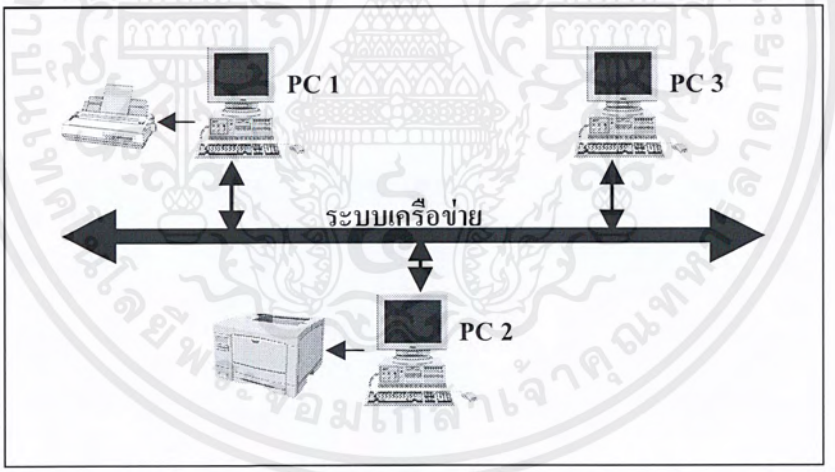
- แบบไคลเอนต์เซิร์ฟเวอร์ (Client–Server) คือ การเชื่อมต่อโดยมีเครื่องคอมพิวเตอร์เครื่องหนึ่ง (หรือหลายเครื่อง) ทำหน้าที่เป็นศูนย์กลาง (Server) ให้บริการและควบคุมดูแลเครื่องลูกข่าย (Client) การเชื่อมต่อในลักษณะนี้เครื่องลูกข่ายมีหน้าที่ขอใช้บริการที่ศูนย์กลาง ส่วนศูนย์กลางก็ทำหน้าที่ให้บริการเครื่องลูกข่าย ตัวอย่างเช่น ระบบเครือข่ายของโนเวลล์เน็ตแวร์ (Novell NetWare) หรือระบบคอมพิวเตอร์ที่ประมวลผลโดยศูนย์กลาง (Centralize Host)

จากรูปที่ 2-2 คือการเชื่อมต่อแบบไคลเอนต์เซิร์ฟเวอร์ เมื่อไคลเอนต์ต้องการขอใช้บริการก็ส่งคำร้องขอไปยังเซิร์ฟเวอร์ และเซิร์ฟเวอร์มีหน้าที่จัดลำดับเพื่อให้บริการตามคำขอ



รูปที่ 2-2 การเชื่อมต่อแบบไคลเอนต์เซิร์ฟเวอร์

- การเชื่อมต่อแบบทุกเครื่องมีความเสมอภาคกัน (Peer-to-peer) คือ การเชื่อมต่อที่ทุกเครื่องในระบบเครือข่ายมีความเสมอภาคกัน ไม่มีเครื่องใดเป็นศูนย์กลาง ทุกเครื่องเป็นได้ทั้งผู้ให้บริการและผู้ขอใช้บริการแล้วแต่ความต้องการของผู้ใช้งานในขณะนั้น ระบบที่ทำงานแบบนี้ เช่น ระบบเครือข่ายของวินโดวส์ (Windows Networking)



รูปที่ 2-3 การเชื่อมต่อแบบทุกเครื่องมีความเสมอภาคกัน

จากรูปที่ 2-3 ถ้ามีการแบ่งใช้ทรัพยากร ทุกเครื่องในระบบเครือข่าย สามารถเรียกใช้ทรัพยากรที่เปิดแชร์ไว้ได้ตามสิทธิ์ที่ตนได้รับ

2.2 ความปลอดภัยในการใช้ทรัพยากรร่วมกัน

เมื่อมีการใช้ทรัพยากรร่วมกัน ต้องมีการดูแลเรื่องความปลอดภัยของทรัพยากร เพราะบางครั้งอาจเกิดจากความไม่ตั้งใจ ความไม่รู้หรือความตั้งใจก็ตาม ที่ก่อให้เกิดความเสียหายแก่ทรัพยากรได้ เช่น

อาจมีผู้ใช้อื่นมาลบข้อมูลในเครื่องที่เปิดแชร์ไว้หรือมีการตั้งใจลักลอบเข้ามาขโมยข้อมูล จึงมีการตรวจสอบความปลอดภัยในลักษณะต่างๆ ขึ้นมา เช่น

- การแจ้งชื่อผู้ใช้และรหัสผ่านก่อนเข้าใช้ทรัพยากร เพื่อจะได้รู้ว่าเป็นใครเข้ามาใช้ทรัพยากร
- การกำหนดสิทธิ์ในการเข้าใช้ทรัพยากร เช่น ให้ใครเข้ามาใช้ทรัพยากรใดได้บ้าง
- การกำหนดระดับของสิทธิ์ในการเข้าใช้ทรัพยากร เช่น ให้เข้ามาอ่านได้อย่างเดียวแก้ไขไม่ได้
- การจัดลำดับก่อนหลังในการเข้าใช้ทรัพยากรเดียวกัน
- การตรวจสอบในกรณีที่มีผู้ใช้หลายคนเข้าใช้ทรัพยากรเดียวกันและมีการแก้ไขพร้อมกัน
- การกำหนดจำนวนผู้เข้าใช้ทรัพยากร ถ้ามีผู้เข้าใช้มากเกินไปอาจให้บริการไม่ทันหรือระบบล้มเหลว

สิ่งต่างๆ เหล่านี้ส่วนมากมีมาให้ในระบบปฏิบัติการที่ใช้ คงเหลือบางส่วนที่ผู้ใช้ต้องเป็นผู้ดูแลความปลอดภัยของทรัพยากรเอง โดยผู้ใช้มีหน้าที่ในการตรวจตราดูแลทรัพยากรของตนเอง และเป็นผู้กำหนดชื่อผู้ใช้และสิทธิ์ต่างๆ ของการเข้าใช้ทรัพยากรรวมถึงการตั้งรหัสผ่านของทรัพยากรนั้นด้วย โดยสิ่งที่อาจเป็นความไม่ปลอดภัยแก่ทรัพยากรมีดังนี้ เช่น

- ไม่มีการตรวจสอบผู้ที่เข้าใช้ทรัพยากร
- ผู้ใช้ไม่ทราบว่าได้เปิดแชร์ทรัพยากรใดไว้บ้าง
- มีการเปิดแชร์ทรัพยากรโดยไม่ได้ตั้งใจ
- ไม่มีการกำหนดระดับของสิทธิ์การเข้าใช้ เช่น ทรัพยากรใดควรอ่านได้อย่างเดียวหรือแก้ไขได้
- เครื่องที่เปิดแชร์ไม่มีรหัสผ่านในการเข้าใช้ทรัพยากร
- รหัสผ่านง่ายต่อการสุ่ม

การดูแลความปลอดภัยของการใช้ทรัพยากรร่วมกันในระบบเครือข่ายเป็นสิ่งที่ผู้เป็นเจ้าของทรัพยากรต้องหมั่นตรวจตรา เพราะถ้ามีจุดบกพร่องอาจเป็นช่องทางให้ผู้อื่นที่ไม่ประสงค์ดีหรือรู้เท่าไม่ถึงการ เข้ามาขโมยหรือทำลายทรัพยากรได้

2.3 การทำลายทรัพยากรซึ่งใช้งานร่วมกันในระบบเครือข่าย

การทำลายทรัพยากรซึ่งใช้งานร่วมกันในระบบเครือข่ายมักไม่ร้ายแรงมากเหมือนการโจมตีระบบเครือข่ายทั่วไป ในระบบการใช้ทรัพยากรร่วมกันนั้น การทำลายทรัพยากรมักเป็นการขโมยข้อมูลหรือการลักลอบเข้ามาทำลายข้อมูลมากกว่า ตัวอย่างการทำลายทรัพยากรมีดังนี้

2.3.1 การขโมยข้อมูล คือ การที่ผู้ใช้อื่นลักลอบเข้ามาอ่านข้อมูลที่เปิดแชร์ไว้ เพื่อนำข้อมูลนั้นไปใช้งานโดยไม่ได้รับอนุญาตจากผู้เป็นเจ้าของข้อมูล เนื่องจากในระบบเครือข่ายซึ่งมีการใช้ทรัพยากรร่วมกันบางระบบอาจไม่มีการตรวจสอบผู้เข้าใช้งาน เช่น ในระบบเครือข่ายของวินโดวส์ซึ่งใช้การแชร์

ทรัพยากรของวินโดวส์ 95/98 ถ้าไม่มีรหัสผ่านในการเข้าใช้ทรัพยากรแล้ว ผู้ใดในระบบเครือข่ายก็สามารถเข้าใช้งานทรัพยากรนั้นได้

การป้องกัน คือ จัดให้มีการรักษาความปลอดภัยของทรัพยากร โดยการตั้งรหัสผ่านในการเข้าใช้ทรัพยากร หรือมีการพิสูจน์ตนก่อนการเข้าใช้ทรัพยากรนั้น

2.3.2 การทำลายข้อมูล คือ การที่ผู้ใช้อื่นเข้ามาใช้ทรัพยากรแล้ว อาจด้วยความรู้เท่าไม่ถึงการณ์หรือด้วยความตั้งใจก็ตาม เมื่อเข้ามาดูข้อมูลซึ่งแชร์ไว้ในระบบเครือข่ายแล้วลบหรือทำการเปลี่ยนแปลงข้อมูลโดยไม่ได้รับอนุญาตจากเจ้าของข้อมูล สาเหตุอาจเกิดจากการที่ผู้เป็นเจ้าของข้อมูลมีการกำหนดสิทธิ์ในการเข้าใช้ข้อมูลไม่ดี เช่น ต้องการให้ผู้ใช้อ่านข้อมูลไปใช้แต่เปิดให้เข้าถึงข้อมูลแบบอ่านเขียนได้ เป็นเหตุให้ผู้ใช้อื่นสามารถแก้ไขข้อมูลหรือทำลายข้อมูลได้

การป้องกัน คือ มีการกำหนดสิทธิ์และระดับของสิทธิ์ในการเข้าใช้ทรัพยากร โดยกำหนดว่าข้อมูลใดควรให้เป็นชนิดอ่านได้อย่างเดียวหรืออ่านเขียนได้ ซึ่งโดยทั่วไปแล้วมักไม่เปิดให้ใช้ข้อมูลแบบอ่านเขียนได้ในการแชร์ทรัพยากร แต่มักเป็นการเปิดแชร์ข้อมูลแบบให้อ่านได้อย่างเดียว

2.3.3 การลักลอบใช้ทรัพยากร ในกรณีนี้ดูคล้ายกับการขโมยข้อมูล แต่อาจไม่ได้สูญเสียข้อมูลเพียงแต่ถูกผู้อื่นลักลอบเข้ามาใช้ทรัพยากรเท่านั้น เช่น การแชร์เครื่องพิมพ์ในระบบเครือข่าย โดยในระบบเครือข่ายของวินโดวส์ไม่สามารถกำหนดผู้ใช้งานและปริมาณการใช้งานเครื่องพิมพ์ที่เปิดแชร์ไว้ในระบบเครือข่ายได้ จึงเป็นสาเหตุให้ผู้อื่นลักลอบเข้าใช้ทรัพยากร หรือในอีกกรณีหนึ่งคือการเปิดแชร์ไดเรกทอรี่แบบอ่านเขียนได้ อาจมีผู้ใช้อื่นนำข้อมูลมาฝากหรือเรียกว่าเป็นการลักลอบใช้ทรัพยากรดิสก์

การป้องกัน คือ ไม่เปิดแชร์ไดเร็กทอรี่แบบอ่านเขียนได้ และในส่วนของเครื่องพิมพ์ควรจัดหาโปรแกรมตรวจสอบการเข้าใช้หรือจำกัดจำนวนการใช้งาน

2.3.4 การก่อกวนทรัพยากร คือ การสร้างความรำคาญให้แก่ผู้เป็นเจ้าของทรัพยากร โดยอาจเข้ามาใช้ทรัพยากรซึ่งเจ้าของต้องการเปิดให้แก่ผู้อื่น เพื่อเพิ่มปริมาณการจราจรในระบบเครือข่ายหรือเพิ่มภาระให้แก่เครื่องที่เปิดให้บริการแชร์ทรัพยากร ก่อให้เกิดความรำคาญแก่ผู้ใช้งาน

การป้องกัน คือ มีการกำหนดสิทธิ์และพิสูจน์สิทธิ์ก่อนการเข้าใช้ทรัพยากร

2.4 การกำหนดสิทธิ์ในการเข้าใช้ทรัพยากร

การกำหนดสิทธิ์ในการเข้าใช้ทรัพยากร คือ การกำหนดผู้ใช้งานและระดับของการเข้าถึงในการเข้าใช้ทรัพยากร เพื่อตรวจสอบ, ป้องกัน การก่อกวนและการทำลายทรัพยากรในลักษณะต่างๆ ซึ่งในระบบปฏิบัติการที่มีการใช้ทรัพยากรร่วมกันต้องมีระบบรักษาความปลอดภัยเหล่านี้มาให้ แต่ผู้ใช้ซึ่งเป็นเจ้าของและเป็นผู้ดูแลทรัพยากรต้องเป็นคนตั้งค่าเหล่านี้เอง โดยในระบบเครือข่ายของวินโดวส์มีการตั้งรหัสผ่านและการกำหนดระดับของสิทธิ์ในการเข้าใช้มาให้ ร่วมถึงการกำหนดชื่อผู้ใช้งานในการเข้าใช้ทรัพยากรด้วย ส่วนในระบบเครือข่ายยูนิกซ์ ยังมีการรักษาความปลอดภัยการเข้าใช้ทรัพยากรที่หละหลวม

กว่า โดยตรวจสอบเพียงหมายเลขไอพีของเครื่องที่เข้าใช้งาน ยังไม่มีการตรวจสอบผู้ใช้งาน ส่วนการกำหนดระดับของสิทธิ์ในการเข้าใช้สามารถกำหนดได้เช่นเดียวกับในระบบเครือข่ายของวินโดวส์

2.5 การเจาะระบบรักษาความปลอดภัยในการเข้าใช้ทรัพยากร

ถึงแม้ว่าในระบบปฏิบัติการที่มีการใช้ทรัพยากรร่วมกันจะมีระบบรักษาความปลอดภัยต่างๆ ให้ผู้ดูแลระบบเลือกใช้ แต่ก็มีผู้ไม่ประสงค์ดีต้องการเจาะระบบรักษาความปลอดภัยเหล่านี้เพื่อก่อกวนหรือขโมยทรัพยากร โดยมีการเจาะระบบรักษาความปลอดภัยในลักษณะต่างๆ มีดังนี้

2.5.1 การขโมยรหัสผ่าน คือ การได้มาซึ่งรหัสผ่านของการเข้าใช้ทรัพยากรโดยมิได้รับอนุญาตจากผู้เป็นเจ้าของทรัพยากร การได้มาซึ่งรหัสผ่านนี้ทำได้หลายวิธี เช่น

- ลักลอบจดจำมาจากผู้ใช้ซึ่งได้รับรหัสผ่านมาจากเจ้าของทรัพยากรในขณะกำลังป้อนข้อมูล
- ใช้โปรแกรมตรวจจับข้อมูลในระบบเครือข่าย เช่น โปรแกรมจำพวก Sniffer ตรวจจับในขณะที่ผู้ใช้ล็อกอินเพื่อเข้าใช้ทรัพยากร การป้องกันทำได้โดยเลือกใช้การเข้ารหัส (Encryption) ในการส่งรหัสผ่านในระบบเครือข่าย โดยในระบบปฏิบัติการวินโดวส์มีให้เลือกใช้การเข้ารหัสนี้ได้
- การทดลองสุ่มรหัสผ่านเพื่อลองติดต่อ อาจทำโดยการเขียนโปรแกรมหรือป้อนรหัสผ่านด้วยมือ การป้องกันทำได้โดยการตั้งค่าตัวนับในการป้อนรหัสผ่าน โดยกำหนดจำนวนครั้งในการป้อนรหัสผ่านว่าสามารถป้อนรหัสผ่านผิดได้กี่ครั้งจึงให้ระบบป้องกันทำงาน
- ใช้โปรแกรมในการถอดรหัสผ่านของการแชร์ทรัพยากรโดยตรง โปรแกรมเหล่านี้หาได้จากในเว็บของการเจาะระบบทั่วไป การป้องกันต้องใช้ระบบรักษาความปลอดภัยที่มีการป้องกันสูง โดยอาจใช้การพิสูจน์สิทธิ์โดยผ่านโดเมนซึ่งมีการรักษาความปลอดภัยที่ดี

2.5.2 การปลอมไอพี คือ การเปลี่ยนแปลงหมายเลขไอพีซึ่งส่งไปกับแพ็กเกจของทีซีพีไอพี เพื่อใช้ในการพิสูจน์ในการเข้าใช้ทรัพยากรของระบบเครือข่ายยูนิกซ์ เพราะการแชร์ทรัพยากรในระบบเครือข่ายยูนิกซ์มีการตรวจสอบเพียงหมายเลขไอพีของเครื่องที่ต้องการเข้าใช้งานเท่านั้น ไม่มีการตรวจสอบชื่อผู้ใช้งานได้ (สิทธิ์ในการเข้าใช้ทรัพยากรที่แชร์ในระบบเครือข่ายยูนิกซ์ขึ้นอยู่กับหมายเลขไอพี) ดังนั้นเมื่อผู้ใช้ใดที่ใช้เครื่องซึ่งมีหมายเลขไอพีตรงตามที่ตั้งค่าไว้ก็สามารถเข้าใช้ทรัพยากรที่แชร์ไว้ได้โดยไม่ต้องดูว่าเป็นผู้ใช้คนใด

2.6 การตรวจจับและป้องกันผู้ลักลอบเข้าใช้ทรัพยากร

การตรวจตราดูแลทรัพยากรที่เปิดให้ใช้ร่วมกันได้ในระบบเครือข่ายเป็นหน้าที่ของผู้เป็นเจ้าของทรัพยากรที่ควรกระทำ เพราะถึงแม้ว่าอาจมีการตั้งค่าการรักษาความปลอดภัยที่ดีแล้ว ยังมีบุคคลซึ่งอาจเข้าใช้ทรัพยากรแบบถูกต้องแต่ไม่ได้รับอนุญาตจากเจ้าของทรัพยากร โดยสิทธิ์ที่ได้มานั้นอาจได้มาจากการเจาะระบบรักษาความปลอดภัย ผู้ดูแลทรัพยากรจึงควรหมั่นตรวจตราการเข้าใช้ทรัพยากรนั้น การตรวจตราการเข้าใช้ทรัพยากรทำได้โดย

- ใช้โปรแกรมเฝ้าดูการเข้าใช้ทรัพยากร (Monitor) เช่นโปรแกรม “Net Watcher” ซึ่งมีมาให้ในระบบปฏิบัติการวินโดวส์ 95/98 เพื่อเฝ้าดูการเข้าใช้ทรัพยากรในเครื่อง

- ตรวจสอบล็อกไฟล์ เพื่อดูโปรเซสการทำงานที่น่าสงสัยว่าเป็นการเข้ามาใช้งานทรัพยากร

- ไม่เปิดแชร์ทรัพยากรที่ไม่จำเป็นและยกเลิกการแชร์เมื่อสิ้นสุดการใช้งาน

2.7 ความเสียหายอื่นๆ ในการใช้ทรัพยากรร่วมกัน

นอกเหนือจากความเสียหายซึ่งเกิดขึ้นแก่ตัวทรัพยากรซึ่งแชร์ไว้ในระบบเครือข่ายแล้ว ยังอาจมีความสูญเสียซึ่งเกิดกับผู้ใช้งานทรัพยากรได้ เช่น

- ถูกตัดการติดต่อก่อนใช้งานเสร็จ คือ ในกรณีที่ผู้ใช้กำลังเข้าใช้ทรัพยากรอยู่และยังใช้งานไม่เสร็จ แต่ผู้ใช้บริการตัดการติดต่อไปก่อนด้วยสาเหตุใดๆ เช่นไฟดับ, ตั้งใจปิดเครื่อง, เครื่องที่ให้บริการอยู่แฮงค์หรือระบบเครือข่ายล้มเหลว ทำให้ผู้ใช้ได้ข้อมูลไม่ครบและเครื่องของผู้ใช้ทรัพยากรนั้นแฮงค์ตามไปด้วย แก้ไขโดยตรวจสอบให้แน่ชัดก่อนการปิดเครื่องว่ายังมีผู้ใช้ใดเชื่อมต่อเพื่อใช้ทรัพยากรอยู่หรือไม่ และ ดูแลระบบเครือข่ายให้ทำงนได้เป็นปกติเสมอ

- การจราจรที่หนาแน่นบนระบบเครือข่าย เกิดจากการโอนถ่ายข้อมูลปริมาณมากผ่านระบบเครือข่ายโดยการแชร์ ซึ่งอาจเป็นการรบกวนผู้อื่นซึ่งใช้งานระบบเครือข่ายเช่นกัน จึงไม่ควรโอนถ่ายข้อมูลปริมาณมากโดยการแชร์

บทที่ 3
การใช้ทรัพยากรร่วมกันบนระบบเครือข่ายของวินโดวส์

3.1 ระบบเครือข่ายของวินโดวส์ (Microsoft Networking)

ระบบเครือข่ายของวินโดวส์ หมายถึง ระบบเครือข่ายที่ใช้ระบบปฏิบัติการของไมโครซอฟท์เป็นหลัก โดยอาจเริ่มนับได้ตั้งแต่แลนแมนเนจเจอร์ (LAN Manager) ซึ่งใช้งานร่วมกับระบบดอส จนกระทั่งเป็นวินโดวส์ฟอร์เวิร์กกรุป 3.1 (Windows for Workgroup 3.1) เป็นต้นมา โดยระบบเครือข่ายของวินโดวส์นี้มีการทำงานแบ่งตามลักษณะการเชื่อมต่อในระบบเครือข่ายได้ 2 แบบคือ

- การเชื่อมต่อแบบเวิร์กกรุป มีลักษณะการเชื่อมต่อแบบทุกเครื่องมีความเสมอภาคกัน โดยเครื่องที่เชื่อมต่อเป็นแบบเวิร์กกรุปนี้ แต่ละเครื่องมีการกำหนดสิทธิ์และพิสูจน์สิทธิ์ในการเข้าใช้ทรัพยากรได้เอง ระบบปฏิบัติการที่ทำงานแบบนี้ คือ วินโดวส์ฟอร์เวิร์กกรุป, วินโดวส์ 95/98, วินโดวส์เอ็นที

- การเชื่อมต่อแบบโดเมน มีลักษณะการเชื่อมต่อแบบไคลเอนต์เซิร์ฟเวอร์ โดยมีตัวควบคุมอยู่ที่ศูนย์กลางเรียกว่า ตัวควบคุมโดเมน (Domain Controller) โดยตัวควบคุมโดเมนนี้ทำหน้าที่ในการควบคุมและพิสูจน์สิทธิ์ต่างๆ ในระบบเครือข่าย โดยเครื่องในระบบเครือข่ายที่ทำงานแบบโดเมนนี้ มีการกำหนดและพิสูจน์สิทธิ์ที่ตัวควบคุมโดเมน ระบบปฏิบัติการที่ทำงานแบบนี้ คือ วินโดวส์เอ็นทีและวินโดวส์ 2000 โดยมีวินโดวส์เอ็นทีเซิร์ฟเวอร์ทำหน้าที่เป็นตัวควบคุมโดเมน และวินโดวส์เอ็นทีเวิร์กสเตชันเป็นลูกข่ายในโดเมน ส่วนระบบปฏิบัติการอื่นๆ สามารถเข้าเป็นส่วนหนึ่งของโดเมนได้แต่ไม่สามารถกำหนดสิทธิ์และพิสูจน์สิทธิ์ผ่านโดเมนได้ คงทำได้เพียงเรียกใช้บริการบางอย่างของโดเมนเท่านั้น

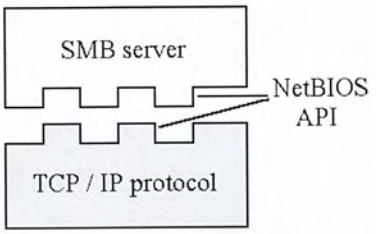
จุดเด่นของระบบเครือข่ายของวินโดวส์ คือ การใช้ทรัพยากรร่วมกัน โดยสามารถใช้งานไฟล์และเครื่องพิมพ์ร่วมกันในระบบเครือข่ายได้อย่างมีประสิทธิภาพ หัวใจของการใช้ทรัพยากรร่วมกันในระบบเครือข่ายของวินโดวส์นี้ คือ โพรโตคอลเอสเอ็มบี (Server Message Block: SMB)

3.2 ประวัติของโพรโตคอลเอสเอ็มบี (SMB)

โพรโตคอลเอสเอ็มบี (SMB: Server Message Block) มีการพัฒนาขึ้นเพื่อใช้งานตั้งแต่ปีคริสตศักราช 1980 แต่มาเป็นที่นิยมใช้งานอย่างแพร่หลายเมื่อไมโครซอฟท์นำมาใส่ลงในผลิตภัณฑ์ของตน เริ่มตั้งแต่ LAN Manager ที่ใช้งานร่วมกับระบบดอสมาจนถึงในระบบเครือข่ายของวินโดวส์ โดยเริ่มพบในวินโดวส์ฟอร์เวิร์กกรุป 3.1 และต่อมาได้ใช้เป็นมาตรฐานในวินโดวส์ 95/98 รวมไปถึงวินโดวส์เอ็นทีหรือวินโดวส์ 2000

เอสเอ็มบีเคยได้รับการพัฒนาโดยหลายบริษัทชั้นนำทางคอมพิวเตอร์ เช่น ซีร็อกซ์ (Xerox), ทรีคอม (3Com) และไมโครซอฟท์ โดยมีหน้าที่เป็นโพรโตคอลหลักสำหรับการใช้งานไฟล์และเครื่องพิมพ์ร่วมกัน (File & printer sharing) มีการทำงานโดยการเรียกใช้หรือติดต่อกับโพรโตคอลในชั้นถัดลงไปโดยผ่านทางเอพีไอ (Application Programming Interface: API) ตามแบบที่เรียกว่าเน็ตไบออส

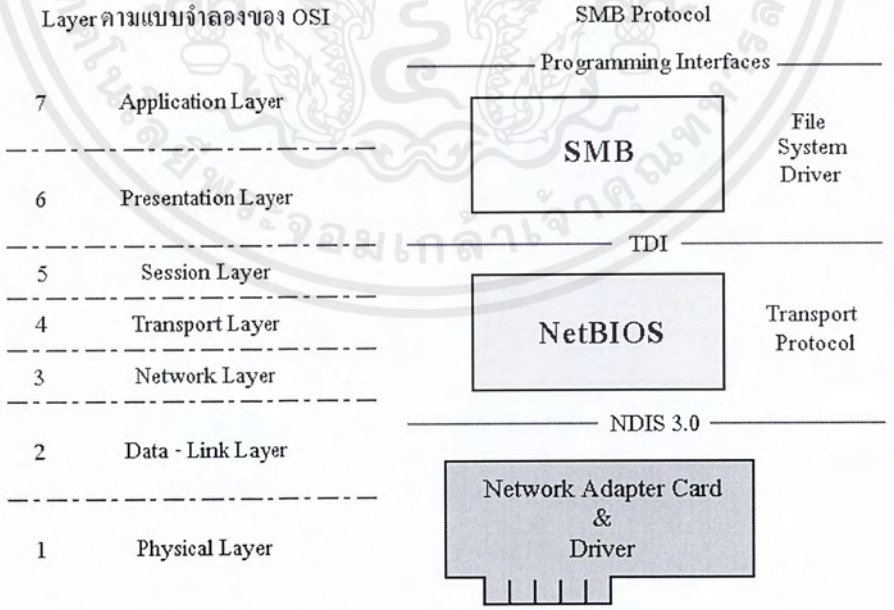
(NetBIOS) แต่เดิมเน็ตไบออสนั้นทำงานบน NetBEUI เป็นหลัก ต่อมาได้มีการพัฒนาให้ทำงานกับโพรโตคอลอื่นๆ คือ ไอพีเอ็กซ์ (IPX) ของโนเวลล์ (Novell) และทีซีพีไอพีได้ด้วย



รูปที่ 3-1 โครงสร้างโพรโตคอล SMB กับ NetBIOS API

3.3 โครงสร้างและการทำงานพื้นฐาน

กระบวนการติดต่อสื่อสารระหว่างอุปกรณ์สื่อสารหรือคอมพิวเตอร์นั้น แบ่งตามหน้าที่ย่อยได้เป็น 7 ระดับตามมาตรฐานไอเอสโอ (ISO: International Organization for Standardization) ในส่วนของโพรโตคอลเอสเอ็มบีนั้นเมื่อเปรียบเทียบกับแบบจำลองโอเอสไอ (OSI: Open System Interface) ดังรูปที่ 3-2 เห็นได้ว่าเอสเอ็มบีนั้นทำงานอยู่ในระดับที่เป็นแอปพลิเคชั่นและพรีเซนเตชันเท่านั้น เพราะเอสเอ็มบีจัดเป็นแอปพลิเคชั่นโปรแกรมอินเทอร์เฟซ การทำงานในลำดับชั้นที่ต่ำลงมาจึงเรียกใช้บริการของเน็ตไบออสต่อไป



รูปที่ 3-2 โพรโตคอลเอสเอ็มบีเทียบกันแบบจำลองโอเอสไอ

โครงสร้างของโพรโตคอลเอสเอ็มบี จากชื่อของโพรโตคอลเอสเอ็มบีนั้นมาจากโครงสร้างภายในโดยการจัดคำสั่งต่างๆ ของโพรโตคอลลงในแพ็กเกจ (Package) ที่เรียกว่า “Server Message Block” และเรียกใช้โพรโตคอลอื่นๆ ในชั้นถัดลงไปโดยผ่านทางเอพีไอของเน็ตไบออส โดยภายในแต่ละบล็อกของข้อมูลประกอบด้วยส่วนหัว (Header) ซึ่งมีขนาดคงที่ กับส่วนพารามิเตอร์และข้อมูลซึ่งมีขนาดไม่แน่นอน แพ็กเกจของโพรโตคอลเอสเอ็มบีแบ่งได้เป็น 4 กลุ่มใหญ่ๆ คือ Session Control ใช้เมื่อเริ่มการติดต่อโดยทำการล๊อกอิน พิสูจน์ตัวผู้ใช้ (Authenticate) และจบการติดต่อ, File ใช้งานเกี่ยวกับไฟล์ข้อมูลต่างๆ, Printer เรียกใช้เครื่องพิมพ์และสั่งงานพิมพ์ และ Message ใช้ในการติดต่อประสานงานกับเครื่องอื่น เช่น การส่งข้อความที่เรียกว่าวินป๊อปอัพ (WinPopup)

หน้าที่ของโพรโตคอลเอสเอ็มบีโดยรวมแล้วเทียบได้กับโพรโตคอลเอ็นเอฟเอส (NFS: Network File System) ซึ่งเป็นระบบการแบ่งใช้ไฟล์ร่วมกันบนเครื่องที่ใช้ระบบปฏิบัติการยูนิกซ์และคำสั่ง “lpd” ที่เรียกใช้เครื่องพิมพ์ของยูนิกซ์รวมกัน แต่เอสเอ็มบีมีความซับซ้อนมากกว่าเพราะมีการจำสถานะปัจจุบันของการติดต่อหรือการอ่านเขียนข้อมูลซึ่งในเอ็นเอฟเอสไม่มีการทำงานในส่วนนี้ โดยในเอ็นเอฟเอสเปิดโอกาสให้เครื่องอื่นอ่านข้อมูลไปใช้หรือแก้ไขตรงๆ เท่านั้น ไม่สามารถจัดการให้เซิร์ฟเวอร์ล็อกข้อมูลเอาไว้หรือจดจำว่ามีใครกำลังใช้งานอยู่บ้างเพื่อป้องกันในกรณีที่ผู้ใช้หลายคนต้องการแก้ไขข้อมูลชุดเดียวกัน โดยจัดการให้แต่ละไฟล์หรือแต่ละพื้นที่มีผู้ใช้เข้ามาใช้งานได้ทีละคนเท่านั้น เพื่อป้องกันการแก้ไขข้อมูลซ้ำซ้อน ส่วนเซิร์ฟเวอร์ที่ใช้เอสเอ็มบีนั้นมีระบบการล็อกไฟล์และเรคอร์ดหรือบันทึกสถานะการทำงานต่างๆ ซึ่งจดจำว่าใครกำลังแก้ไขหรือใช้ข้อมูลส่วนใดอยู่ เพื่อไม่ให้มีการแก้ไขซ้ำซ้อนกัน รวมทั้งการทำงานแบบ opportunistic lock หรือ op-lock ที่ยอมให้เครื่องไคลเอนต์อ่านข้อมูลไปพัก (Cache) เพื่อใช้งาน หากมีผู้ใช้อื่นมาแก้ไขข้อมูลต้นฉบับก็มีกลไกตามมาแก้ไขเป็นทอดๆไป เป็นการช่วยเพิ่มความเร็วในการทำงานให้สูงขึ้น

โพรโตคอลเอสเอ็มบีมีคำสั่งในระดับบนสุดหรือคำสั่งหลักกว่า 60 คำสั่ง เช่น SMB Read, SMB Write และบางคำสั่งหลักยังมีคำสั่งแยกย่อยออกไปอีกมาก และเอสเอ็มบียังมีความสามารถในการขยายโพรโตคอลออกไปโดยเพิ่มคำสั่งพิเศษต่างๆ เข้ามาใหม่ได้

การรับส่งแพ็กเกจของเอสเอ็มบีมี 3 ลักษณะ คือ

- UDP/137 คือ แพ็กเกจแบบยูดีพี (UDP: User Datagram Protocol) รับส่งผ่านพอร์ต 137 ใช้สำหรับการแจ้งชื่อเครื่องและการสอบถามชื่อเครื่องกับเซิร์ฟเวอร์ที่ทำหน้าที่รวบรวมรายชื่อ (Windows Internet Name Service Server: WINS Server)

- UDP/138 คือ แพ็กเกจแบบยูดีพี รับส่งผ่านพอร์ต 138 ใช้สำหรับการเรียกดูทรัพยากรต่างๆ ในระบบเครือข่าย (Browsing) เป็นการส่งแบบกระจาย (Broadcast) ภายในซับเน็ตเดียวกันหรือเรียกดูทรัพยากรข้ามซับเน็ต

- TCP/139 คือ แพ็กเกจแบบทีซีพี ซึ่งมีการตรวจสอบและตอบรับที่รัดกุมดีกว่ายูดีพีทำให้ข้อมูลที่รับส่งมีความเชื่อถือได้สูงกว่า รับส่งผ่านพอร์ต 139 ใช้สำหรับส่งข้อมูลหลักของโพรโตคอลเอสเอ็มบี เช่น การอ่านเขียนไฟล์และการส่งงานไปพิมพ์บนเครื่องพิมพ์ที่แชร์ไว้

3.4 เน็ตไบออส (NetBIOS)

การทำงานของเอสเอ็มบี เป็นการเรียกใช้งานหรือติดต่อกับโพรโตคอลในชั้นถัดลงไปโดยผ่านทางเอพีไอซึ่งเรียกว่าเน็ตไบออส โดยเน็ตไบออสไม่ได้เป็นโพรโตคอลแต่เป็นข้อกำหนดของวิธีการติดต่อเพื่อเรียกใช้โพรโตคอลในระดับของทรานสปอร์ตเลเยอร์จากโปรแกรมหรือโพรโตคอลที่อยู่ในระดับชั้นสูงกว่า ส่วนตัวโปรแกรมหรือโพรโตคอลที่ถูกเรียกใช้งานจริงผ่านการติดต่อแบบเน็ตไบออสนั้น อาจเป็นโพรโตคอลทีซีพีไอพี, ไอพีเอ็กซ์ หรือ NetBEUI ก็ได้ โดยในปัจจุบันยังไม่มีการกำหนดมาตรฐานของเน็ตไบออสอย่างเป็นทางการ เพียงแต่อ้างอิงกันโดยถือตามแบบอย่างที่ใช้กับเครื่องคอมพิวเตอร์ส่วนบุคคลของไอบีเอ็ม (IBM PC) ซึ่งเริ่มใช้งานกันในปี 1984

3.4.1 ชนิดของเน็ตไบออส ตัวโพรโตคอลจริงๆ นั้น มีให้เรียกใช้กันอยู่ 3 แบบ คือ NetBEUI, เน็ตไบออสบนไอพีเอ็กซ์ และเน็ตไบออสบนทีซีพีไอพี แต่ละชนิดมีรูปแบบดังนี้

- NetBEUI ย่อมาจาก NetBIOS Extended User Interface หรือเรียกกันอีกอย่างว่าเป็น RAW NetBIOS frame คือ การรับส่งข้อมูลที่เรียกใช้จากเน็ตไบออสเอพีไอโดยตรง นับเป็นโพรโตคอลแบบแรกที่ใช้งานบนเซิร์ฟเวอร์ซึ่งรันโปรแกรมแลนแมนเนจเจอร์ของไอบีเอ็มตั้งแต่ปี 1984 และได้รับการพัฒนาต่อโดยไมโครซอฟท์ ข้อเด่นของเน็ตไบออสแบบนี้ คือ มีกระบวนการที่ง่ายไม่ซับซ้อนทำให้ โอเวอร์เฮด (Overhead) ต่ำและทำงานได้เร็วเหมาะสำหรับระบบเครือข่ายขนาดเล็ก แต่มีข้อเสีย คือ ไม่สามารถจัดหาเส้นทางใหม่ (Non-routable) และไม่สามารถทำการเร้าท์ (Routing) หรือจัดเส้นทางรับส่งข้อมูลต่อไปยังระบบเครือข่ายอื่นๆ ได้ จึงใช้งานได้เฉพาะภายในส่วนหรือเซ็กเมนต์ (Segment) เดียวของระบบเครือข่ายที่เชื่อมกันอยู่โดยตรงเท่านั้น

- NetBIOS บน IPX เกิดขึ้นในช่วงต้นทศวรรษ 1990 โดยทางโนเวลล์ (Novell) ได้พัฒนาการติดต่อระบบเครือข่ายของวินโดวส์ให้ทำงานร่วมกับโพรโตคอลไอพีเอ็กซ์เอสพีเอ็กซ์ (IPX / SPX) ที่ใช้ในการติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์ในระบบเครือข่ายได้ จึงได้พัฒนาโพรโตคอลไอพีเอ็กซ์ในแบบที่ใช้การเชื่อมต่อของเน็ตไบออสขึ้นมาโดยมีลักษณะที่สำคัญ คือ ทำงานตามเอพีไอของเน็ตไบออสแต่ใช้ไอพีเอกซ์เป็นโพรโตคอลในระดับทรานสปอร์ตแทน และโพรโตคอลไอพีเอ็กซ์มีความสามารถในการจัดหาเส้นทางใหม่ได้ (Routable) จึงใช้งานในระบบเครือข่ายขนาดใหญ่ที่ประกอบด้วยเซ็กเมนต์ต่างๆ กันเป็นจำนวนมากได้ และต่อมาทางไมโครซอฟท์ได้รวมเอา NetBIOS บน IPX เข้าไว้เป็นส่วนหนึ่งของโปรแกรมวินโดวส์เพื่อให้ใช้งานร่วมกับระบบเครือข่ายของวินโดวส์ได้

- NetBIOS บน TCP / IP หรือเรียกสั้นๆ ว่าเน็ตบีที (NetBT) มีลักษณะการทำงานเหมือนเน็ตไบออสตัวอื่นและใช้การเชื่อมต่อตามแบบของเน็ตไบออสแต่ใช้โพรโตคอลทีซีพี (Transmission Control - Protocol: TCP) และยูดีพี (User Datagram Protocol: UDP) แทน เน็ตบีทีมีข้อดี คือ เนื่องจากทีซีพีไอพีมีความสามารถในการจัดหาเส้นทางใหม่ได้ และเป็นโพรโตคอลหลักที่ใช้งานบนอินเตอร์เน็ตจึงทำให้เน็ตบีทีสามารถทำงานผ่านอินเตอร์เน็ตหรือระบบเครือข่ายระยะไกลได้ และทีซีพีไอพี เป็นโพรโตคอลที่ได้รับความนิยมใช้งานกันอย่างแพร่หลายในระบบปฏิบัติการต่างๆ ทุกชนิด จึงสามารถนำเน็ตบีทีมาประยุกต์ในการสร้างระบบที่ให้บริการตามแบบของเน็ตไบออสข้ามเครื่องและข้ามระบบปฏิบัติการได้ไม่ยุ่งยาก

3.4.2 บริการต่างๆ ของเน็ตไบออส

บริการต่างๆ ของเน็ตไบออส ใช้ในการติดต่อแสดงตัวและจัดการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์รวมถึงการเข้าใช้ทรัพยากรบนเครื่องเหล่านั้น แบ่งเป็น 3 กลุ่มบริการ คือ

- Name Service คือ กลุ่มคำสั่งที่เกี่ยวกับการอ้างอิงชื่อเครื่องและทรัพยากรต่างๆ โดยการอ้างอิงใช้ชื่อตามแบบของเน็ตไบออสเป็นหลัก คือ ชื่อที่ประกอบด้วยตัวอักษรยาวไม่เกิน 16 ตัว และไม่ขึ้นต้นด้วย "*" เมื่อเรียกชื่อแล้วจึงนำชื่อไปแปลงเป็นข้อมูลจริงอีกครั้ง เช่น ในกรณีของเน็ตบีทีมีการแปลงชื่อเป็นหมายเลขไอพี (IP address) ชื่อตามแบบของเน็ตไบออสแบ่งเป็น 2 กลุ่ม ได้แก่ exclusive name คือ ชื่อที่กำหนดให้ใช้เฉพาะในแอปพลิเคชั่นตัวใดตัวหนึ่งเท่านั้น (unique name) และ group name คือชื่อที่ใช้ร่วมกันระหว่างกลุ่มของหลายๆ แอปพลิเคชั่น

- Session Service คำว่าเซสชั่นของเน็ตไบออสหมายถึงการที่สองแอปพลิเคชั่นมีการติดต่อกันผ่านเน็ตไบออส อาจเทียบได้กับวงจรเสมือน (Virtual circuit) ของทีซีพีไอพี โดยแต่ละฝ่ายสามารถเรียกใช้ชื่อต่างๆ ที่เป็นของอีกฝ่ายหนึ่งได้ และระหว่างสองแอปพลิเคชั่นอาจมีเซสชั่นเชื่อมต่อกันมากกว่าหนึ่งเซสชั่นก็ได้ คำสั่งพื้นฐานที่เกี่ยวกับเซสชั่น เช่น Call คือ การเรียกใช้บริการของเน็ตไบออสตามชื่อที่ระบุ, Listen คือ การตอบรับการเรียกใช้บริการจากฝ่ายที่เรียกเข้ามา, Send คือ การส่งข้อมูลผ่านทางเซสชั่น และ Receive คือ การรับข้อมูลผ่านทางเซสชั่น

- Datagram service คือ กลุ่มคำสั่งที่ใช้ในการรับส่งข้อมูลระหว่างแอปพลิเคชั่นที่ติดต่อกันผ่านเน็ตไบออส แต่การรับส่งแบบนี้ไม่มีการตรวจสอบที่รัดกุมทุกขั้นตอนเหมือนกับบริการของเซสชั่น อาจเทียบได้กับยูดีพีของโพรโตคอลทีซีพีไอพีนั่นเอง โดยการส่งข้อมูลที่เป็นลักษณะของดาต้าแกรมในรูปแบบของเน็ตไบออสนี้ จะส่งให้กับผู้รับที่มีการแจ้งให้ทราบหรือลงทะเบียนไว้กับฝ่ายที่เป็นผู้ส่งเท่านั้น หากเป็นการส่งไปยังชื่อกลุ่ม (Group name) ในแบบของเน็ตไบออสนั้น อาศัยการส่งแบบกระจายไปทั่ว (Broadcast)

3.5 การแชร์และการล็อกข้อมูล

การล็อกข้อมูลเป็นกลไกที่สำคัญในระบบที่มีการใช้ทรัพยากรข้อมูลร่วมกัน การล็อกเมื่อโปรแกรมหรือเครื่องใดเครื่องหนึ่งกำลังแก้ไขข้อมูลนั้นๆ เพื่อเป็นการบอกให้เครื่องอื่นรับทราบและได้รับข้อมูลใหม่ล่าสุดไปก่อนที่จะใช้หรือลงมือแก้ไขบ้าง กลไกการล็อกในเอสเอ็มบีแบ่งออกได้เป็น 3 ลักษณะ คือ

- Byte range locking คือ การล็อกข้อมูลเฉพาะบางช่วงในไฟล์ โดยระบุว่าเป็นช่วงใดในไฟล์ โดยกำหนดขอบเขตเป็นไบต์ เช่น การล็อกเฉพาะบางเรคคอร์ดในฐานข้อมูล เพื่อให้ผู้ใช้อื่นยังคงใช้งานส่วนที่ไม่ถูกล็อกของไฟล์นั้นได้ ส่วนที่ล็อกไว้ผู้ใช้อื่นต้องรอให้มีการแก้ไขเสร็จและปลดล็อกก่อน จึงเรียกข้อมูลล่าสุดที่แก้ไขแล้วไปใช้งานได้

- Share mode คือ การกำหนดลักษณะการแชร์ข้อมูลให้กับทั้งไฟล์ โดยเป็นการกำหนดว่าผู้ใช้รายอื่นสามารถทำอะไรกับไฟล์นี้ได้บ้างในขณะที่ใช้ไฟล์ร่วมกัน เช่น กำหนดว่าในขณะที่มีผู้ใช้รายหนึ่งกำลังเปิดใช้ไฟล์เพื่อแก้ไขข้อมูลผู้ใช้รายอื่นมีสิทธิ์อ่านได้อย่างเดียวเท่านั้นแก้ไขข้อมูลในไฟล์พร้อมกันไม่ได้

- Opportunistic lock หรือเรียกสั้นๆ ว่าอ็อปล็อก (Op-lock) เป็นกลไกที่ช่วยเพิ่มเติมความเร็วในการทำงานกับไฟล์ของโพรโตคอลเอสเอ็มบีได้มาก โดยการยอมให้ผู้ที่กำลังใช้งานไฟล์นั้นดึงข้อมูลที่ต้องการแก้ไขไปพัก (Cache) ไว้บนเครื่องที่เป็นไคลเอนต์ของตนเพื่อแก้ไขได้ ช่วยให้ทำงานได้รวดเร็วกว่าการที่อ่านเขียนผ่านระบบเครือข่ายทุกครั้ง

3.6 ส่วนขยายของโพรโตคอลเอสเอ็มบี

เนื่องจากเอสเอ็มบีมีการแก้ไขมาหลายรุ่นตั้งแต่ช่วงแรกที่ออกมาพร้อมกับระบบปฏิบัติการรุ่นต่างๆ ของไมโครซอฟท์ ได้มีการปรับปรุงแก้ไขข้อผิดพลาดอยู่ตลอดเวลา และยังมีการปรับปรุงในด้านความเร็วในการทำงาน จึงเกิดเป็นมาตรฐานสำหรับส่วนขยายรุ่นต่างๆ ของเอสเอ็มบีขึ้น 6 รุ่นคือ PC-NET 1.0 (Core), Microsoft Network Program 1.03 (Core Plus), LAN Manager 1.0, LAN Manager 2.0 (LM 2.0), NT LAN Manager 0.12 (NTLM 0.12) และ Common Internet File System (CIFS 1.0) โดยแต่ละรุ่นมีรายละเอียดดังนี้

3.6.1 PC-NET 1.0 (Core)

เป็นรุ่นแรกสุดของโพรโตคอลเอสเอ็มบีในระบบเครือข่ายของวินโดวส์เมื่อประมาณปี 1987 โดยออกมาพร้อมกับโปรแกรม MS-NET/PC-NET บางครั้งเรียกว่า "PC LAN 1.0" เป็นตัวที่มีความสามารถต่ำสุด และจัดได้ว่าเป็นพื้นฐานของเอสเอ็มบีเซิร์ฟเวอร์ โดยโปรแกรมที่เป็นเอสเอ็มบีเซิร์ฟเวอร์ควรมีความสามารถเทียบเท่าเป็นอย่างน้อย จึงจัดได้ว่าเป็นแกนหลัก (Core) ของโพรโตคอลเอสเอ็มบีทั้งหลาย มีกลุ่มคำสั่งหลักดังนี้

- Negotiate Protocol ใช้ในการเจรจาระหว่างไคลเอนต์และเซิร์ฟเวอร์ว่าในการติดต่อครั้งนั้นมีส่วนขยาย (protocol extension) อะไรบ้าง

- Tree Connect / Disconnect ใช้ในการขอเชื่อมต่อเพื่อแชร์ดิสก์หรือเครื่องพิมพ์จากเซิร์ฟเวอร์ โดยจะต้องมีการแจ้งชื่อผู้ใช้และรหัสผ่านเพื่อขอเข้าใช้ระบบ เพื่อให้ได้หมายเลขของแผนภูมิต้นไม้ (Tree ID) กลับมา โดยหมายเลขที่ได้มานี้ใช้ในการอ้างอิงเพื่อการทำงานในคำสั่งถัดไป

- File Access Operation เป็นคำสั่งที่ใช้ทำงานกับไฟล์โดยตรง เช่น สร้างหรือดูข้อมูลในไฟล์ โดยเรียกผ่านไฟล์แฮนเดิ้ล (File handle)

- Other File Operation เป็นคำสั่งที่ทำงานกับไฟล์โดยตรงเช่นกัน แต่ไม่ต้องใช้ไฟล์แฮนเดิ้ล เช่น สร้างไดเร็กทอรี อ่านหรือปรับเปลี่ยนค่าแอททริบิว

- Other Miscellaneous Command คำสั่งอื่นๆ นอกเหนือจากนี้ เช่น คำสั่งในการค้นหา

3.6.2 Microsoft Network Program 1.03 (Core Plus)

ส่วนขยายนี้ออกมาประมาณปี 1988 โดยได้มีการเพิ่มเติมคำสั่งย่อยที่ใช้ทำงานหลายๆ คำสั่งเดิมรวมกัน เพื่อเพิ่มความเร็วในการทำงาน เช่น คำสั่ง Lock&Read หรือ Write&Unlock

3.6.3 LAN Manager 1.0

มีคุณสมบัติต่างๆ เพิ่มเติมคือ ความสามารถในการสนับสนุนการทำงานของระบบปฏิบัติการแบบมัลติทาสก์ เช่น โอเอสทู (OS/2) ซึ่งอาจมีการเชื่อมต่อกับระบบเครือข่ายหลายๆ การเชื่อมต่อพร้อมกัน ให้ความสำคัญเกี่ยวกับการพิสูจน์ทราบตัวตนของผู้ใช้ (Authentication) และมีการจัดการรหัสผ่านมากขึ้น โดยแต่เดิมในตัวแกนหลักของโพรโตคอลนั้นสามารถทำงานแบบการพิสูจน์ตนโดยแบ่งตามระดับผู้ใช้งาน (User-level authentication) ได้ แต่ไคลเอนต์ต้องแจ้งชื่อผู้ใช้และรหัสผ่านทุกครั้งที่ติดต่อขอใช้ทรัพยากร ส่วนใน LAN Manager 1.0 ขึ้นไปนั้น ผู้ใช้ทำการล็อกออนเพียงครั้งเดียวแล้วได้โทเคน (token) ซึ่งสามารถนำโทเคนนี้ไปอ้างอิงในการเรียกใช้ทรัพยากรอื่นๆ บนเครื่องเซิร์ฟเวอร์ได้ทันที โดยไม่ต้องแจ้งชื่อและรหัสผ่านใหม่ทุกครั้ง นอกจากนี้ยังเพิ่มความสามารถในการล็อกไฟล์แบบ Op-lock เพื่อเพิ่มความเร็วในการทำงานโดยยอมให้ผู้ใช้อ่านข้อมูลไปพักไว้ที่เครื่องตนเอง ไม่ต้องอ่านเขียนผ่านระบบเครือข่ายทุกครั้ง และยังสนับสนุนการอ้างชื่อกับไดเรกทอรีปัจจุบันโดยเขียนเป็นสัญลักษณ์ "." กับ ".." แบบที่ใช้งานกันในดอส

3.6.4 LAN Manager 2.0 (LM 2.0)

มีส่วนเพิ่มเติมหลักๆ คือสนับสนุนชื่อไฟล์แบบยาว (Long file name) โดยเพิ่มคำสั่งสำหรับใช้งานกับชื่อไฟล์ประเภทนี้ขึ้นมาใหม่ทั้งชุด และยอมให้เซิร์ฟเวอร์เลือกว่าให้ใช้งานชื่อไฟล์แบบตัวอักษรตัวใหญ่-ตัวเล็กไม่เหมือนกัน (Case sensitive) นอกจากนี้ยังแก้ไขข้อผิดพลาดทั่วไปและเพิ่มคำสั่งแบบรวม (Compound command) เพื่อให้ทำงานเร็วขึ้นโดยพิมพ์คำสั่งน้อยครั้งลง เช่น "file open and read"

3.6.5 NT LAN Manager 0.12 (NTLM 0.12)

เพิ่มความสามารถเฉพาะของวินโดวส์เอ็นทีเข้าไป เช่น การเข้ารหัสข้อมูล (Encryption)

3.6.6 Common Internet File System (CIFS 1.0)

ในปัจจุบันยังไม่มีมาตรฐานที่ชัดเจน โดยมีจุดเริ่มต้นมาจากซันไมโครซิสเต็ม ซึ่งได้ออกแบบ WebNFS ขึ้นมาโดยมีแนวคิดว่าการรับส่งไฟล์ผ่านเครือข่ายอินเตอร์เน็ตควรทำได้ง่ายเหมือนการก๊อปปี้ไฟล์ในระบบเครือข่ายท้องถิ่นไม่น่าใช้โพรโตคอลการส่งผ่านข้อมูล (File Transfer protocol: FTP) จากแนวคิดนี้ได้มีการรวมกลุ่มเพื่อร่างเป็นมาตรฐานเดียวกันโดยให้ชื่อว่า Internet Draft for Common Internet File System: CIFS ซึ่งยังไม่เป็นมาตรฐานที่แน่ชัด แต่จัดว่าเป็นข้อกำหนดของโพรโตคอลสำหรับการแชร์ไฟล์แบบสาธารณะ

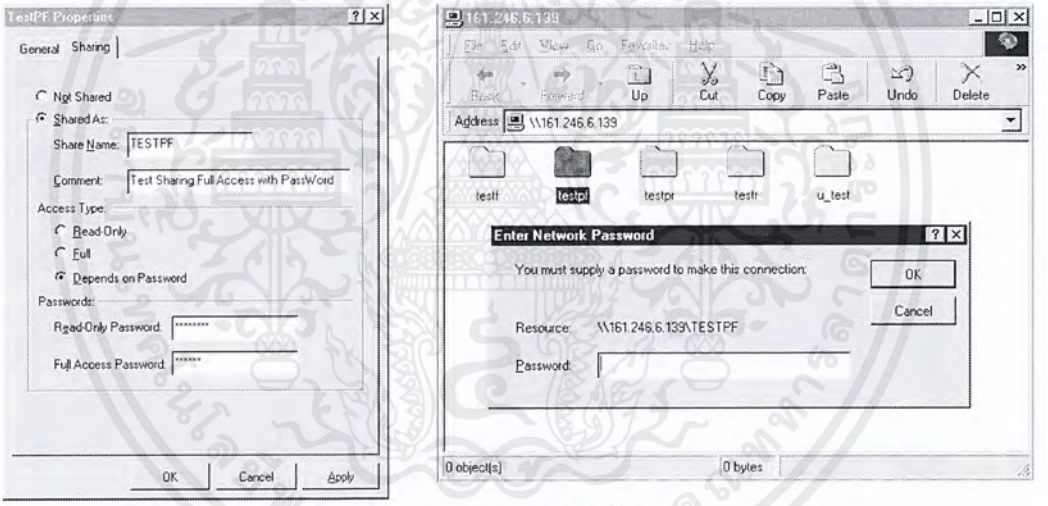
3.7 กลไกการรักษาความปลอดภัย

การติดต่อเพื่อขอใช้ทรัพยากรในระบบเครือข่ายของวินโดวส์โดยโพรโตคอลเอสเอ็มบีนั้นต้องให้เซิร์ฟเวอร์ตรวจสอบการพิสูจน์ตนของผู้ใช้งานก่อนการยอมรับให้ใช้งานทรัพยากรของตน โดยเครื่องไคลเอนต์เป็นฝ่ายส่งข้อมูลการพิสูจน์ตนไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ยอมรับจึงให้มีการเข้าใช้

ทรัพยากรได้ ในเอสเอ็มบีมีกลไกการรักษาความปลอดภัยที่เซิร์ฟเวอร์สามารถเลือกใช้ได้ว่าให้ไคลเอนต์เข้าใช้งานในระดับใด โดยการรักษาความปลอดภัยแบ่งเป็น 2 รูปแบบ คือ การรักษาความปลอดภัยแบ่งตามระดับการแชร์ (Share-level security) และ การรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งาน (User-level security) โดยแต่ละแบบมีรายละเอียด ดังนี้

3.7.1 การรักษาความปลอดภัยแบ่งตามระดับการแชร์ (Share-level security)

คือ การกำหนดรหัสผ่านที่ทรัพยากรแต่ละตัว ให้มีหลายรหัสผ่านแบ่งเป็นรหัสผ่านแบบเข้าอ่านได้อย่างเดียว (Read only) และรหัสผ่านสำหรับการเข้าอ่านและแก้ไขข้อมูลได้ (Full access) การกำหนดรหัสผ่านแบบนี้ เมื่อผู้ใช้ต้องการติดต่อเข้ามายังเซิร์ฟเวอร์และมีการถามรหัสผ่าน ผู้ใช้ต้องป้อนรหัสผ่านที่ตนทราบ และได้สิทธิ์ในการเข้าใช้ทรัพยากรตามที่กำหนดไว้ในรหัสผ่านนั้น การรักษาความปลอดภัยแบบนี้มีความยุ่งยาก คือ เมื่อมีหลายทรัพยากรในการแชร์และแต่ละทรัพยากรต้องการควบคุมการทำงานจึงต้องมีรหัสผ่าน 1 ถึง 2 ตัวกำกับอยู่ ทำให้ผู้ใช้ต้องจำรหัสผ่านจำนวนมากสำหรับการแชร์แต่ละอย่าง ทำให้ไม่เหมาะสำหรับงานในระบบขนาดใหญ่ที่มีการแชร์ทรัพยากรจำนวนมาก



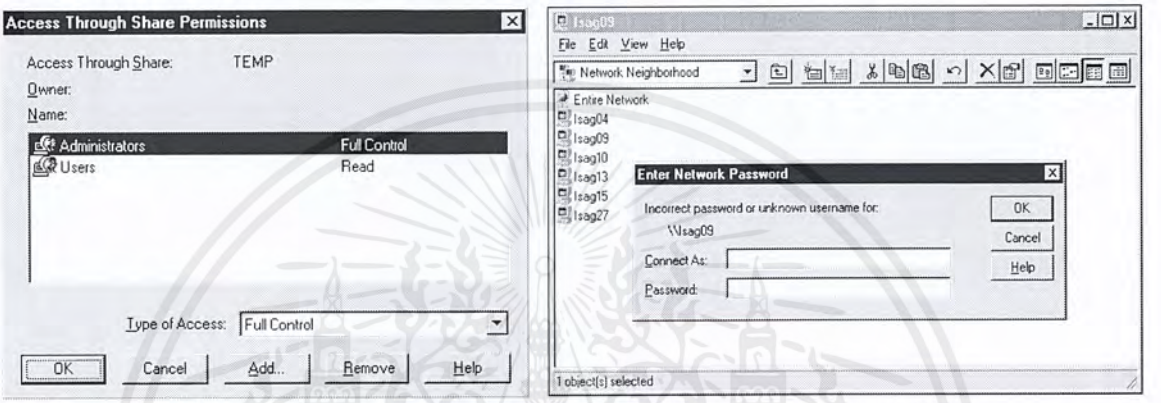
รูปที่ 3-3 การตั้งค่าและใช้งานการรักษาความปลอดภัยแบ่งตามระดับการแชร์

จากรูปที่ 3-3 ผู้ใช้สามารถเห็นทรัพยากรก่อนการเข้าใช้งานได้ และเมื่อต้องการใช้งานจึงมีการถามรหัสผ่าน เมื่อผู้ใช้ใส่รหัสผ่านแบบอ่านอย่างเดียวก็ได้สิทธิ์ในการเข้าใช้ทรัพยากรแบบอ่านได้อย่างเดียว และถ้าผู้ใช้ใส่รหัสผ่านแบบอ่านเขียนได้ก็ได้สิทธิ์ในการเข้าใช้ทรัพยากรแบบอ่านเขียนได้ และเมื่อต้องการเข้าใช้ทรัพยากรอื่นบนเครื่อง ถ้าทรัพยากรนั้นมีรหัสผ่านผู้ใช้ก็ต้องใส่รหัสผ่านของทรัพยากรนั้นก่อนการเข้าใช้งานอีกครั้ง

3.7.2 การรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งาน (User-level security)

คือ การกำหนดสิทธิ์โดยตั้งชื่อผู้ใช้ขึ้นมาพร้อมทั้งกำหนดว่าให้ผู้ใช้ใดมีสิทธิ์ในการใช้ทรัพยากรบนเครื่องนั้นอย่างไรบ้าง เช่น ให้มีสิทธิ์ในการใช้ทรัพยากรทุกตัวตัวที่มีแบบทำได้ทุกอย่าง (Full access)

หรือ ให้ใช้ทรัพยากรได้บางส่วนและใช้งานแบบอ่านอย่างเดียว การรักษาความปลอดภัยแบบนี้มีการกำหนดสิทธิ์ที่ซับซ้อนขึ้น ไม่ได้จำกัดอยู่เพียงอ่านอย่างเดียวและใช้งานได้เต็มที่เท่านั้น เมื่อกำหนดสิทธิ์ต่างๆ ผูกติดไว้กับผู้ใช้แล้ว เมื่อผู้ใช้ขอเข้ามาใช้ทรัพยากรที่เซิร์ฟเวอร์ ต้องประกาศตัวให้เซิร์ฟเวอร์ทราบก่อนว่าเป็นใคร โดยล็อกออนด้วยการแจ้งชื่อ (User name) และรหัสผ่านประจำตัวให้ถูกต้อง จึงได้รับสิทธิ์ทั้งหมดของผู้ใช้ที่กำหนดไว้บนเซิร์ฟเวอร์นั้น ซึ่งการล็อกออนนี้ทำจากเครื่องใดบนระบบเครือข่ายก็ได้ เพราะเป็นเพียงการส่งข้อมูลไปถามที่เซิร์ฟเวอร์ว่าผู้ใช้ที่แจ้งชื่อนี้มีสิทธิ์เข้าใช้ทรัพยากรบนเซิร์ฟเวอร์หรือไม่ ถ้ามีชื่อและรหัสผ่านถูกต้องก็ได้รับสิทธิ์เข้าใช้ทรัพยากรตามที่เซิร์ฟเวอร์กำหนดไว้



รูปที่ 3-4 การตั้งค่าและใช้งานการรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งาน

จากรูปที่ 3-4 เห็นได้ว่าเมื่อต้องการเข้าใช้ทรัพยากร ผู้ใช้ต้องแจ้งชื่อผู้ใช้และรหัสผ่านก่อนการเข้าใช้งาน ซึ่งแตกต่างจากการรักษาความปลอดภัยแบบแบ่งตามระดับการแชร์ที่แจ้งรหัสผ่านเพียงอย่างเดียวก็เข้าใช้งานทรัพยากรได้ แต่การรักษาความปลอดภัยแบบแบ่งตามระดับของผู้ใช้งานนี้ ผู้ใช้ไม่สามารถเห็นทรัพยากรบนเครื่องก่อนการล็อกอินได้ แต่ภายหลังการล็อกอินแล้วผู้ใช้จะได้สิทธิ์เข้าใช้ทรัพยากรทุกตัวตามที่กำหนดไว้บนเซิร์ฟเวอร์

3.7.3 การรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งานแบบควบคุมโครงสร้างผ่านโดเมน

จัดเป็นอีกรูปแบบหนึ่งของการรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งาน โดยในระบบที่ใช้การรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งานนั้น ช่วยให้ผู้ใช้จำรหัสผ่านในการเข้าใช้ทรัพยากรน้อยลง เหลือเพียงจำชื่อผู้ใช้และรหัสผ่านเพื่อเข้าใช้ทรัพยากรของเซิร์ฟเวอร์นั้น แต่ถ้ามีหลายเซิร์ฟเวอร์ผู้ใช้ก็ยังคงต้องจำชื่อผู้ใช้และรหัสผ่านของแต่ละเซิร์ฟเวอร์ และต้องมีการล็อกอินไปยังทุกเซิร์ฟเวอร์ที่แยกจากกัน จึงได้มีการพัฒนาระบบที่มอบหมายให้เครื่องใดเครื่องหนึ่งในระบบเครือข่ายมีหน้าที่ในการเก็บชื่อผู้ใช้และรหัสผ่านสำหรับการเข้าใช้ทรัพยากรทุกตัวบนทุกเครื่องในระบบเครือข่ายไว้ในที่เดียว และให้ผู้ใช้ล็อกอินที่เครื่องนั้นเพียงครั้งเดียวแล้วเข้าใช้งานเครื่องอื่นบนระบบเครือข่ายได้ทุกเครื่อง วิธีการเช่นนี้ทำให้เกิดระบบเครือข่ายที่เรียกว่า “โดเมน” (Domain) เครื่องที่ทำหน้าที่ในการเก็บชื่อผู้ใช้และรหัสผ่านนี้เรียกว่า “ตัวควบคุมโดเมน” (Domain Controller) โดยข้อมูลต่างๆ ซึ่งประกอบด้วยรายชื่อผู้ใช้, รหัสผ่าน, ทรัพยากรต่างๆ ที่แชร์และสิทธิ์ในการเข้าใช้งาน เรียกรวมกันว่าเป็นฐานข้อมูล “Security Account

Manger: SAM" เมื่อผู้ใช้ล็อกอินเข้ามายังเครื่องที่มีฐานข้อมูลนี้อยู่ เครื่องไคลเอนต์ของผู้ใช้นั้นจะได้รับรหัสย่อยที่เรียกว่า "โทเคน" (token) สำหรับไปเรียกใช้ทรัพยากรบนเครื่องอื่นๆ ในระบบเครือข่ายโดยอัตโนมัติ โดยที่ผู้ใช้ไม่สามารถเห็นหรือรับรู้รหัสนี้แต่อย่างใด

ข้อดีของการล็อกอินแบบโดเมน คือ ช่วยให้ผู้ใช้จำชื่อและรหัสผ่านเพียงชุดเดียวในการเข้าใช้ทรัพยากร และการกำหนดสิทธิการเข้าใช้งานก็ทำที่ตัวควบคุมโดเมนเพียงที่เดียวไม่ต้องทำทุกเครื่อง แต่ข้อจำกัดในการใช้งานโดเมน คือ เครื่องในเครือข่ายต้องใช้ระบบปฏิบัติการวินโดวส์เอ็นทีและเครื่องที่เป็นตัวควบคุมโดเมนต้องเป็นวินโดวส์เอ็นทีเซิร์ฟเวอร์เท่านั้น

3.8 ความสามารถของระบบปฏิบัติการวินโดวส์ในการรักษาความปลอดภัยทรัพยากร

ระบบปฏิบัติการวินโดวส์แต่ละชนิดมีความสามารถในการรักษาความปลอดภัยในการแชร์ทรัพยากรที่ต่างกัน โดยแต่ละชนิดมีความสามารถ ดังนี้

ระบบปฏิบัติการ	ความสามารถในการรักษาความปลอดภัย
Windows 3.x	ทำงานแบบรักษาความปลอดภัยแบ่งตามระดับการแชร์ได้อย่างเดียว และล็อกอินเข้าเป็นส่วนหนึ่งของโดเมนเพื่อดึงข้อมูลบางอย่างไปใช้ได้ แต่ไม่สามารถจัดทรัพยากรของตัวเองร่วมในกลไกการควบคุมผ่านโดเมนได้
Windows 95/98	ทำงานได้ทั้งแบบแบบรักษาความปลอดภัยแบ่งตามระดับการแชร์และการรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งาน โดยล็อกอินเข้าเป็นส่วนหนึ่งของโดเมนได้ แต่ไม่สามารถจัดทรัพยากรของตัวเองร่วมในกลไกการควบคุมผ่านโดเมนได้
Windows NT Workstation	ทำงานได้ทั้งแบบแบบรักษาความปลอดภัยแบ่งตามระดับการแชร์และการรักษาความปลอดภัยแบ่งตามระดับของผู้ใช้งาน และสามารถจัดทรัพยากรของตัวเองร่วมในกลไกการควบคุมผ่านโดเมนได้
Windows NT Server	ทำได้ทุกอย่างและเป็นตัวควบคุมโดเมนได้ด้วย

ตารางที่ 3-1 ความสามารถของระบบปฏิบัติการวินโดวส์ในการรักษาความปลอดภัยทรัพยากร

จากตารางที่ 3-1 เห็นได้ว่า วินโดวส์ 95/98 ไม่สามารถควบคุมและพิสูจน์สิทธิ์ผ่านโดเมนได้อย่างเต็มที่ เพียงแต่เข้าใช้งานโดเมนได้บางอย่าง ส่วนตัวที่ทำหน้าที่เป็นตัวควบคุมโดเมนได้มีเพียงวินโดวส์เอ็นทีเซิร์ฟเวอร์เท่านั้น

3.9 การอ้างอิงชื่อเครื่องในระบบเครือข่าย

ในการใช้งานเน็ตบีทีนั้นเมื่อเครือข่ายมีขนาดใหญ่ขึ้นโดยแบ่งเป็นหลายสับเน็ตที่ไม่สามารถส่งข้อมูลแบบกระจายถึงกันได้โดยตรงทุกเครื่องแล้ว มีความจำเป็นต้องมีกลไกเพื่อการอ้างชื่อ หรือเป็นตัวที่

บอกได้ว่าเครื่องที่มีชื่อตามแบบเน็ตไบออสเครื่องนั้นมีหมายเลขไอพีเป็นเท่าใด เพื่อให้โพรโตคอลทีซีพีไอพีสามารถทำงานต่อได้ ซึ่งมีด้วยกันอยู่ 2 วิธี คือ มีเซิร์ฟเวอร์คอยรวบรวมรายชื่อเหล่านี้ (Windows Internet Name Service: WINS) หรือการใช้ไฟล์ LMHOSTS (LAN Manager Hosts) เพื่อจัดเก็บรายชื่อและหมายเลขไอพีโดยตรง

3.8.1 Windows Internet Name Service: WINS

เป็นระบบที่พัฒนาโดยไมโครซอฟท์ เป็นการทำงานในลักษณะของพีโนด (Point-to-point node: p-node) โดยเมื่อแต่ละเครื่องในระบบเครือข่ายเริ่มการทำงานก็ต้องไปลงทะเบียนให้ตนเองมีชื่ออยู่ในบัญชีรายชื่อของเซิร์ฟเวอร์เครื่องหนึ่งซึ่งทำหน้าที่รวบรวมรายชื่อเครื่องในระบบเครือข่ายโดยเฉพาะ เซิร์ฟเวอร์ดังกล่าวทำงานตามโพรโตคอลชื่อเอ็นบีเอ็นเอส (NetBIOS Name Service: NBNS) ซึ่งคอยดูแลให้แต่ละเครื่องหรือแต่ละไอพีมีชื่อที่ไม่ซ้ำกัน และถ้าเครื่องใดต้องการหยุดทำงานก็ต้องไปแจ้งที่เซิร์ฟเวอร์นี้ให้ลบชื่อออกด้วย ด้วยวิธีนี้ทุกเครื่องในระบบเครือข่ายถึงแม้ว่าอยู่คนละสับเน็ตกัน ก็สามารถติดต่อกันโดยการอ้างชื่อได้ ซึ่งต่อมาไมโครซอฟท์นำโพรโตคอลนี้มาพัฒนาเป็น Windows Internet Name Service: WINS แต่ยังไม่สมบูรณ์นักเพราะทำงานได้บนระบบเครือข่ายขนาดเล็กเท่านั้น ไม่สามารถจัดการกับรายชื่อบนระบบเครือข่ายขนาดใหญ่อย่างอินเตอร์เน็ตได้

การทำงานของเซิร์ฟเวอร์รายชื่อนั้นเมื่อมีเครื่องใหม่เพิ่มเข้ามาในระบบเครือข่าย เซิร์ฟเวอร์รายชื่อต้องทำการบันทึกชื่อและหมายเลขไอพีของเครื่องนั้นตลอดจนเวิร์คกรุ๊ปหรือโดเมนที่เครื่องนั้นเป็นสมาชิกอยู่ และเมื่อมีเครื่องใดต้องการอ้างชื่อถึงเครื่องอื่นก็ต้องมาถามที่เซิร์ฟเวอร์รายชื่อนี้ก่อน ถ้าไม่พบชื่อที่ต้องการจึงพยายามค้นหาเองโดยการส่งแบบกระจายทั่วไป ถ้ายังไม่พบอีกก็แจ้งให้ผู้ใช้ทราบว่าไม่พบเครื่องหรือทรัพยากรที่แชร์ไว้

ในกรณีที่มีเซิร์ฟเวอร์รายชื่อหลายๆ เครื่องนั้น โดยทั่วไปต้องมีกระบวนการในการทำสำเนารายชื่อจากเซิร์ฟเวอร์หนึ่งไปยังเซิร์ฟเวอร์เครื่องอื่นให้ตรงกัน (Replication) และยังสามารถกำหนดได้ว่าให้เครื่องใดเป็นเซิร์ฟเวอร์หลัก (Primary WINS Server) และเครื่องอื่นให้เป็นเซิร์ฟเวอร์สำรอง (Backup WINS Server) แต่อาจเกิดปัญหาขึ้นในกระบวนการปรับข้อมูลให้ตรงกันระหว่างเซิร์ฟเวอร์หลักและเซิร์ฟเวอร์สำรองหรือระหว่างเซิร์ฟเวอร์รายชื่อหลายๆ ตัวมีการปรับในทุกระยะเวลาคงที่ค่าใดค่าหนึ่งเท่านั้น หากยังไม่ถึงเวลาที่กำหนด ข้อมูลในแต่ละเซิร์ฟเวอร์อาจไม่ตรงกันได้ทั้งๆ ที่อยู่บนระบบเครือข่ายเดียวกันซึ่งไม่ถูกต้อง และหากมีการถามชื่อเครื่องไปที่เซิร์ฟเวอร์รายชื่อที่ไม่รู้จักเครื่องนั้นหรือไม่มีอยู่ในรายชื่อ เซิร์ฟเวอร์จะตอบทันทีว่าไม่มีเครื่องที่ระบุ โดยที่ไม่ยอมไปถามยังเซิร์ฟเวอร์ตัวอื่น

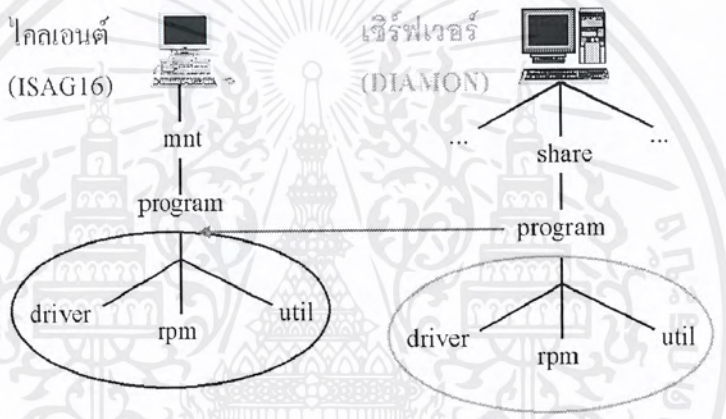
3.8.2 ไฟล์ LMHOSTS

ไฟล์ LMHOSTS หรือ LAN Manager Hosts เทียบได้กับไฟล์รายชื่อเครื่องหรือโฮสต์ (host) คือไฟล์ /etc/host ในระบบยูนิกซ์หรือไฟล์ hosts ในเครื่องที่ใช้ทีซีพีไอพีทั่วไปโดยในไฟล์นี้เป็นการกำหนดว่าเครื่องใดมีหมายเลขไอพีเป็นเท่าใดและอยู่ในโดเมนใดตายตัวไว้เลย รวมทั้งยังบอกวิธีให้ไปเรียกไฟล์ LMHOSTS มาจากไดเรกทอรีที่แชร์กันอยู่บนเซิร์ฟเวอร์มาใช้ได้อย่างไร

บทที่ 4
การใช้ไดเร็กทอรีร่วมกันบนระบบยูนิกซ์

4.1 เบื้องต้นการแชร์บนระบบยูนิกซ์

การใช้ไฟล์หรือไดเร็กทอรีร่วมกันบนระบบยูนิกซ์ คือ การที่เครื่องใดเครื่องหนึ่งในระบบเครือข่ายยูนิกซ์ได้แชร์ไดเร็กทอรีไว้ให้เครื่องอื่นที่มีสิทธิ์เข้าใช้ได้ โดยเครื่องที่เปิดแชร์นั้นทำหน้าที่เป็นเซิร์ฟเวอร์ และเครื่องที่เข้าใช้งานไดเร็กทอรีถือเป็นไคลเอนต์ การที่ไคลเอนต์เข้าไปใช้ไดเร็กทอรีที่เซิร์ฟเวอร์ได้แชร์ไว้นั้น ต้องเมาต์ (mount) ก่อนการเข้าใช้งาน และการเมาต์ไดเร็กทอรีจากเซิร์ฟเวอร์นั้นเปรียบเสมือนว่าไดเร็กทอรีจากเซิร์ฟเวอร์มาอยู่ที่ไดเร็กทอรีที่เครื่องไคลเอนต์นั้น



รูปที่ 4-1 ลักษณะการเมาต์

จากรูปที่ 4-1 คือ การเมาต์ไดเร็กทอรี /share/program จากเซิร์ฟเวอร์มาไว้ที่ไดเร็กทอรี /mnt/program ของไคลเอนต์และเสมือนว่าไดเร็กทอรี /share/program จากเซิร์ฟเวอร์นั้นมาอยู่ที่ไดเร็กทอรี /mnt/program ของเครื่องไคลเอนต์

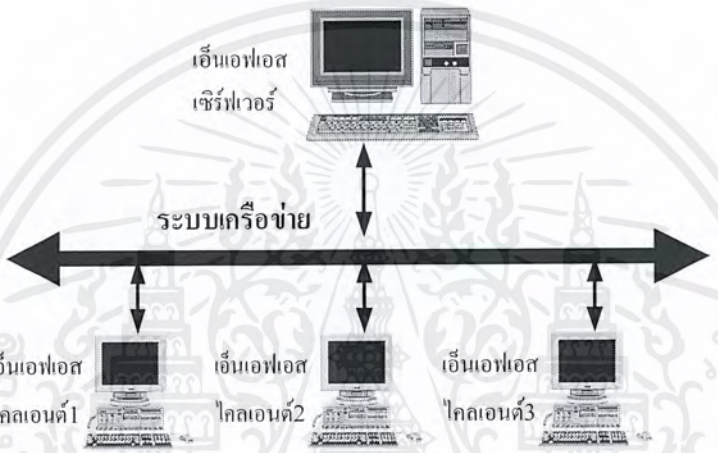
หัวใจสำคัญในการแชร์ไดเร็กทอรีบนระบบเครือข่ายยูนิกซ์ คือโพรโตคอลเอ็นเอฟเอส (NFS: Network File System) ซึ่งพัฒนาโดย ซัน ไมโครซีสเท็มส์ (Sun Microsystems) จากเวอร์ชัน 1.0 จนถึงเวอร์ชัน 3.0 และได้รับการพัฒนาต่อโดย ไออีทีเอฟ (IETF) เป็นเวอร์ชัน 4.0 ต่อมา

4.2 โพรโตคอลที่ใช้ในการแชร์ในระบบยูนิกซ์

โพรโตคอลเอ็นเอฟเอส (NFS: Network File System) พัฒนาขึ้นโดยซันไมโครซีสเท็มส์ เป็นโพรโตคอลที่ให้ผู้ใช้สามารถใช้งานไฟล์ร่วมกัน (File sharing) การแชร์ไฟล์โดยเอ็นเอฟเอสนั้น นับเป็นการแชร์ไฟล์แบบมาตรฐานของระบบยูนิกซ์ มีลักษณะการทำงานเป็นแบบไคลเอนต์เซิร์ฟเวอร์ โดยการ

ทำงานต้องรันเดมอน (daemon) ต่างๆ ไว้ที่ไคลเอนต์และเซิร์ฟเวอร์ ในส่วนของเซิร์ฟเวอร์ใช้เมาต์เดมอน (mountd) ซึ่งมีหน้าที่รับความต้องการในการขอเมาต์จากไคลเอนต์ และเอ็นเอฟเอสเดมอน (nfsd) ซึ่งมีหน้าที่รอรับความต้องการในการโอนถ่ายข้อมูล

การทำงานของเอ็นเอฟเอสในส่วนไคลเอนต์นั้น ใช้คำสั่งเมาต์เพื่อเชื่อมต่อไดเร็กทอรีของเครื่องเซิร์ฟเวอร์บนระบบเครือข่ายให้เป็นเสมือนว่าไดเร็กทอรีนั้นอยู่บนเครื่องของตน และสามารถทำงานกับไดเร็กทอรีนั้นได้เสมือนเป็นไดเร็กทอรีของตนเอง แต่ต้องอยู่ภายใต้การควบคุมสิทธิ์ของการเข้าถึงซึ่งกำหนดโดยเซิร์ฟเวอร์ การใช้งานการแชร์ไดเร็กทอรีโดยเอ็นเอฟเอสนั้นเครื่องคอมพิวเตอร์เครื่องหนึ่งสามารถเป็นได้ทั้งไคลเอนต์และเซิร์ฟเวอร์ขึ้นอยู่กับความต้องการของผู้ใช้งานในขณะนั้น



รูปที่ 4-2 การแบ่งของ เอ็นเอฟเอส ระหว่างไคลเอนต์กับเซิร์ฟเวอร์

ก่อนการเข้าใช้งานไดเร็กทอรีซึ่งแชร์โดยเอ็นเอฟเอสนั้น ไคลเอนต์ต้องพิสูจน์สิทธิ์กับเซิร์ฟเวอร์ก่อน การพิสูจน์สิทธิ์ในเอ็นเอฟเอส เซิร์ฟเวอร์ตรวจสอบจากหมายเลขไอพีของเครื่องไคลเอนต์

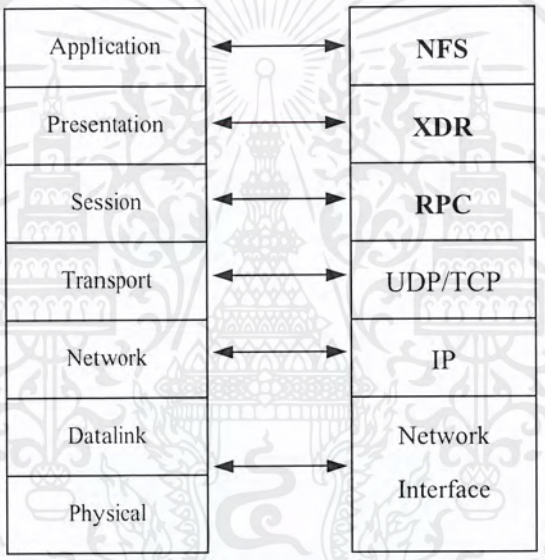
4.2.1 การทำงานของเอ็นเอฟเอส

เอ็นเอฟเอสทำงานแบบไม่จำสถานะ (Stateless) กล่าวคือ การที่ไม่เก็บสถานะปัจจุบันของการติดต่อหรืออ่านเขียนข้อมูล โดยเปิดโอกาสให้เครื่องอื่นเข้ามาอ่านข้อมูลไปใช้หรือแก้ไขข้อมูลได้โดยตรงเท่านั้น และไม่สามารถให้เซิร์ฟเวอร์ล็อกข้อมูลหรือจดจำสถานะเพื่อป้องกันในกรณีการแก้ไขข้อมูลชุดเดียวกันโดยผู้ใช้หลายคน ในการใช้งานจึงจัดสรรให้แต่ละไฟล์มีผู้เข้ามาใช้ได้ที่ละคน และจากการทำงานในไม่จำสถานะนี้จึงไม่ต้องมายุ่งยากในการบำรุงรักษาหรือมีการกู้ข้อมูล (Recover) ทำให้เอ็นเอฟเอสมีการทำงานได้อย่างต่อเนื่องเมื่อไคลเอนต์หรือเซิร์ฟเวอร์เสียหายมากกว่าระบบที่จำสถานะ

เนื่องจากเอ็นเอฟเอสเป็นโพรโตคอลที่ไม่เก็บสถานะของการทำงาน ไคลเอนต์และเซิร์ฟเวอร์จึงไม่ต้องจดจำการทำงานเกี่ยวกับไฟล์ครั้งก่อน ไคลเอนต์อาจเปิดไฟล์หนึ่งโดยการส่งชื่อของไฟล์ไปยังเซิร์ฟเวอร์ และไคลเอนต์จะได้รับการตอบกลับมาว่าขณะนี้ไฟล์ถูกเปิดใช้งานอยู่ และเมื่อไคลเอนต์

ต้องการอ่านไฟล์ จึงร้องขอเพื่ออ่านไฟล์ไปยังเซิร์ฟเวอร์โดยส่งชื่อและตำแหน่งปัจจุบันของไฟล์ เพื่อที่เซิร์ฟเวอร์จะไม่ต้องเก็บสถานะการเปิดไฟล์ การทำงานแบบนี้มีข้อเสีย คือ เอ็นเอฟเอสทำงานช้าเมื่อเทียบกับระบบอื่น อันเป็นผลมาจากปริมาณข้อมูลบนระบบเครือข่ายที่เพิ่มขึ้นจากแพ็กเกจขนาดใหญ่และจากการประมวณผลที่เพิ่มขึ้นสำหรับแต่ละแพ็กเกจ แต่มีข้อดี คือ การที่ไม่เก็บสถานะ และเมื่อเซิร์ฟเวอร์หยุดทำงาน ไคลเอนต์จะส่งการร้องขอไปใหม่จนกระทั่งได้รับการตอบสนอง โดยไคลเอนต์ไม่สามารถแบ่งแยกความแตกต่างระหว่างเซิร์ฟเวอร์ที่ช้ากับเซิร์ฟเวอร์ที่หยุดทำงานได้ และเมื่อเซิร์ฟเวอร์ถูกรีบูตก็สามารถตอบสนองการร้องขอของไคลเอนต์ได้ทันทีโดยไม่ต้องรีบูตไคลเอนต์ด้วย เอ็นเอฟเอสใช้โพรโตคอลแบบยูดีพี (UDP) ในการส่งและรับการร้องขอไฟล์

4.2.2 โครงสร้างของเอ็นเอฟเอส



รูปที่ 4-3 โครงสร้างของเอ็นเอฟเอสเปรียบเทียบกับมาตรฐานโอเอสไอ

จากรูปที่ 4-3 เห็นได้ว่าเอ็นเอฟเอสนั้นทำงานอยู่ในระดับแอปพลิเคชั่น และเรียกใช้งานเอ็กซ์ดีอาร์อาร์พีซีซึ่งอยู่ในระดับของเซสชันเพื่อใช้เรียกคำสั่งต่างๆ ออกไปยังระบบเครือข่าย และใช้โพรโตคอลทีซีพีไอพีเป็นโพลโตคอลระดับทรานสปอร์ทต่อไป

นอกเหนือจากเอ็นเอฟเอสแล้ว พอร์ตแมปเปอร์ (Portmapper) เป็นอีกตัวหนึ่งซึ่งใช้งานอาร์พีซีโดยมีหน้าที่แมปไปยังเซิร์ฟเวอร์ และเป็นตัวกำหนดการเชื่อมต่อแบบไคลเอนต์เซิร์ฟเวอร์โดยใช้อาร์พีซี การที่ไคลเอนต์ได้หมายเลขพอร์ตมาจากเซิร์ฟเวอ ร์นั้นเพื่อให้สามารถติดต่อไปยังเซิร์ฟเวอร์และใช้ประโยชน์จากเซิร์ฟเวอร์นั้นได้ โดยเริ่มแรกไคลเอนต์จะใช้พอร์ตแมปเปอร์ไปกำหนดแอปพลิเคชั่นที่ต้องการใช้งานบนเซิร์ฟเวอร์ ถ้าเป็นแอปพลิเคชั่นที่เซิร์ฟเวอร์เปิดให้บริการอยู่แล้ว เซิร์ฟเวอร์จะส่งหมายเลขพอร์ตที่ถูกต้องกลับมาให้ยังไคลเอนต์ และเครื่องไคลเอนต์จะใช้หมายเลขพอร์ตนี้จนจบสิ้นการใช้งาน

4.2.3 เอ็กซ์ดีอาร์และอาร์พีซี

- เอ็กซ์ดีอาร์ (XDR: External Data Representation) มีการทำงานเทียบเท่าชั้นพรีเซนเทชั่นในระบบโอเอสไอ โดนทำหน้าที่ปรับรูปแบบของข้อมูลให้เป็นข้อมูลที่มีรูปแบบเดียวกันเมื่อรับส่งระหว่างไคลเอนต์กับเซิร์ฟเวอร์ โดยใช้ในการแลกเปลี่ยนข้อมูลระหว่างระบบที่มีความแตกต่างกันของข้อมูล (Heterogeneous) เช่น แตกต่างกันที่ระบบไบต์ (byte orders), ชนิดของข้อมูล (EBCD กับ ASCII) และโครงสร้างข้อมูล (data structure)

เมื่อมีการรับส่งข้อมูลระหว่างไคลเอนต์เซิร์ฟเวอร์ เอ็กซ์ดีอาร์ทำหน้าที่แปลงข้อมูลให้อยู่ในรูปแบบข้อมูลมาตรฐานของเอ็กซ์ดีอาร์ก่อนการส่งไปในระบบเครือข่าย เพื่อให้ข้อมูลที่ส่งออกไปสามารถสื่อสารกันได้เข้าใจในระบบของเครื่องที่แตกต่างกัน

- อาร์พีซี (RPC: Remote Procedure Call) มีการทำงานเทียบเท่าชั้นเซสชั่นในระบบโอเอสไอ ซันไมโครซีสเท็มส์วางพื้นฐานการออกแบบของเอ็นเอฟเอสไว้บนแนวคิดของอาร์พีซี ซึ่งช่วยให้ซอฟต์แวร์ที่อยู่ต่างเครื่องกันสื่อสารกันได้

อาร์พีซีช่วยให้นักเขียนโปรแกรมสามารถสร้างโปรแกรมในระบบเครือข่ายของคอมพิวเตอร์ที่แตกต่างกันได้ โดยในการสร้างแอปพลิเคชั่นแบบไคลเอนต์เซิร์ฟเวอร์ ในส่วนของเซิร์ฟเวอร์โมดูลให้เป็นแบ็คเอนด์ (Back end) ของแอปพลิเคชั่นมีหน้าที่ในการประมวลผลข้อมูล ขณะที่ส่วนของไคลเอนต์โมดูลมีหน้าที่ติดต่อกับผู้ใช้ส่วนหน้า (Front end) โปรแกรมจะถูกสร้างเป็นอาร์พีซีสครีป ซึ่งแบ่งแยกระหว่างไคลเอนต์และเซิร์ฟเวอร์โมดูล จากนั้นใช้ตัวแปรอาร์พีซีเพื่อสร้างซอร์สโค้ดภาษาซีเพื่อเชื่อมโมดูลเข้าด้วยกัน โค้ดที่สร้างขึ้นจะสร้างเซสชันการสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์โมดูล บนคอมพิวเตอร์ต่างเครื่องกัน อาร์พีซีจึงเป็นเสมือนตัวเชื่อมโยงการทำงานในส่วน ของไคลเอนต์และเซิร์ฟเวอร์ให้ทำงานได้สัมพันธ์กัน

4.2.4 ส่วนอื่นๆ ที่เกี่ยวข้องกับเอ็นเอฟเอส

- เอ็นไอเอส (NIS: Network Information Services) คือ ตัวช่วยเสริมบริการของเอ็นเอฟเอส โดยการให้บริการค้นหาและติดต่อเครือข่าย โดยเครื่องเซิร์ฟเวอร์เอ็นไอเอ็สนี้เก็บข้อมูลที่เรียกว่าแมป (maps) ซึ่งบรรจุข้อมูลเกี่ยวกับผู้ใช้, กลุ่ม, ที่อยู่ในระบบเครือข่าย, เกตเวย์และสิ่งอื่นๆ ที่มีอยู่บนเครือข่าย สิ่งเหล่านี้ช่วยให้ไคลเอนต์สามารถค้นหาเซิร์ฟเวอร์ได้ง่ายขึ้น เอ็นไอเอ็สมีการทำงานเป็นเชลล์สคริปต์เพื่อช่วยในการสร้างแมปไฟล์ (map file) เช่น สคริปต์ชื่อ ypinit ซึ่งใช้สร้างแมปของเอ็นไอเอ็สเซิร์ฟเวอร์จากไฟล์ที่มีอยู่บนคอมพิวเตอร์ที่ใช้งานระบบยูนิกซ์ ร่วมทั้งรหัสผ่าน กรุ๊ป และคอนฟิกูเรชั่นไฟล์ เมื่อผู้ใช้แบ่งปันทรัพยากรบนเครือข่ายยูนิกซ์ เซิร์ฟเวอร์เอ็นไอเอ็สจะแยกแยะการกำหนดชื่อและแอดเดรสที่แตกต่างกันระหว่างคอมพิวเตอร์บนเครือข่าย โปรแกรมเอ็นไอเอ็สที่รู้จักกันคือเยลโลเพจ (The Yellow Page)

- เมาต์โพรโตคอล (Mount protocol) เป็นเครื่องมือที่จำเป็นต่อการใช้งานพื้นฐานในระบบปฏิบัติการยูนิกซ์ โดยเมาส์โพรโตคอลมีหน้าที่ในการกำหนดชื่อเส้นทางการเข้าถึงข้อมูล (Pathname) ของ

เซิร์ฟเวอร์, ตรวจสอบผู้ใช้, ตรวจสอบสิทธิ์ของการเข้าถึง ตัวอย่างคำสั่งในการเมาส์ เช่น mnt, dump, umnt, umntall, export

- ล็อกไฟล์ของเอ็นเอฟเอส (NFS Log file) เป็นตัวเก็บข้อมูลการเข้าใช้งานเพื่อให้ไคลเอนต์ที่เชื่อมต่อเข้ามาไม่ไปใช้ข้อมูลส่วนเดียวกัน เช่น เอ็นแอลเอ็มโพรโตคอล (NLM: Network lock manager) มีหน้าที่แสดงให้เห็นว่าไคลเอนต์ยังคงล็อคไฟล์อยู่ โดยการล็อคนั้นอาจล็อคแบบเอ็กคลูซีฟล็อค (exclusive lock) หรือแชร์ล็อค (share lock) ก็ได้

- ระบบแคชไฟล์ (CacheFS) เป็นส่วนช่วยให้ระบบมีประสิทธิ์ภาพมากขึ้น โดยลดภาระการทำงานของเซิร์ฟเวอร์และลดปริมาณข้อมูลในระบบเครือข่าย โดยไคลเอนต์ซึ่งต้องการติดต่อกับเซิร์ฟเวอร์ที่ทำงานช้าๆ ระบบแคชไฟล์ในการดึงข้อมูลมาพักทำให้ไคลเอนต์ทำงานได้เร็วขึ้น โดยเมื่อไคลเอนต์ขอใช้ไฟล์เป็นครั้งแรก ไฟล์นั้นจะถูกเก็บไว้ในแคช ในช่วงแรกนี้ระบบอาจทำงานช้าเพราะต้องอ่านข้อมูลผ่านระบบเครือข่ายโดยตรง แต่ถ้ามีการร้องขอใช้ไฟล์นั้นอีกครั้งไคลเอนต์จะค้นหาจากส่วนของแคชจึงไม่ต้องอ่านข้อมูลผ่านระบบเครือข่ายอีก ซึ่งช่วยให้ระบบทำงานได้เร็วขึ้น

- เอสอาร์ซี (SRC: System Resource Controller) คือ ชุดคำสั่งซึ่งใช้ในการควบคุมการทำงานและตรวจสอบสถานะของเอ็นเอฟเอส เช่น การเริ่มการทำงาน (startsrc), หยุดการทำงาน (stopsrc), ดูสถานะต่างๆ (lssrc) ของเอ็นเอฟเอส

- พีซีเอ็นเอฟเอส (PC-NFS) คือ โปรแกรมที่ใช้ในเครื่องพีซีที่เพื่อติดต่อกับเอ็นเอฟเอสเซิร์ฟเวอร์ โดยพีซีสามารถร้องขอที่อยู่ในระบบเครือข่ายและชื่อโฮสท์ (Host names) จากเอ็นเอฟเอสเซิร์ฟเวอร์และสามารถเข้าไปใช้บริการเครื่องพิมพ์จากเอ็นเอฟเอสเซิร์ฟเวอร์ได้

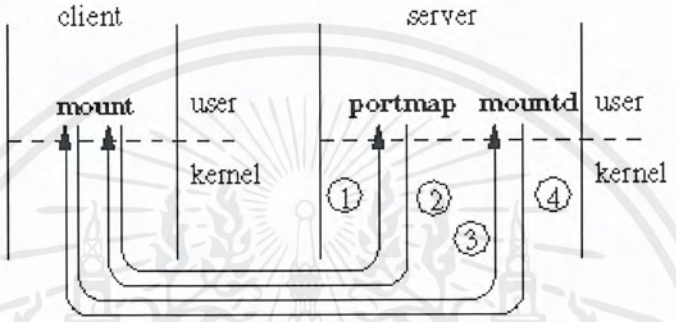
- เว็บเอ็นเอฟเอส (WebNFS) ออกแบบโดยซันไมโครซีสเท็มส์เพื่อช่วยให้การรับส่งไฟล์ระหว่างไคลเอนต์กับเซิร์ฟเวอร์ผ่านทางอินเทอร์เน็ตทำได้ง่ายขึ้น และมีลักษณะเสมือนว่าไฟล์นั้นอยู่บนเครื่องของผู้ใช้อยู่แล้ว โดยซันไมโครซีสเท็มส์หวังว่าจะนำไปแทนโปรแกรมเอฟทีพี (Ftp) ที่ใช้งานยุ่งยากและซับซ้อน เว็บเอ็นเอฟเอสออกแบบมาเพื่อส่งเสริมเว็บบราวเซอร์โดยใช้ยูอาแอล (URL: Universal Resource Locator) ในการเข้าถึง

- เอ็นเอฟเอสแบบมีระบบรักษาความปลอดภัย (Secure NFS) คือ เอ็นเอฟเอสที่ใช้ระบบรักษาความปลอดภัยแบบการเข้ารหัสข้อมูลแบบมาตรฐาน (Data Encryption Standard: DES) และการเข้ารหัสแบบกุญแจสาธารณะ (Public key cryptography) แต่ในมาตรฐานของเอ็นเอฟเอสแล้วอาจไม่มีระบบรักษาความปลอดภัยก็ได้ ในส่วนนี้เป็นส่วนที่เพิ่มเข้ามาช่วยเรื่องความปลอดภัยให้ดีขึ้นอีกระดับหนึ่ง

4.3 ขั้นตอนการติดต่อกันระหว่างไคลเอนต์กับเซิร์ฟเวอร์

การติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์นั้น ไคลเอนต์ต้องหาหมายเลขพอร์ตจากเซิร์ฟเวอร์เสียก่อนและเมื่อได้หมายเลขพอร์ตมาแล้ว ไคลเอนต์จะส่งคำสั่งในรูปแบบของอาร์พีซีออกไปโดยผ่านเคอร์แนลซึ่งรันเอ็นเอฟเอสเดมอน ทางด้านเซิร์ฟเวอร์ก็เอ็นเอฟเอสเดมอนอยู่เพื่อรอรับการติดต่อเช่นกัน

4.3.1 การเข้างานใช้ไดเร็กทอรี ไคลเอนต์สามารถเข้างานไดเร็กทอรีที่เซิร์ฟเวอร์ได้โดยการเมาต์ ซึ่งหมายถึงการที่ไคลเอนต์ส่งชื่อเส้นทางการเข้าถึงข้อมูลไปยังเซิร์ฟเวอร์ และเซิร์ฟเวอร์ก็จะส่งหมายเลขพอร์ตกลับมาให้ยังไคลเอนต์



รูปที่ 4-4 การโต้ตอบกันเมื่อระบบไฟล์ระยะไกลทำการเมาส์

จากรูปที่ 4-4 มีขั้นตอนการทำงานดังนี้

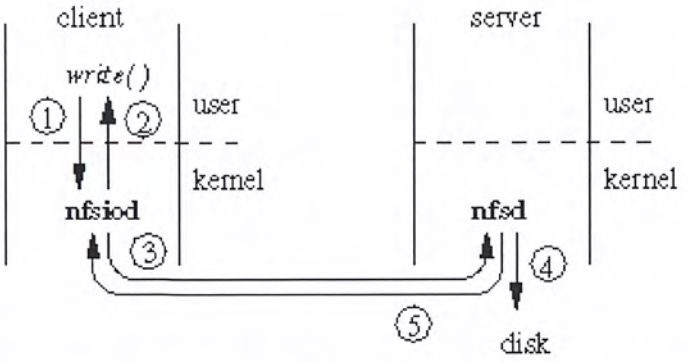
1. ไคลเอนต์ส่งเมสเซส (message) ไปยังพอร์ตที่เซิร์ฟเวอร์ซึ่งเรียกว่าพอร์ตแมป โดยไคลเอนจะร้องขอหมายเลขพอร์ตของไคลเอนต์นั้นจากเซิร์ฟเวอร์

2. เซิร์ฟเวอร์ตอบสนองและส่งหมายเลขพอร์ตกลับไปให้ไคลเอนต์

3. ไคลเอนต์ทำร้องขอการเมาส์ไปที่เซิร์ฟเวอร์โดยใช้ชื่อเส้นทางการเข้าถึงข้อมูล (Pathname) ที่ต้องการที่เมาต์

4. เซิร์ฟเวอร์เรียกขอไฟล์แฮนด์เดิล (File-handle) ที่ไคลเอนต์ต้องการเมาต์ ถ้าการร้องขอสำเร็จเซิร์ฟเวอร์ก็ตอบกลับไปยังไคลเอนต์ว่าการร้องขอเชื่อมต่อสำเร็จแล้ว

4.3.2 การเข้าไปใช้ไฟล์ในเซิร์ฟเวอร์หลังจากการเมาส์เสร็จเรียบร้อยแล้ว ไคลเอนต์สามารถอ่านไฟล์จากเซิร์ฟเวอร์ได้โดยส่งคำสั่งอาร์พีซีไปยังเซิร์ฟเวอร์แล้วจึงเริ่มกระบวนการอ่านไฟล์ โดยไคลเอนต์จะดูที่แคชก่อนถ้ามีไฟล์ที่ต้อง การอยู่แล้วก็ทำการอ่านจากแคชได้เลยโดยต้องอ่านข้อมูลผ่านระบบเครือข่าย แต่ถ้าไม่มีไฟล์ที่ต้องการในแคชไคลเอนต์ต้องอ่านไฟล์จากเซิร์ฟเวอร์โดยผ่านระบบเครือข่าย การโต้ตอบระหว่างไคลเอนต์เซิร์ฟเวอร์ แสดงในรูปที่ 4-5



รูปที่ 4-5 โต้ตอบระหว่างไคลเอนต์กับเซิร์ฟเวอร์ในการใช้งานส่วนของอินพุทเอาท์พุท

จากรูปที่ 4-5 มีขั้นตอนการทำงานดังนี้

1. เมื่อไคลเอนต์ต้องการเขียนข้อมูลที่เซิร์ฟเวอร์จึงส่งคำสั่งเขียนในรูปแบบอาร์พีซี คือ write( ) ผ่านไปยังบัฟเฟอร์ของเคอร์แนล

2. ข้อมูลที่ต้องการเขียนถูกส่งมารอที่บัฟเฟอร์ของไคลเอนต์

3. เดมอน nfsiod ซึ่งรันอยู่ที่เคอร์เนลนำข้อมูลที่พักอยู่ในบัฟเฟอร์ส่งไปยังเซิร์ฟเวอร์

4. ไคลเอนต์ส่งคำร้องร้องขอการเขียนไปที่เซิร์ฟเวอร์แล้วเดมอน nfsd ซึ่งทำงานอยู่บนเคอร์แนลของเซิร์ฟเวอร์รับคำร้องและเขียนข้อมูลลงสู่หน่วยความจำ

5. ภายหลังการเขียนข้อมูลลงหน่วยความจำเสร็จแล้ว เดมอน nfsd จะส่งสัญญาณตอบรับ (acknowledgment) กลับไปบอกไคลเอนว่าเขียนเสร็จแล้ว เป็นอันสิ้นสุดการติดต่อ

4.4 การเตรียมการก่อนการใช้งานเอ็นเอฟเอส

ก่อนการใช้งานเอ็นเอฟเอสเพื่อการใช้ไดเร็กทรี่ร่วมกันในระบบเครือข่าย ต้องตรวจสอบส่วนประกอบต่างดังนี้

4.4.1 ตรวจสอบว่าเคอร์แนล (kernel) ของระบบนั้นสนับสนุนการใช้งานเอ็นเอฟเอสหรือไม่ โดยมีตัวอย่างการตรวจสอบดังนี้

```
#cat /proc/filesystem
minix
        ext2
        msdos
nodev   proc
#
```

จากผลที่แสดงทางหน้าจอเห็นได้ว่าระบบยังไม่สนับสนุนเอ็นเอฟเอส ต้องติดตั้งโมดูลของเอ็นเอฟเอสโดยใช้คำสั่งดังนี้

```
# insmod nfs                          ; install NFS
# cat /proc/filesystem
minix
       ext2
       msdos
nodev  proc
nodev nfs
#
```

ภายหลังการติดตั้ง เห็นได้ว่ามีข้อความ nodev nfs แสดงว่าเคอร์แนลของระบบนั้นสนับสนุนเอ็นเอฟเอสแล้ว แต่ถ้าเคอร์เนลยังไม่สนับสนุนเอ็นเอฟเอสอาจต้องติดตั้งเคอร์เนลใหม่

4.4.2. ตรวจสอบไฟล์ต่างๆ ต่อไปนี้ว่ามีครบถ้วนหรือไม่

```
# cd  /usr/sbin/
# ls -al rpc*
rpc.nfsd
rpc.mountd
rpc.portmap
#
```

เนื่องจากว่าเอ็นเอฟเอสใช้อาร์พีซีในการรับส่งข้อมูล จึงใช้ rpc.portmap ในการรับส่งข้อมูล

4.4.3. ทำการรันเดมอนต่างๆ ดังนี้ portmap, mountd, nfsd และทำการตรวจสอบการทำงานของเดมอนว่าครบตามที่ต้องการใช้งานหรือไม่

```
# /etc/rc.d/init.d/portmap start          ; Start Portmap
# /etc/rc.d/init.d/nfs start              ; Start NFS deamon
# rpcinfo -p
program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp    759  mountd
    100005    1   tcp    761  mountd
    100005    2   udp    764  mountd
    100005    2   tcp    766  mountd
    100005    3   udp    769  mountd
    100005    3   tcp    771  mountd
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
#
```

ภายหลังการเริ่มรันเดมอน สังเกตได้ว่า portmap, mountd และ nfsd ได้สตาร์ทเรียบร้อยแล้ว จากนี้ก็ใช้เอ็นเอฟเอสในการแชร์ได้

4.5 การแชร์และเมาต์ไดเร็กทอรี

4.5.1 การแชร์ไดเร็กทอรีด้วยเอ็นเอฟเอส คือ การที่เซิร์ฟเวอร์ยอมให้ไคลเอนต์เข้ามาใช้งานไฟล์และไดเร็กทอรีที่เซิร์ฟเวอร์เปิดแชร์ไว้ และก่อนการเข้าใช้งานนั้นต้องมีการพิสูจน์สิทธิ์ว่าไคลเอนต์มีสิทธิ์ในการเข้าใช้งานหรือไม่ โดยเซิร์ฟเวอร์ตรวจสอบจากหมายเลขไอพีของไคลเอนต์

การตรวจสอบไดเร็กทอรีที่แชร์และไคลเอนต์ที่สามารถเข้ามาใช้งานได้นั้น ดูได้จากไฟล์ /etc/exports โดยรูปแบบข้อมูลของไฟล์ exports เป็นดังนี้

```
</ไดเร็กทอรีที่แชร์>  <ไคลเอนต์ที่สามารถเข้าใช้งานได้ (Options) >
```

ตัวอย่างไฟล์ export

```
# cat etc/exports
/15share 161.246.10.21(ro) 161.246.5.16(rw)
/usr/local 161.246.5.16(rw)
#
```

จากผลที่ได้ หมายถึง เซิร์ฟเวอร์แชร์ไดเร็กทอรี /home โดยยอมให้ไคลเอนต์ซึ่งมีหมายเลขไอพี 161.246.10.21 อ่านได้อย่างเดียว และได้แชร์ไดเร็กทอรี /usr/local โดยยอมให้ไคลเอนต์ซึ่งมีหมายเลขไอพี 161.246.5.16 เข้ามาอ่านและเขียนได้

การแชร์ไดเร็กทอรีโดยเอ็นเอฟเอส ทำได้ 2 แบบ คือ

- แก้ไฟล์ exports โดยตรง
- ใช้คำสั่ง export ดังรูปแบบและตัวอย่างต่อไปนี้

```
# export  /ไดเร็กทอรีที่เปิดแชร์  ไอพีของไคลเอนต์ที่ยอมให้เข้าใช้งาน  (option)
```

ตัวอย่างการใช้คำสั่ง export

```
# export  /home 161.246.5.15(ro) 161.246.5.16(rw)
```

จากตัวอย่าง คือ การแชร์ไดเร็กทอรี /home โดยยอมให้ไคลเอนต์ซึ่งมีหมายเลขไอพี 161.246.5.15 อ่านได้อย่างเดียวและยอมให้ไคลเอนต์ซึ่งมีหมายเลขไอพี 161.246.5.16 อ่านเขียนได้ ส่วนอ็อปชั่นต่างๆ ในการแชร์แบบเอ็นเอฟเอสนั้น สามารถดูรายละเอียดเพิ่มเติมได้จากคู่มือ (manual) ของคำสั่ง export

ภายหลังการแชร์ต้องทำการสตาร์ทเดมอนใหม่ทุกครั้ง ไม่เช่นนั้นระบบจะยังไม่ทำการแชร์เด็ดขาด การสตาร์ทเดมอนใหม่ทำได้ดังนี้

```
# /etc/rc.d/init.d/portmap stop
# /etc/rc.d/init.d/nfs stop
# /etc/rc.d/init.d/portmap start
# /etc/rc.d/init.d/nfs start
```

หรือ

```
# killall -HUP rpc.nfsd rpc.mountd
```

4.5.2 การยกเลิกการแชร์ ต้องแก้ไขไฟล์ /etc/exports โดยลบชื่อไดเร็กทอรีและหมายเลขไอพีของไคลเอนต์ หลังการแก้ไขไฟล์ต้องสตาร์ทเดมอนใหม่ทุกครั้ง ไม่เช่นนั้นระบบจะไม่ยกเลิกการแชร์

4.5.3 การเข้าใช้งานไดเร็กทอรี ก่อนที่ไคลเอนต์จะเมาต์เพื่อใช้งานไดเร็กทอรีได้นั้นต้องได้รับสิทธิ์จากเซิร์ฟเวอร์ก่อน และถ้าได้รับสิทธิ์จากเซิร์ฟเวอร์แล้ว ก็สามารถเข้าเมาต์ไดเร็กทอรีจากเซิร์ฟเวอร์เพื่อใช้งานได้ คำสั่งในการเมาต์มีรูปแบบดังนี้

```
# mount –t nfs ชื่อเซิร์ฟเวอร์:/ไดเร็กทอรีที่แชร์ /เมาต์พอยน์
```

ตัวอย่าง การเมาต์ไดเร็กทอรี /home บนเครื่อง 161.246.5.16 ไปไว้ที่ /mnt/mount1

```
# mount –t nfs 161.246.5.16:/home /mnt/mount1
```

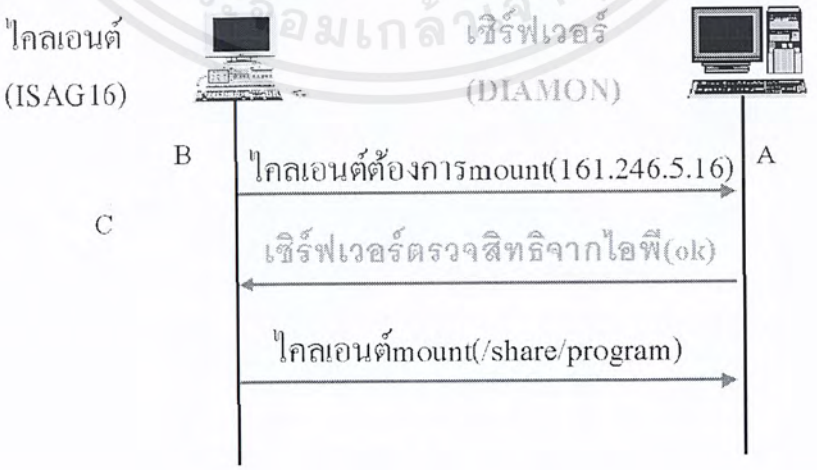
4.5.4 การยกเลิกการเมาต์ ทำได้โดยใช้คำสั่ง umount แล้วตามด้วยเมาต์พอยน์ที่ต้องการยกเลิกการเมาต์ ดังนี้

```
# umount /mnt/mount1
```

จากตัวอย่าง คือ การยกเลิกการเมาต์ไดเร็กทอรี /mnt/mount1

4.6 ปัญหาและความปลอดภัยในระบบยูนิกซ์

การพิสูจน์สิทธิ์ในเอ็นเอฟเอส ตรวจสอบจากหมายเลขไอพีของไคลเอนต์ว่ามีสิทธิ์เข้าใช้ไดเร็กทอรีที่แชร์ไว้หรือไม่ โดยไม่มีรหัสผ่านในการเข้าใช้จึงทำให้ผู้ใช้ที่มีรายชื่ออยู่ในบัญชีรายชื่อผู้ใช้งานในเครื่องนั้นทุกคนมีสิทธิ์เข้าใช้ไดเร็กทอรีนั้นได้ และอาจทำให้ผู้ใช้อื่นซึ่งไม่มีส่วนเกี่ยวข้องเข้าไปทำความเสียหายแก่ข้อมูลได้



รูปที่ 4-6 การพิสูจน์สิทธิ์โดยใช้ไอพี

จากรูปที่ 4-6 สมมุติให้ผู้ใช้ A แชร์ไดเร็กทอรีไว้และต้องการใช้งานร่วมกับผู้ใช้ B เมื่อทำงานเสร็จแล้วผู้ใช้ A อาจลืมยกเลิกการแชร์ ทำให้ผู้ใช้ C ซึ่งไม่มีส่วนเกี่ยวข้อง เข้ามาใช้งานไดเร็กทอรีนี้ได้ โดยผู้ใช้ C อาจเข้าไปทำลายหรือลบข้อมูลในไดเร็กทอรีนั้น ผู้ใช้จึงควรยกเลิกการแชร์และการเมาต์ทุกครั้งเมื่อเลิกใช้งานเพื่อป้องกันข้อมูลเสียหายได้

การแชร์ไดเร็กทอรีโดยใช้โพรโตคอลเอ็นเอฟเอสนั้น มีความยุ่งยากในการใช้งานมาก ดังนั้นผู้ใช้ควรมีความรู้ในการใช้งานยูนิกซ์มาก่อน

บทที่ 5
การเขียนโปรแกรมภาษาซีติดต่อระบบเครือข่าย

5.1 การเขียนโปรแกรมภาษาซีติดต่อระบบเครือข่าย

ในการเขียนโปรแกรมใดๆ ต้องศึกษาแอปพลิเคชั่นโปรแกรมอินเตอร์เฟส (Application Program Interface: API) ของโปรแกรมนั้น ในการเขียนโปรแกรมในระบบเครือข่ายก็เช่นกัน การเขียนโปรแกรมติดต่อกับระบบเครือข่ายโดยภาษาซีใช้เอพีไอที่เรียกว่า ซ็อกเก็ต (Socket API)

สาเหตุที่เรียกว่าซ็อกเก็ตเนื่องจาก ก่อนทำการติดต่อระหว่างเครื่องนั้น ต้องสร้างการเชื่อมต่อ (Connection) ขึ้นมาก่อน ซึ่งเปรียบเสมือนการโยงสายสัญญาณระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องเพื่อส่งข้อมูล โดยในแต่ละการเชื่อมต่อใช้งานได้ลักษณะเดียวในเครื่องคอมพิวเตอร์สองเครื่องเท่านั้น หากต้องการติดต่อระหว่างเครื่องคอมพิวเตอร์หลายๆ เครื่อง ต้องใช้การเชื่อมต่อหลายการเชื่อมต่อด้วยกัน หรือแม้กระทั่งในคอมพิวเตอร์สองเครื่องที่ต้องการติดต่อในหลายลักษณะงาน ก็ต้องมีการเชื่อมต่อหลายการเชื่อมต่อเช่นกัน ดังนั้นในคอมพิวเตอร์ 1 เครื่อง อาจมีการใช้งานหลายการเชื่อมต่อในเวลาเดียวกันก็ได้ ซึ่งจะเรียกแต่ละการเชื่อมต่อนี้ว่า ซ็อกเก็ต

การใช้งานซ็อกเก็ตนั้น ต้องสร้างซ็อกเก็ตขึ้นมาก่อน โดยในหนึ่งซ็อกเก็ตมีเพียงหน้าที่เดียวในหนึ่งการเชื่อมต่อ หลังจากการสร้างซ็อกเก็ตสำเร็จแล้ว ก็เสมือนมีท่อที่มองไม่เห็นเชื่อมโยงระหว่างเครื่องสองเครื่องอยู่

หลังจากการสร้างซ็อกเก็ตสำเร็จแล้ว เรานำซ็อกเก็ตที่สร้างขึ้นไปใช้งาน โดยสามารถใช้งานได้ทั้งไคลเอนต์และเซิร์ฟเวอร์ แต่ละฝั่งมีลักษณะการทำงานดังนี้

- ฝั่งเซิร์ฟเวอร์มีขั้นตอนการทำงานคือ Socket -> Bind -> Listen จากนั้นจะรอการติดต่อจากฝั่งไคลเอนต์ เมื่อติดต่อกันได้แล้วจึงเรียกใช้ ฟังก์ชัน Accept ต่อไป

  Socket คือ สร้างซ็อกเก็ต

  Bind คือ นำซ็อกเก็ตที่สร้างมากำหนดเข้ากับแอดเดรสจริง

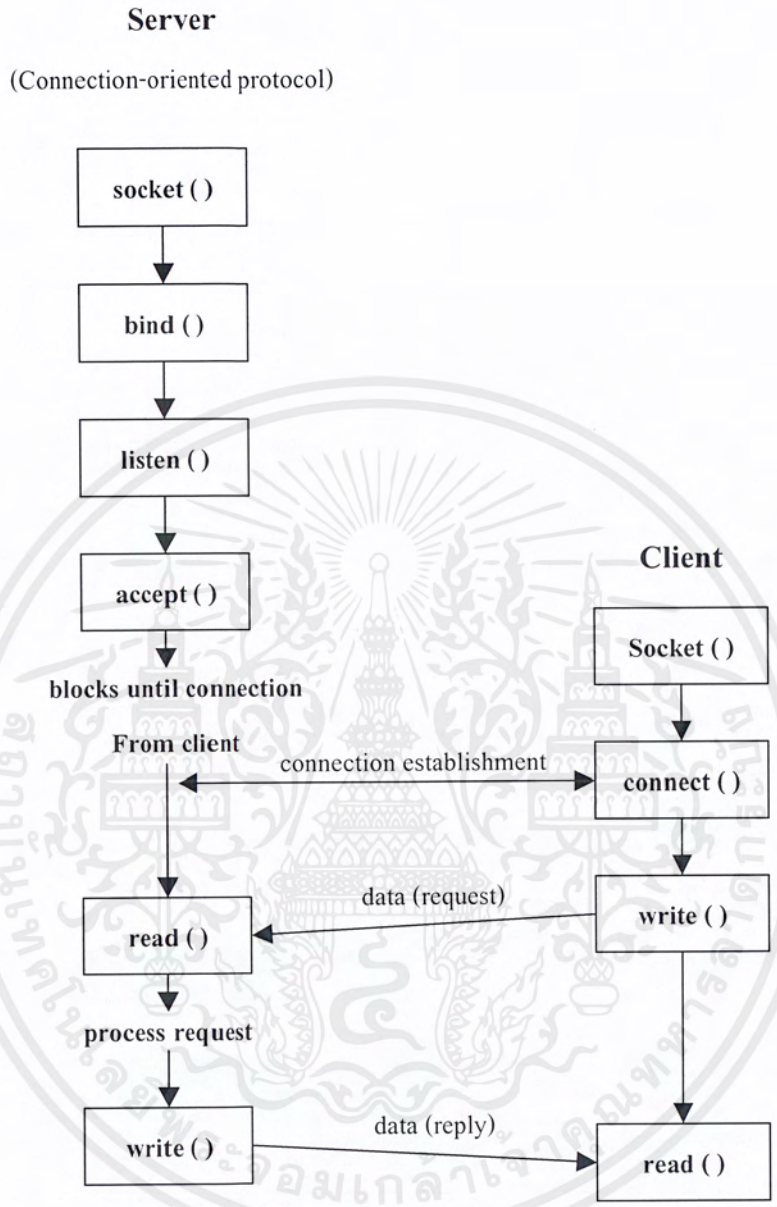
  Listen คือ ให้ซ็อกเก็ตรอรับการติดต่อจากโปรแกรมอื่นๆ

  Accept คือ ตอบรับการติดต่อ

- ฝั่งไคลเอนต์มีขั้นตอนการทำงานคือ Socket -> Connect หลังจากติดต่อสำเร็จก็เริ่มรับส่งข้อมูลกันได้

  Socket คือ สร้างซ็อกเก็ต

  Connect คือ ร้องขอทำการติดต่อ



รูปที่ 5-1 แผนผังแสดงการทำงานของโปรแกรมที่ใช้งานซ็อกเก็ต

จากรูปที่ 5-1 คือ ตัวอย่างการทำงานของโปรแกรมที่เรียกใช้งานซ็อกเก็ตเพื่อเขียนโปรแกรมติดต่อกับระบบเครือข่าย โดยจากตัวอย่างนี้เป็นการเขียนโปรแกรมที่เรียกใช้โพรโตคอลแบบต้องสร้างช่องทางการสื่อสารก่อน (Connection-oriented protocol) ซึ่งเห็นได้ว่าที่เซิร์ฟเวอร์ต้องรอการติดต่อจากเครื่องไคลเอนต์จนสร้างการติดต่อได้สำเร็จ โปรแกรมจึงทำงานต่อไปได้

5.2 การเขียนโปรแกรมภาษาซีติดต่อกับระบบเครือข่ายของวินโดวส์โดยใช้คำสั่งเอสเอ็มบี

การเขียนโปรแกรมภาษาซีติดต่อกันระบบเครือข่ายของวินโดวส์สามารถทำได้หลายวิธี เช่น เขียนโปรแกรมโดยเรียกใช้งานซ็อกเก็ตเอพีไอ โดยอาจเขียนโปรแกรมบนเบิร์กเลย์ซ็อกเก็ต (Berkeley Sockets) หรือบนวินโดวส์ซ็อกเก็ต (Winsock) และทำการจัดรูปแบบของแพ็กเกจตามแบบของเอสเอ็มบี แล้วจึงติดต่อผ่านพอร์ต 137, 138 หรือ 139 ตามประเภทของแพ็กเกจนั้น

5.2.1 โครงสร้างแพ็กเกจของเอสเอ็มบี

การจัดข้อมูลตามรูปแบบของเอสเอ็มบี กระทำได้โดยจัดรูปแบบข้อมูลเป็นโครงสร้างดังรูปที่ 5-2 โดยในส่วนหัวของข้อมูลต้องเริ่มด้วยค่า "0xFF" เพื่อเป็นการประกาศว่าเป็นข้อมูลของเอสเอ็มบี ตามด้วยรหัสคำสั่งของเอสเอ็มบี และมีฟิลด์ต่างๆ เกี่ยวกับบิตสถานะ, การแสดงข้อผิดพลาด และรหัสทรัพยากรที่ใช้ในการติดต่อ ตามรูปที่ 5-2

```
typedef struct
{
    UCHAR Protocol[4];                  // Contains 0xFF,'SMB'
    UCHAR Command;                      // Command code
    union
    {
        struct
        {
            UCHAR ErrorClass;           // Error class
            UCHAR Reserved;             // Reserved for future use
            USHORT Error;               // Error code
        } DosError;
        ULONG NtStatus;                 // NT-style 32-bit error code
    } Status;
    UCHAR Flags;                        // Flags
    USHORT Flags2;                      // More flags
    union
    {
        USHORT Pad[6];                  // Ensure this section is 12 bytes
        struct
        {
            USHORT PidHigh;             // High part of PID (NT Create And X)
            struct
            {
                ULONG  HdrReserved;     // Not used
                USHORT Sid;             // Session ID
                USHORT SequenceNumber;  // Sequence number
             } Connectionless;          // IPX
        }
    };
    USHORT Tid;                         // Tree identifier
    USHORT Pid;                         // Caller's process id
    USHORT Uid;                         // Unauthenticated user id
    USHORT Mid;                         // multiplex id
    UCHAR  WordCount;                   // Count of parameter words
    USHORT ParameterWords[ WordCount ]; // The parameter words
    USHORT ByteCount;                   // Count of bytes
    UCHAR  Buffer[ ByteCount ];         // The bytes
} SMB_HEADER;
```

รูปที่ 5-2 โครงสร้างแพ็กเกจของเอสเอ็มบี

โครงสร้างแพ็กเกจของเอสเอ็มบีทั่วไปมีลักษณะที่เหมือนกันตั้งแต่ฟิลด์ *ParameterWords* ขึ้นไป มีเพียงหมายเลขและการแปรความหมายใน *ParameterWords* และ *Buffer* ที่แตกต่างกัน ในทุกฟิลด์ที่สงวนไว้และไม่มีการใช้งานในแพ็กเกจของเอสเอ็มบีมีค่า "0" แต่ละฟิลด์ในแพ็กเกจของเอสเอ็มบีมีความหมายดังนี้

Command : รหัสคำสั่งของเอสเอ็มบีที่ใช้สั่งงานเซิร์ฟเวอร์หรือเป็นการตอบรับคำร้องขอจากเซิร์ฟเวอร์ โดยรหัสคำสั่งเอสเอ็มบีสามารถดูได้จากภาคผนวก ก.

Status.DosError.ErrorClass และ *Status.DosError.Error* : เป็นฟิลด์แสดงข้อผิดพลาดซึ่งถูกเซ็ตโดยเซิร์ฟเวอร์ในกรณีที่ไม่สามารถทำงานตามคำสั่งที่ได้รับมา ถ้าไคลเอนต์สามารถรองรับค่าผิดพลาดแบบ 32 บิตได้ ค่าผิดพลาดจะถูกส่งมาที่ฟิลด์ *Status.NtStatus* แทน ในขณะที่มีการส่งค่าผิดพลาดกลับมาเซิร์ฟเวอร์จะส่งมาเพียงส่วนที่แสดงข้อผิดพลาดเท่านั้นไม่มีการส่งข้อมูลมาด้วย

Flags and *Flags2* : เป็นบิตแสดงสถานะ ขึ้นอยู่กับการเจรจาโดยโพรโตคอลก่อนการติดต่อ

PidHigh : ใช้ในคำสั่ง *NtCreateAndX*

Connectionless. Sid, และ *Connectionless.SequenceNumber* : ใช้โดยไคลเอนต์ในกรณีที่ต้องการติดต่อกับเซิร์ฟเวอร์โดยใช้โพลโตคอลแบบไม่ต้องสร้างช่องทางการสื่อสารก่อน (datagram oriented protocol) เช่น ไอพีเอ็กซ์หรือยูดีพี (IPX or UDP)

StreamProtocol.SMBLength : ใช้โดยไคลเอนต์ในกรณีที่ต้องการติดต่อกับเซิร์ฟเวอร์โดยใช้โพลโตคอลแบบต้องสร้างช่องทางการสื่อสารก่อน (Connection oriented protocol) เช่น ทีซีพี (TCP) โดยในฟิลด์นี้ใช้เก็บค่าความยาวของแพ็กเกจโดยเริ่มตั้งแต่ฟิลด์เริ่มต้น "0xFF" ไปจนฟิลด์สุดท้าย

Tid : เป็นรหัสหมายเลขของไดเร็กทอรีย่อยหรือแผนภูมิต้นไม้ (tree) บนเครื่องเซิร์ฟเวอร์ที่เครื่องไคลเอนต์ต้องการเข้าถึง ถ้าไม่สามารถระบุได้ให้ตั้งค่าเป็น "0xFFFF"

Pid : คือหมายเลขของกระบวนการทำงาน (Process id) จากเครื่องไคลเอนต์ที่เรียกใช้งาน ซึ่งหมายเลขนี้ไม่มีการซ้ำบนเครื่องไคลเอนต์เดียวกัน

Mid : สงวนไว้ใช้ในกรณีที่มีการผสมหลายๆ ข้อมูล (Multiplex) บนวงจรเสมือน (Virtual Circuit) เดียวกัน

5.2.2 กระบวนการเขียนโปรแกรมติดต่อกับระบบเครือข่ายของวินโดวส์โดยใช้คำสั่งเอสเอ็มบี

กระบวนการในการเขียนโปรแกรมเพื่อติดต่อกับระบบเครือข่ายของวินโดวส์ ดูได้จากตารางที่ 5-1 ซึ่งเป็นตัวอย่างการอ่านไฟล์ข้อมูลจากเซิร์ฟเวอร์ โดยในคำสั่งแรกที่ใช้ในการติดต่อต้องเป็นคำสั่งที่ใช้ในการเจรจากับเซิร์ฟเวอร์เกี่ยวกับส่วนขยายของโพรโตคอลที่ไคลเอนต์สนับสนุนและข้อตกลงต่างๆ ในการสื่อสาร และต่อไปเป็นคำสั่งเกี่ยวกับการพิสูจน์ตนของเครื่องไคลเอนต์ ในส่วนการติดต่อกับทรัพยากรที่แชร์บนเครื่องเซิร์ฟเวอร์มีการอ้างถึงโดยใช้หมายเลขแทนตัวทรัพยากรนั้น เช่น หมายเลขแผนภูมิต้นไม้หรือหมายเลขไฟล์

คำสั่งจากไคลเอนต์	การตอบรับจากเซิร์ฟเวอร์
SMB_COM_NEGOTIATE	เป็นคำสั่งแรกที่ไคลเอนต์ต้องส่งให้เซิร์ฟเวอร์เสมอ โดยภายในแสดงรายการคำสั่งเอสเอ็มบีที่ไคลเอนต์สนับสนุนไปให้เซิร์ฟเวอร์ และเซิร์ฟเวอร์ตอบรับการใช้คำสั่งเอสเอ็มบีนั้น
SMB_COM_SESSION_SETUP_ANDX	แจ้งชื่อผู้ใช้เพื่อเป็นการพิสูจน์ตนกับเซิร์ฟเวอร์ และเซิร์ฟเวอร์ตอบโดยส่งหมายเลขผู้ใช้งานมาให้ที่ฟิลด์ *Uid*
SMB_COM_TREE_CONNECT	แจ้งชื่อของไดเร็กทอรีที่ไคลเอนต์ต้องการเข้าถึง และเซิร์ฟเวอร์ตอบโดยส่งหมายเลขแผนภูมิต้นไม้มาให้ที่ฟิลด์ *Tid*
SMB_COM_OPEN	แจ้งชื่อไฟล์โดยไฟล์นั้นต้องสัมพันธ์กับหมายเลขแผนภูมิต้นไม้ที่ต้องการติดต่อ และเซิร์ฟเวอร์ส่งหมายเลขไฟล์กลับมาให้
SMB_COM_READ	ไคลเอนต์ส่งหมายเลขแผนภูมิต้นไม้, หมายเลขไฟล์, จุดเริ่มต้นในไฟล์และจำนวนไบต์ที่ต้องการอ่าน และเซิร์ฟเวอร์ส่งข้อมูลในไฟล์นั้นกลับมา
SMB_COM_CLOSE	ไคลเอนต์ขอปิดไฟล์ที่อ้างถึงในฟิลด์ *Tid* และ *fid* และเซิร์ฟเวอร์ตอบกลับโดยรหัสที่แจ้งว่าการทำงานสำเร็จ
SMB_COM_TREE_DISCONNECT	ไคลเอนต์ขอหยุดการติดต่อกับทรัพยากรที่ระบุในฟิลด์ Tid

ตารางที่ 5-1 แสดงกระบวนการการส่งคำสั่งเอสเอ็มบีในการอ่านไฟล์จากเซิร์ฟเวอร์

5.3 แลนแมนเนจเจอร์

แลนแมนเนจเจอร์ คือ ส่วนขยายของเอสเอ็มบีซึ่งได้รับการพัฒนาขึ้นมาตั้งแต่ PC-NET 1.0 ที่จัดว่าเป็นแกนหลักของเอสเอ็มบี จนมาเป็นแลนแมนเนจเจอร์ที่ทำงานได้บนวินโดวส์ โดยในระบบปฏิบัติการวินโดวส์มีตัวจัดการระบบเครือข่ายของวินโดวส์หรือแลนแมนเนจเจอร์ซึ่งถูกสร้างอยู่ในรูปแบบดีแอลแอล (DLL: Dynamic Link Library) เพื่อให้โปรแกรมในวินโดวส์สามารถเรียกใช้งานได้

5.3.1 การเรียกใช้คำสั่งของแลนแมนเนจเจอร์

การเขียนโปรแกรมติดต่อกับระบบเครือข่ายของวินโดวส์อีกวิธีหนึ่ง คือ การเรียกใช้คำสั่งของแลนแมนเนจเจอร์ โดยคอมไพลเลอร์ในปัจจุบันส่วนมากมีไลบรารีเกี่ยวกับระบบเครือข่ายของวินโดวส์มาให้ เมื่อเรียกใช้งานคำสั่งแลนแมนเนจเจอร์เหล่านี้ โปรแกรมที่เขียนจะมีการเรียกใช้ไลบรารีดีแอลแอลและไดรเวอร์ของวินโดวส์โดยอัตโนมัติ โดยการทำงานของโปรแกรมจะมีการเปิดซ็อกเก็ตของทีซีพีไอพีและเรียกรับส่งข้อมูลผ่านพอร์ต 137, 138 และ 139 ให้เอง นอกจากนี้คอมไพลเลอร์ยังจัดรูปแบบแพ็กเกจของข้อมูลและคำสั่งตามรูปแบบของเอสเอ็มบีให้โดยอัตโนมัติ รวมถึงการถอดแพ็กเกจผลลัพธ์ที่ส่งกลับมาด้วย

ในบางครั้งการเรียกใช้คำสั่งของแลนแมนเนจเจอร์เพียงคำสั่งเดียวอาจเป็นการใช้คำสั่งของเอสเอ็มบีหลายคำสั่ง ซึ่งการเขียนโปรแกรมวิธีนี้สะดวกว่าการเขียนโปรแกรมแบบเปิดซ็อกเก็ตเอง แต่ก็มีข้อเสีย คือ คำสั่งของแลนแมนเนจเจอร์ที่มีในคอมไพลเลอร์นั้นอาจไม่สนับสนุนไลบรารีของวินโดวส์ซึ่งอาจทำให้โปรแกรมที่เขียนนั้นคอมไพล์ผ่านแต่ไม่สามารถทำงานได้

5.2.2 ไลบรารีของคำสั่งแลนแมนเนจเจอร์

คอมไพลเลอร์สำหรับเขียนโปรแกรมบนวินโดวส์ส่วนมากจะมีไลบรารีของคำสั่งแลนแมนเนจเจอร์มาให้ เช่น เดลไฟล์มีไลบรารีชื่อ “Windows” วิชวลซี (Visual C) และ บอร์แลนด์ซีพลัสพลัส (Borland C++) มีไลบรารีชื่อ “lm.h”

5.2.3 ไลบรารีชื่อ “lm.h”

คือ ไลบรารีในคอมไพลเลอร์ภาษาซีที่ทำงานบนวินโดวส์ โดยสามารถพบได้ทั้งบนวิชวลซีของไมโครซอฟท์และบอร์แลนด์ซีพลัสพลัสของบอร์แลนด์ การทำงานของไลบรารีนี้จะติดต่อกับไลบรารีของวินโดวส์ซึ่งทำหน้าที่เกี่ยวกับระบบเครือข่ายของวินโดวส์ เมื่อในโปรแกรมมีการเรียกใช้คำสั่งซึ่งอยู่ในไลบรารี “lm.h” คอมไพลเลอร์จะทำการสร้างลิงค์เชื่อมต่อกับไลบรารีดีแอลแอลของวินโดวส์เพื่อให้สามารถติดต่อกับระบบเครือข่ายของวินโดวส์ได้ คำสั่งในไลบรารี “lm.h” มีหลายคำสั่งโดยมีการแยกย่อยเป็นหลายหมวดหมู่เช่น คำสั่งเกี่ยวกับการแชร์, คำสั่งเกี่ยวกับเซิร์ฟเวอร์, คำสั่งเกี่ยวกับไคลเอนต์, คำสั่งเกี่ยวกับการปรับแต่งระบบและคำสั่งเกี่ยวกับการอ่านข้อผิดพลาด

การเรียกใช้คำสั่งในไลบรารี “lm.h” สามารถทำงานได้ดีบนวินโดวส์เอ็นที ส่วนการทำงานบนวินโดวส์ 95/98 ยังพบว่ามีข้อผิดพลาดค่อนข้างมาก โดยเมื่อคอมไพล์โปรแกรมแล้วไม่สามารถรันโปรแกรมได้และมีการแจ้งว่าหาไฟล์ไลบราลีดีแอลแอลของวินโดวส์ที่เกี่ยวกับระบบเครือข่ายไม่พบ คำสั่งของแลนแมนเนจเจอร์ที่ใช้ในการตรวจสอบการแชร์ทรัพยากรสามารถดูได้ในภาคผนวก ข.

5.4 การเขียนโปรแกรมติดต่อกับระบบเครือข่ายยูนิกซ์

5.4.1 ระดับการเขียนโปรแกรมติดต่อกับเครือข่ายยูนิกซ์โดยใช้อาร์พีซี

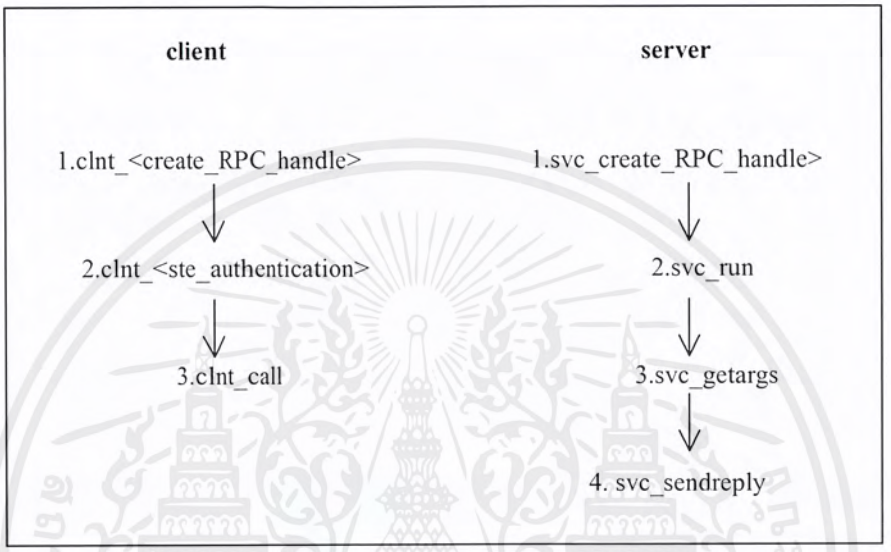
การเขียนโปรแกรมติดต่อระบบเครือข่ายแบบอาร์พีซีนั้น เหมือนกับการอ้างถึงไลบรารีฟังก์ชันของภาษาซีทั่วไป โดยอาร์พีซีสามารถแบ่งได้เป็นสามระดับ คือ ระดับบนสุด, ระดับกลางและระดับล่าง โดยมีรายละเอียด คือ

- ระดับบนสุด โดยในระดับนี้ผู้ใช้สามารถเรียกใช้ฟังก์ชันได้โดยตรงเหมือนการเรียกใช้ไลบารีฟังก์ชันของภาษาซีทั่วไป จึงง่ายต่อการเรียกใช้งานฟังก์ชัน

- ระดับกลาง คือการเรียกใช้ตัวแปรอาร์พีซีเจ็น (rpcgen) ที่ทำให้เกิดการสะตับ (stub) ของเซิร์ฟเวอร์และไคลเอนต์โดยอัตโนมัติ ข้อดีของการทำงานระดับนี้ คือ ผู้ใช้สามารถเขียนฟังก์ชันอาร์พีซีและฟังก์ชันหลักของไคลเอนต์ได้ โดยไม่ต้องทราบเอพีไอของอาร์พีซีระดับต่ำ และช่วยเพิ่มความปลอด

ภัยในการเขียนโปรแกรมและเกิดข้อผิดพลาดน้อย แต่ในอาร์พีซีระดับนี้ไม่สามารถจัดการกับหน่วยความจำแบบไดนามิกโดยใช้ฟังก์ชันของเอ็กซ์ดีอาร์ได้

- ระดับล่าง คือ การเรียกใช้เอพีไอของอาร์พีซีโดยตรง ข้อดีคือ สามารถจัดการและกระทำกับหน่วยความจำแบบไดนามิกได้โดยใช้ฟังก์ชันของเอ็กซ์ดีอาร์ได้ การเขียนโปรแกรมติดต่อกับเครือข่ายแบบอาร์พีซีระดับล่างนี้ต้องประกาศอินคลูดที่หัวของโปรแกรมก่อน (#include rpc/rpc.h) โดยลำดับในการเรียกใช้งานอาร์พีซีระหว่างเซิร์ฟเวอร์และไคลเอนต์เป็น ดังนี้



รูปที่ 5-3 การเรียกใช้งานอาร์พีซีระหว่างไคลเอนต์และเซิร์ฟเวอร์

จากรูปที่ 5-3 ที่ไคลเอนต์ (1) คือ การสร้างการติดต่อกับเซิร์ฟเวอร์สำหรับกำหนดโปรแกรมอาร์พีซีและเลขเวอร์ชั่นโปรแกรม (2) คือ การสร้างการพิสูจน์ตนที่ก่อนเข้าไปยังเซิร์ฟเวอร์ (3) คือ การเรียกใช้อาร์พีซีไปยังเซิร์ฟเวอร์ โดยที่เซิร์ฟเวอร์มีการทำงานตามหมายเลขโปรแกรมที่ไคลเอนต์ส่งไป

ทางด้านเซิร์ฟเวอร์ (1) คือ ตอบสนองไคลเอนต์ที่ร้องขอมา (2) คือ กระทำโปรแกรมที่ไคลเอนต์ร้องขอมา (3) และ (4) คือ ส่งผลที่ได้กลับไปยังไคลเอนต์ที่ร้องขอมา

5.4.2 ฟังก์ชันการเปลี่ยนชนิดข้อมูลแบบเอ็กซ์ดีอาร์

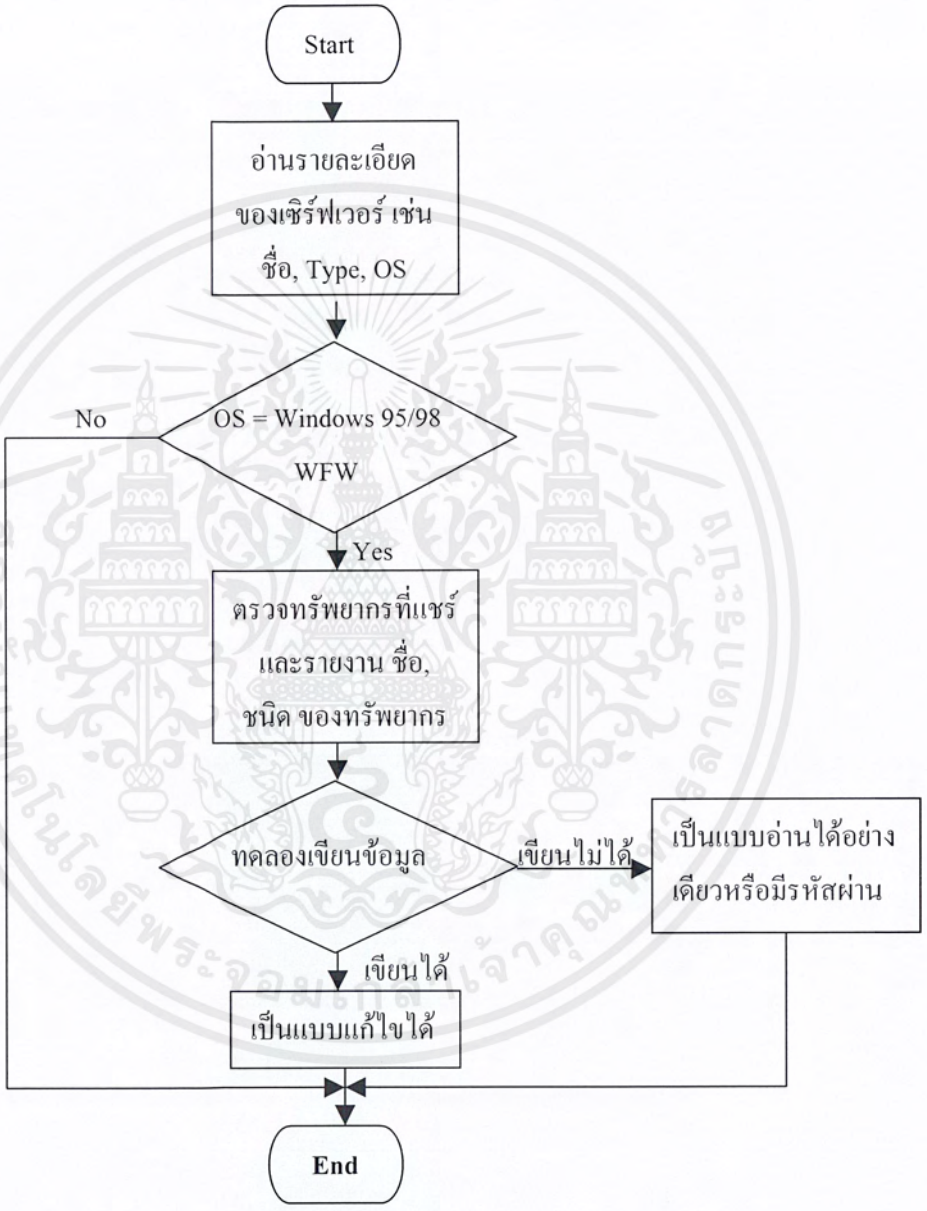
ก่อนการที่ส่งข้อมูลออกไปยังระบบเครือข่ายแบบใช้ฟังก์ชันอาร์พีซีนั้น ต้องแปลงให้อยู่ในรูปของเอ็กซ์ดีอาร์เสียก่อน โดยการแปลงข้อมูลให้อยู่ในรูปของเอ็กซ์ดีอาร์นั้น ช่วยให้เครื่องที่ต่างระบบกันสามารถติดต่อกันได้ รายละเอียดฟังก์ชันของเอ็กซ์ดีอาร์ต่างๆ นั้น มีดังนี้

ชนิดของข้อมูล	ฟังก์ชันเอ็กซ์ดีอาร์
int	xdr_ int
long	xdr_ long
short	xdr_ short
char	xdr_ char
u_int	xdr_ u_int
u_long	xdr_ u_long
u_short	xdr_ u_short
u_char	xdr_ u_char
float	xdr_ float
double	xdr_ double
enum	xdr_ enum
bool	xdr_ bool
string	xdr_ string
union	xdr_ union
opaque	xdr_ opaque

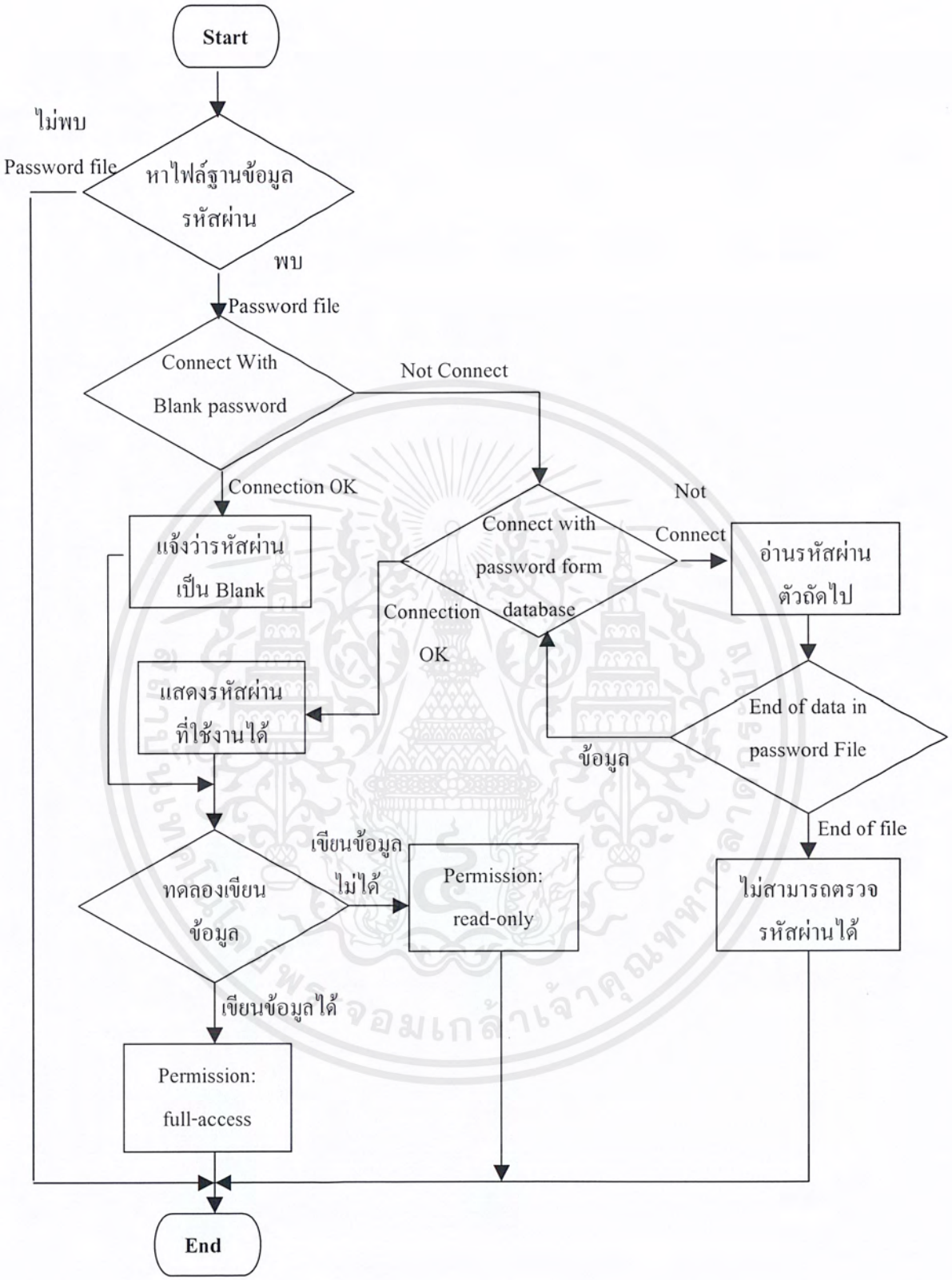
ตารางที่ 5-2 การแปลงข้อมูลให้อยู่ในรูปของเอ็กซ์ดีอาร์

บทที่ 6
การออกแบบและการสร้าง

6.1 การออกแบบโปรแกรมตรวจการแชร์บนระบบเครือข่ายของวินโดวส์



รูปที่ 6-1 แผนผังการทำงานของโปรแกรมตรวจการแชร์ทรัพยากร



รูปที่ 6-2 แผนผังการทำงานของโปรแกรมตรวจหารหัสผ่าน

6.2 การทำงานของโปรแกรมตรวจสอบการแชร์บนระบบเครือข่ายของวินโดวส์

6.2.1 การทำงานของโปรแกรมตรวจการแชร์ทรัพยากร (SMBScan)

หลักการทำงานของโปรแกรมตรวจการแชร์ทรัพยากร คือ การไปอ่านรายการแชร์ทรัพยากรที่เครื่องเซิร์ฟเวอร์และตรวจสอบระดับของสิทธิ์ในการเข้าใช้งานและรายงานผล โดยมีการทำงานดังนี้

- เมื่อเรียกใช้งานโปรแกรมตรวจทรัพยากรโดยมีรูปแบบคือ “SMBScan \\HostName” โปรแกรมจะทำการอ่านรายละเอียดของเครื่องต้องการตรวจสอบและแสดงผล โดยรายงานชื่อเครื่อง, ระบบปฏิบัติการที่ใช้, เวิร์กกรุป รายละเอียดและหมายเหตุอื่นๆ

- ตรวจสอบว่าเครื่องเซิร์ฟเวอร์ที่ต้องการสำรวจนั้น เป็นระบบปฏิบัติการที่ใช้รูปแบบการรักษาความปลอดภัยแบบแบ่งตามระดับการแชร์หรือไม่ (วินโดวส์ 95/98 หรือวินโดวส์ฟอร์เวิร์กกรุป) เพื่อให้เห็นทรัพยากรได้โดยไม่ต้องล็อกอิน

- ตรวจสอบทรัพยากรที่แชร์ถ้าเป็นเครื่องพิมพ์ให้รายงานผลได้เลย ถ้าเป็นไดเรกทอรีต้องทดลองเขียนข้อมูลลงไปเพื่อทดสอบว่าเป็นแบบ อ่านเขียนได้หรือเป็นแบบอ่านได้อย่างเดียว

6.2.2 การทำงานของโปรแกรมตรวจรหัสผ่าน (PWScan)

หลักการทำงานของโปรแกรมตรวจสอบรหัสผ่าน คือ อ่านรหัสผ่านจากไฟล์ฐานข้อมูลและทดลองส่งไปยังเครื่องเซิร์ฟเวอร์ เมื่อติดต่อสำเร็จก็ตรวจสอบระดับของสิทธิ์การเข้าใช้งานและรายงานผล โดยมีการทำงานดังนี้

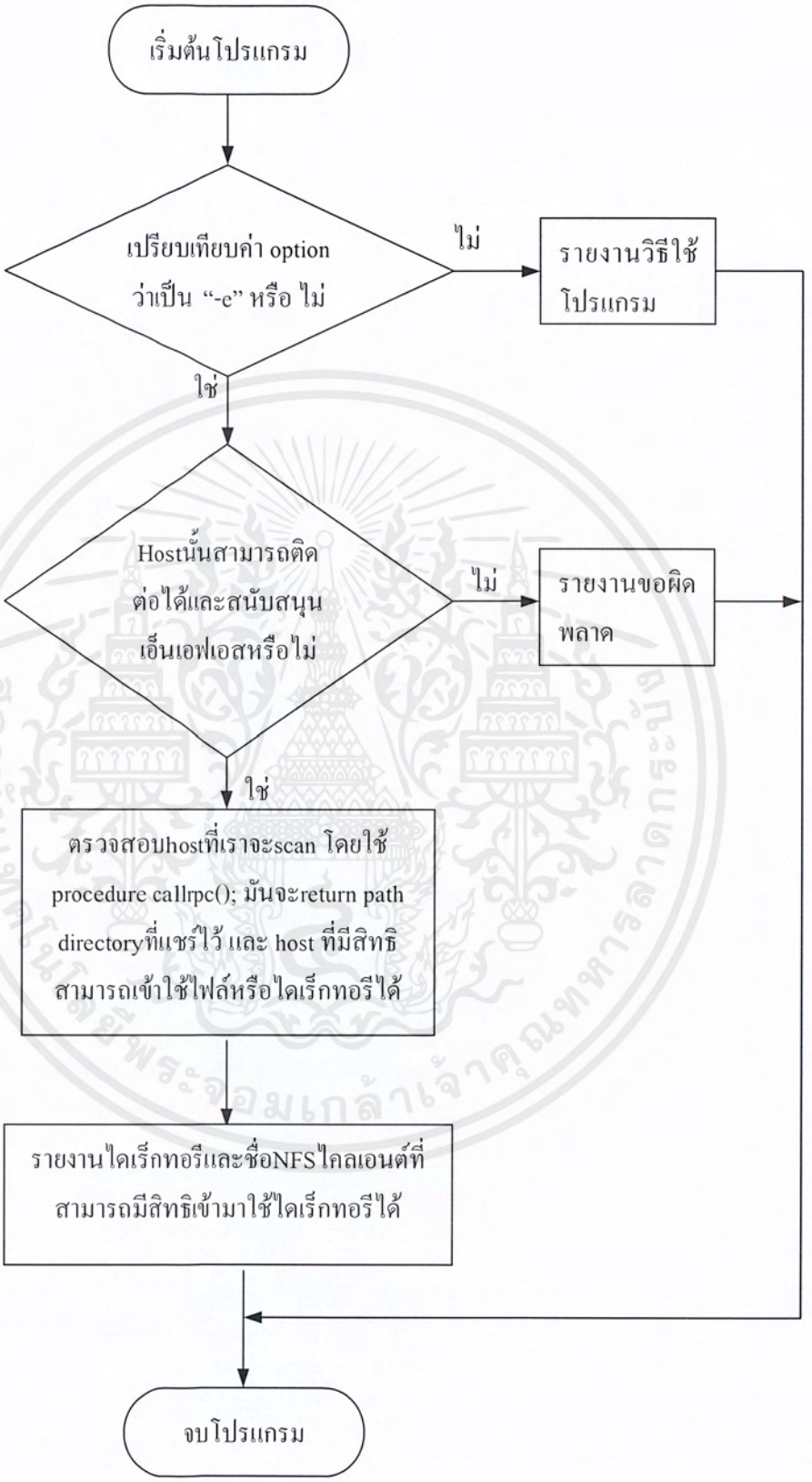
- ตรวจสอบไฟล์รหัสผ่านถ้าไม่พบให้แจ้งข้อผิดพลาดและออกจากโปรแกรม

- ทดลองติดต่อโดยใช้รหัสผ่านเป็นช่องว่าง (Blank) และรหัสผ่านจากในไฟล์ฐานข้อมูลจนกระทั่งพบรหัสผ่านที่ใช้งานได้หรือหมดข้อมูลในไฟล์รหัสผ่าน

- ถ้าพบรหัสผ่านก็ให้แสดงรหัสผ่านนั้นและทดลองเขียนข้อมูลเพื่อตรวจสอบว่าเป็นการแชร์แบบอ่านได้อย่างเดียวหรืออ่านเขียนได้

- ถ้าไม่พบรหัสผ่านให้แสดงข้อผิดพลาด

6.3 โปรแกรมตรวจการใช้ทรัพยากรร่วมกันบนระบบเครือข่ายยูนิกซ์



รูปที่ 6.3 แผนผังการทำงานของโปรแกรมตรวจสอบการแชร์ของยูนิกซ์

การทำงานของโปรแกรมตรวจสอบการแชร์ของยูนิกซ์

- เริ่มโปรแกรมโดยการเปรียบเทียบค่าที่ได้รับเข้ามาโดยจะนำค่าที่เป็นส่วนของอ็อปชันมาเปรียบเทียบว่าทำงานอะไร เช่น -? คือการรายงานวิธีใช้โปรแกรม, -e คือการแสดงรายงานผลของการตรวจสอบการแชร์ทรัพยากร

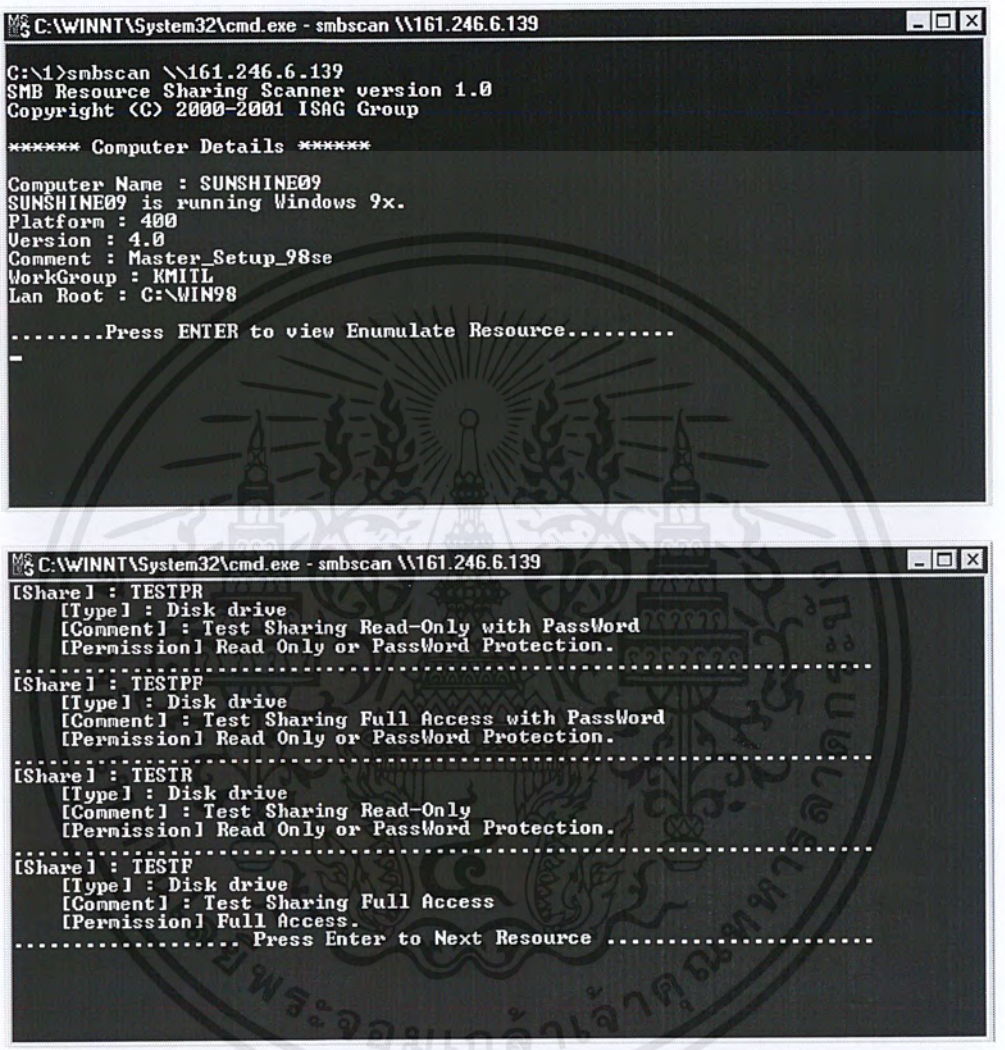
- ส่วนตรวจสอบการเปิดให้บริการเอ็นเอฟเอส ถ้าเปิดให้บริการก็ตรวจสอบการแชร์ทรัพยากร ถ้าเครื่องที่ตรวจสอบไม่เปิดให้บริการเอ็นเอฟเอส ก็รายงานข้อผิดพลาด

- ส่วนของการตรวจสอบการแชร์ทรัพยากร จะรับค่าไอพีเข้ามา และไปตรวจสอบการแชร์ที่เครื่องตามไอพีที่รับเข้ามานั้น โดยจะได้ผลลัพธ์เป็นรายงานการแชร์ทรัพยากรของเครื่องนั้น

- รายงานผลการตรวจสอบโดยมีรายละเอียด คือ ไดเรกทอรีที่แชร์และชื่อเครื่องที่ได้รับสิทธิ์เข้ามาใช้ไดเรกทอรีนี้ได้

บทที่ 7
ตัวอย่างการใช้งานและทดสอบ

7.1 ตัวอย่างการรันโปรแกรมตรวจสอบการแชร์บนระบบเครือข่ายของวินโดวส์



รูปที่ 7-1 ผลการรันโปรแกรมตรวจสอบการแชร์บนระบบเครือข่ายของวินโดวส์

7.1.1 ผลการรันโปรแกรม

จากรูปที่ 7-1 เมื่อรันโปรแกรม "SMBScan \\HostName" โปรแกรมจะตรวจและแสดงรายละเอียดเกี่ยวกับเครื่องที่เราตรวจโดยมีรายละเอียดดังนี้

- Computer Name ชื่อของเครื่องคอมพิวเตอร์ที่ใช้อ้างอิงในระบบเครือข่าย
- ระบบปฏิบัติการที่เครื่องคอมพิวเตอร์นั้นใช้อยู่
- Platform รหัสของแพลตฟอร์มเครื่องและระบบปฏิบัติการที่ใช้
- Version เวอร์ชันของโพรโตคอลเอสเอ็มบี

- Comment หมายเหตุของเครื่อง
- Workgroup เวิร์กกรุปของเครื่อง
- Lan Root ไดเรกทอรีหลักของระบบเครือข่าย

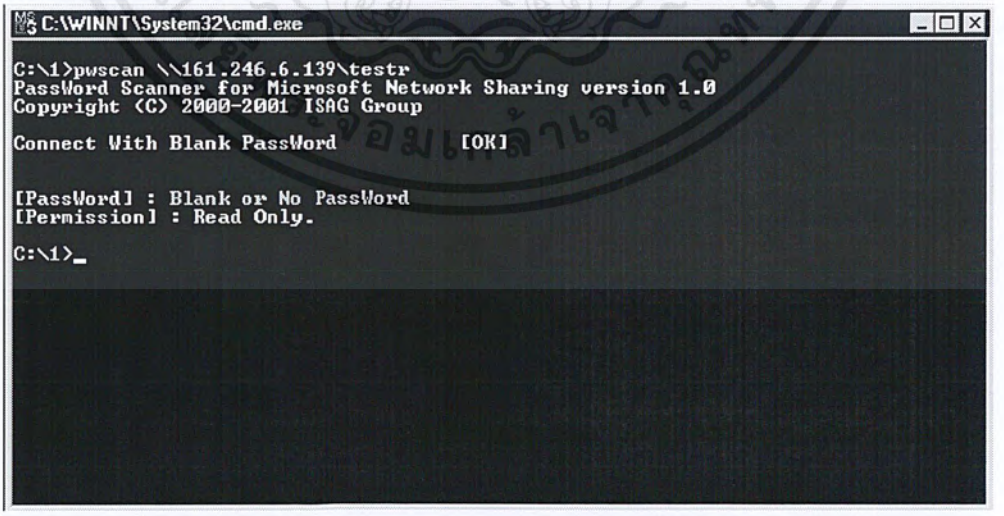
หน้าถัดไปเป็นหน้าจอแสดงทรัพยากรที่เปิดแชร์ไว้ในเครื่องคอมพิวเตอร์เครื่องนั้นโดยมีรายละเอียดดังนี้

- Share ชื่อของทรัพยากรที่แชร์
- Type ชนิดของทรัพยากรที่แชร์ (ดิสก์หรือเครื่องพิมพ์)
- Comment หมายเหตุของทรัพยากร
- Permission สิทธิ์ในการเข้าใช้งานทรัพยากร

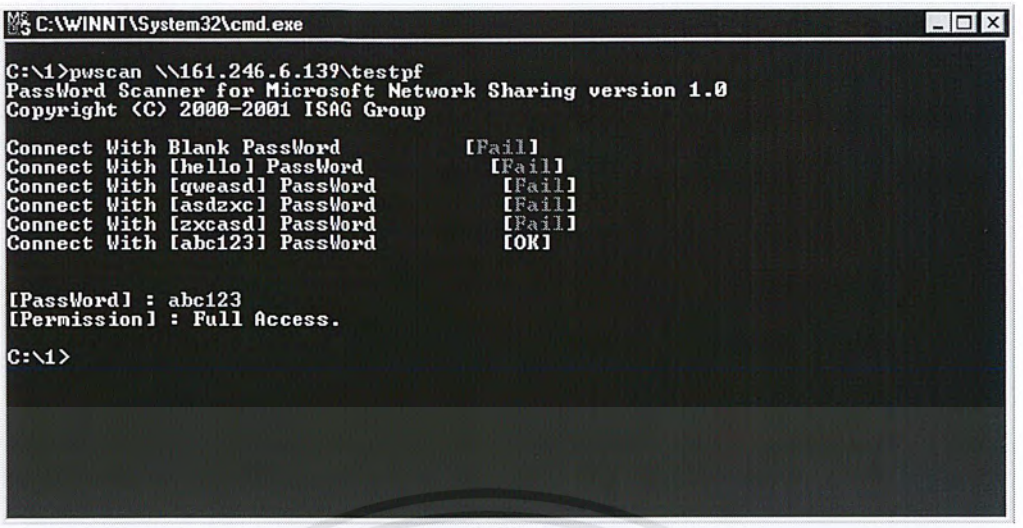
7.1.2 ข้อจำกัดของโปรแกรม

โปรแกรมที่ทำหน้าที่ตรวจการแบ่งใช้ทรัพยากรในระบบเครือข่ายของวินโดวส์นี้ ยังไม่สามารถทำงานได้กับระบบที่มีการรักษาความปลอดภัยแบ่งตามระดับผู้ใช้งานเพราะการรักษาความปลอดภัยแบบแบ่งตามระดับผู้ใช้งานนั้น ไม่สามารถเรียกดูทรัพยากรก่อนการพิสูจน์สิทธิ์ โดยผู้ใช้ต้องแจ้งชื่อผู้ใช้งานและรหัสผ่านก่อนจึงจะได้สิทธิ์เข้ามาดูและใช้ทรัพยากรตามสิทธิ์ที่ตนได้รับ โปรแกรมจึงไม่สามารถเข้าไปตรวจสอบทรัพยากรได้ ส่วนระบบที่มีการรักษาความปลอดภัยแบ่งตามระดับการแชร์นั้นไม่ต้องแจ้งชื่อผู้ใช้และรหัสผ่านก่อนการเข้าใช้งานจึงสามารถดูและตรวจสอบทรัพยากรได้ ส่วนทรัพยากรใดที่ต้องใช้รหัสผ่านจะเข้าใช้ไม่ได้

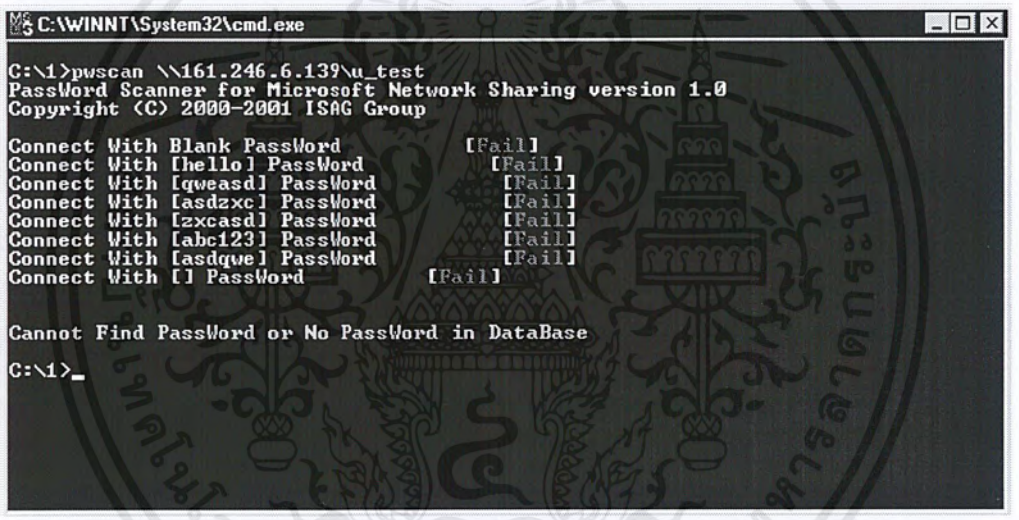
7.2 ตัวอย่างการรันโปรแกรมตรวจสอบรหัสผ่านการแชร์บนระบบเครือข่ายของวินโดวส์



รูปที่ 7-2 ตัวอย่างผลการรันโปรแกรมตรวจรหัสผ่าน (ไม่มีรหัสผ่าน)



รูปที่ 7-3 ตัวอย่างผลการรันโปรแกรมตรวจรหัสผ่าน (ตรวจพบรหัสผ่าน)



รูปที่ 7-4 ตัวอย่างผลการรันโปรแกรมตรวจรหัสผ่าน (ไม่สามารถตรวจรหัสผ่านได้)

7.2.1 ผลการรันโปรแกรม

จากรูปที่ 7-2, 7-3, 7-4 เมื่อรันโปรแกรม “PWScan \\HostName\ShareName” โปรแกรมจะทำการตรวจรหัสผ่านของทรัพยากร โดยขั้นแรกเป็นการทดสอบด้วยรหัสผ่านที่เป็นช่องว่างหรือไม่มีรหัสผ่าน (Blank) ถ้าใช้งานได้โปรแกรมจะแจ้งว่า “รหัสผ่านเป็น Blank” ถ้าใช้งานไม่ได้โปรแกรมจะอ่านรหัสผ่านจากไฟล์ฐานข้อมูลรหัสผ่านที่ละตัวแล้วลองทดสอบดูจนกระทั่งพบรหัสผ่านที่ใช้งานได้และแสดงรหัสผ่านนั้นออกมาหรือจนกระทั่งหมดข้อมูลในฐานข้อมูลก็จะแจ้งว่าไม่สามารถตรวจรหัสผ่านได้

เมื่อได้รหัสผ่านแล้วโปรแกรมก็จะทดสอบโดยการเขียนข้อมูลไปยังไดเร็กทอรีที่แชร์นั้นเพื่อตรวจว่าเป็นแบบอ่านได้อย่างเดียวหรืออ่านเขียนได้

7.2.2 ข้อจำกัดของโปรแกรม

เนื่องจากโปรแกรมที่ทำหน้าที่ในการตรวจรหัสผ่านนี้ ทำงานโดยอ่านรหัสผ่านที่เก็บไว้ในฐานข้อมูลมาใช้ในการติดต่อ ถ้าในฐานข้อมูลมีรหัสผ่านเก็บไว้มากก็มีโอกาสที่ตรวจพบสูง แต่โปรแกรมจะทำงานช้ามากเพราะโปรแกรมต้องอ่านรหัสผ่านมาทดสอบทีละตัว

7.3 โปรแกรมตรวจสอบการแชร์บนระบบเครือข่ายยูนิกซ์

7.3.1 โปรแกรมตรวจสอบการแชร์บนระบบเครือข่ายยูนิกซ์ มีหน้าที่ตรวจสอบการแชร์ไดเร็กทอรีที่แต่ละเครื่องเปิดแชร์ไว้โดยโพรโตคอลเอ็นเอฟเอส เพื่อตรวจสอบในกรณีหลงลืมว่าได้เปิดแชร์อะไรไว้บ้าง เป็นการป้องกันผู้ไม่ประสงค์ดีเข้ามาบุกรุกใช้เนื้อที่ที่แชร์ไว้ โปรแกรมนี้สามารถตรวจจับการแชร์ไดเร็กทอรีได้ทีละเครื่อง ตามหมายเลขไอพีที่ใส่เข้าไป โดยมีรูปแบบการใช้งาน คือ

```
# unixscan -<option> <ไอพีที่ต้องการตรวจสอบ>
```

เครื่องที่ต้องการตรวจสอบต้องสนับสนุนเอ็นเอฟเอสด้วย ไม่เช่นนั้น โปรแกรมจะรายงานข้อผิดพลาดออกมา อ็อปชั่นในการใช้งานมีรายละเอียดดังนี้

-e คือ การรายงานไดเร็กทอรีที่เซิร์ฟเวอร์แชร์ไว้และหมายเลขไอพีที่มีสิทธิ์ในการเข้าใช้

-? คือ คำอธิบายการใช้โปรแกรม(help)

7.3.2 ตัวอย่างและผลการทดสอบโปรแกรม UNIXSCAN

7.3.2.1 ลองทดสอบภายในเครื่องตนเอง

```
[root@isag16 yut]# unixscan -e 127.0.0.1
Exports list on 127.0.0.1:
/16share                              161.246.5.16
/16share                              161.246.5.15

[root@isag16 yut]# unixscan -e 161.246.5.16
Exports list on 161.246.5.16:
/16share                              161.246.5.16
/16share                              161.246.5.15

[root@isag16 yut]#
```

7.3.2.2 ลองทดสอบตรวจจับเครื่องอื่นๆ

```
[root@isag16 yut]# unixscan -e 161.246.10.21
Exports list on 161.246.10.21:
/misc/tmp          akharin.it.kmitl.ac.th neutrino.kmitl.ac.th Khaesad-IPATM
/opt/enware         Everyone
/opt/SUNWxt/v2.1    Everyone
/dar03/inet         neutrino.kmitl.ac.th Khaesad-IPATM
/tmp2               neutrino.kmitl.ac.th Khaesad-IPATM
/home2              neutrino.kmitl.ac.th Khaesad-IPATM
/dar01              neutrino.kmitl.ac.th Khaesad-IPATM
[root@isag16 yut]#
[root@isag16 yut]# unixscan -e 161.246.10.20
Exports list on 161.246.10.20:
/dar02              jumbo.crsc.kmitl.ac.th
[root@isag16 yut]#
```

7.3.2.3 ลองทดสอบตรวจจับเครื่องอื่นๆที่ไม่สนับสนุนโพรโตคอลที่เป็นโพรโตคอลเอ็นเอฟเอส

```
[root@isag16 yut]# unixscan -e 161.246.4.3
RPC: Program not registered
Please type :unixscan -?
[root@isag16 yut]#
```

7.3.2.4ลองทดสอบตรวจจับทั้งสับเน็ต

```
[root@isag16 yut]#unixscan -e 161.246.10.255
Exports list on 161.246.10.22:
/etc/mgc                                Chaokhun-IPATM
Exports list on 161.246.10.27:
RPC: Port mapper failure
Exports list on 161.246.10.28:
/usr                                    Everyone
/dev/rmt/0m                             Everyone
[root@isag16 yut]#
```

7.3.2.5 คำบรรยายการช่วยใช้โปรแกรม

```
[root@isag16 yut]# unixscan -?
unixscan: illegal option -- ?
  Help usage: unixscan -<option> hostname
  option    : -? is help
  option    : -e is show export list
  Example   : unixscan -e 127.0.0.1
```

บทที่ 8
สรุปและวิจารณ์

การใช้งานทรัพยากรร่วมกันในระบบเครือข่ายทั้งระบบเครือข่ายของวินโดวส์และระบบเครือข่ายยูนิกซ์มีความคล้ายคลึงกัน โดยในระบบเครือข่ายของวินโดวส์ใช้โพรโตคอลเอสเอ็มบี และในระบบเครือข่ายของยูนิกซ์ใช้โพรโตคอลเอ็นเอฟเอสเป็นหัวใจหลัก ซึ่งทั้งคู่เน้นการให้บริการแชร์ไฟล์และเครื่องพิมพ์ โดยเอสเอ็มบีมีความซับซ้อนกว่าเอ็นเอฟเอสเล็กน้อย คือ เอสเอ็มบีมีการเก็บสถานีการทำงานว่ามีใครทำงานอยู่บ้าง และยอมให้มีผู้ใช้งานหลายคนเข้าใช้งานในพื้นที่เดียวกันได้ ในขณะที่เอ็นเอฟเอสไม่ยอมเพราะอาจเกิดการเขียนข้อมูลซ้ำซ้อนกันได้

ในการทำปริญญานิพนธ์นี้ ได้ศึกษาการใช้ทรัพยากรร่วมกัน และสร้างเครื่องมือสำหรับผู้ดูแลระบบ เพื่อใช้ในการตรวจสอบการใช้ทรัพยากรร่วมกัน โดยโปรแกรมสามารถตรวจสอบการแชร์ทรัพยากรในระบบเครือข่ายของวินโดวส์และระบบเครือข่ายของยูนิกซ์ และสามารถตรวจสอบระดับการรักษาความปลอดภัยของการแชร์ทรัพยากรนั้น เพื่อเป็นแนวทางให้ผู้ดูแลระบบใช้ในการปรับแต่งระบบรักษาความปลอดภัยต่อไป

8.1 ปัญหาและอุปสรรค

- รายละเอียดเกี่ยวกับระบบเครือข่ายของวินโดวส์มีค่อนข้างน้อยเพราะไมโครซอฟท์ไม่เปิดเผยรายละเอียด
- โปรแกรมตรวจสอบรหัสผ่านของการแชร์ในระบบเครือข่ายของวินโดวส์เป็นการส่งรหัสผ่านต่างๆ เข้าไปเพื่อทำการติดต่อบางครั้งระบบเครือข่ายอาจมีข้อผิดพลาดหรือจำสถานะของข้อมูลเดิมทำให้โปรแกรมทำงานผิดพลาด
- การพิสูจน์สิทธิ์ของการแชร์ในระบบเครือข่ายของยูนิกซ์แบบเอนเอฟเอสนั้น ใช้ไอพีของเครื่องไคลเอนต์เป็นตัวพิสูจน์ จึงไม่สามารถเขียนโปรแกรมเพื่อปลอมไอพีเพื่อติดต่อขอใช้บริการจากเซิร์ฟเวอร์ได้
- การติดตั้งและปรับแต่งการแชร์แบบเอ็นเอฟเอสนั้นมีความยุ่งยากซับซ้อน ผู้ที่จะใช้งานต้องมีความชำนาญในการใช้งาน
- การแชร์แบบเอ็นเอฟเอสต้องกำหนดระดับของการเข้าใช้ทรัพยากรให้ดี เพราะผู้ใช้ไม่สามารถแก้ไขระดับของการเข้าใช้ทรัพยากรได้เอง ซึ่งอาจทำให้ใช้งานได้ไม่เต็มที่

8.2 แนวทางการวิจัยต่อ

- พัฒนาโปรแกรมตรวจสอบการแชร์บนระบบเครือข่ายของวินโดวส์ให้สามารถตรวจสอบระบบที่ใช้การรักษาความปลอดภัยแบบแบ่งตามระดับผู้ใช้งานได้

- ศึกษาวิธีการตรวจรหัสผ่านของการแชร์บนระบบเครือข่ายของวินโดวส์ โดยไม่ใช้เป็นการอ่านข้อมูลจากฐานข้อมูลมาเพื่อทำการติดต่อ เพราะการเขียนโปรแกรมแบบนี้ถ้ามีรหัสผ่านในฐานข้อมูลมากจะมีโอกาสตรวจพบสูงแต่โปรแกรมทำงานช้า และในบางครั้งระบบเครือข่ายอาจจำสถานะของข้อมูลเดิมทำให้โปรแกรมทำงานผิดพลาด

- ศึกษาเกี่ยวกับ NIS เพิ่มเติมเพื่อเป็นเครื่องมือใช้งานคู่กับ NFS

- ศึกษาวิธีปลอมไอพีเพื่อพิสูจน์สิทธิ์เข้าใช้งานในเซิร์ฟเวอร์

- ศึกษาเอ็นเอฟเอสรุ่นใหม่ๆ และแอพลิเคชันที่ใช้งานประกอบเพิ่มเติม

บรรณานุกรม

[1] Brian Komer (1988): *"Teach Yourself TCP/IP Network Administration in 21 Days"*, ISBN: 0-672-31250-6

[2] Robertson Don (1996): *"Accessing transport networks"*, 1996, ISBN: 0-07-053199-4

[3] W. Richard Stevens (1990): *"Unix Network Programming"*, Prentice-Hall of India, 1999, ISBN: 81-203-0749-6

[4] Microsoft Corporation (1995): *"Support Fundamentals for Microsoft Windows NT"*, Microsoft Press, 1995

[5] สาโรจน์ ไพชยนต์ฤทธา, วศิน เพิ่มทรัพย์, *"ใช้ Linux + Samba แทน Windows NT"*, Provision, June 1999, ISBN: 974-7821-72-9

[6] สมศักดิ์ ลิมาวงษ์ปราณี, *"RedHat Linux ฉบับเพื่อการใช้งานจริง"*, ซัคเซส มีเดีย, กรุงเทพฯ

[7] Sun Microsystems, Inc., External Data Representation Specification, RFC 1014

[8] Sun Microsystems, Inc., Remote Procedure Call Specification, RFC 1057

[9] Sun Microsystems, Inc., Network File System Specification, RFC 1094

[10] http://msdn.microsoft.com/library/specs/cifs1099.htm

[11] http://www.netapp.com/tech_library/nfsbook.html

[12] http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixbman/commadmn/ch10_nfs.html

[13] http://www.ietf.org ,RFC 1001,1002

[14] ftp:// ftp. cs. columbia. edu/ archives/ doc/ rfc (NFS V. 2/ 3/ 4 specs and drafts)

[15] ftp:// ftp. cs. columbia. edu/ archives/ doc/ internet- drafts (NFS V. 2/ 3/ 4 specs and drafts)

[16] rfc1094 Network File System2 Protocol Specification

[17] rfc1014 External Data Representation (XDR)

[18] rfc1057 Remote Procedure Call (RPC)

[19] rfc2624 NFS Version 4 Design Considerations

[20] rfc3010 NFS version 4 Protocol

[21] Microsoft Developer Network Library Visual Studio 6.0

ภาคผนวก ก.
รหัสคำสั่งเอสเอ็มบี

คำสั่งเอสเอ็มบี	รหัส
SMB_COM_CREATE_DIRECTORY	0x00
SMB_COM_DELETE_DIRECTORY	0x01
SMB_COM_OPEN	0x02
SMB_COM_CREATE	0x03
SMB_COM_CLOSE	0x04
SMB_COM_FLUSH	0x05
SMB_COM_DELETE	0x06
SMB_COM_RENAME	0x07
SMB_COM_QUERY_INFORMATION	0x08
SMB_COM_SET_INFORMATION	0x09
SMB_COM_READ	0x0A
SMB_COM_WRITE	0x0B
SMB_COM_LOCK_BYTE_RANGE	0x0C
SMB_COM_UNLOCK_BYTE_RANGE	0x0D
SMB_COM_CREATE_TEMPORARY	0x0E
SMB_COM_CREATE_NEW	0x0F
SMB_COM_CHECK_DIRECTORY	0x10
SMB_COM_PROCESS_EXIT	0x11
SMB_COM_SEEK	0x12
SMB_COM_LOCK_AND_READ	0x13
SMB_COM_WRITE_AND_UNLOCK	0x14
SMB_COM_READ_RAW	0x1A
SMB_COM_READ_MPX	0x1B
SMB_COM_READ_MPX_SECONDARY	0x1C
SMB_COM_WRITE_RAW	0x1D
SMB_COM_WRITE_MPX	0x1E
SMB_COM_WRITE_COMPLETE	0x20
SMB_COM_SET_INFORMATION2	0x22
SMB_COM_QUERY_INFORMATION2	0x23

SMB_COM_LOCKING_ANDX	0x24
SMB_COM_TRANSACTION	0x25
SMB_COM_TRANSACTION_SECONDARY	0x26
SMB_COM_IOCTL	0x27
SMB_COM_IOCTL_SECONDARY	0x28
SMB_COM_COPY	0x29
SMB_COM_MOVE	0x2A
SMB_COM_ECHO	0x2B
SMB_COM_WRITE_AND_CLOSE	0x2C
SMB_COM_OPEN_ANDX	0x2D
SMB_COM_READ_ANDX	0x2E
SMB_COM_WRITE_ANDX	0x2F
SMB_COM_CLOSE_AND_TREE_DISC	0x31
SMB_COM_TRANSACTION2	0x32
SMB_COM_TRANSACTION2_SECONDARY	0x33
SMB_COM_FIND_CLOSE2	0x34
SMB_COM_FIND_NOTIFY_CLOSE	0x35
SMB_COM_TREE_CONNECT	0x70
SMB_COM_TREE_DISCONNECT	0x71
SMB_COM_NEGOTIATE	0x72
SMB_COM_SESSION_SETUP_ANDX	0x73
SMB_COM_LOGOFF_ANDX	0x74
SMB_COM_TREE_CONNECT_ANDX	0x75
SMB_COM_QUERY_INFORMATION_DISK	0x80
SMB_COM_SEARCH	0x81
SMB_COM_FIND	0x82
SMB_COM_FIND_UNIQUE	0x83
SMB_COM_NT_TRANSACT	0xA0
SMB_COM_NT_TRANSACT_SECONDARY	0xA1
SMB_COM_NT_CREATE_ANDX	0xA2
SMB_COM_NT_CANCEL	0xA4
SMB_COM_OPEN_PRINT_FILE	0xC0
SMB_COM_WRITE_PRINT_FILE	0xC1

SMB_COM_CLOSE_PRINT_FILE	0xC2
SMB_COM_GET_PRINT_QUEUE	0xC3
SMB_COM_READ_BULK	0xD8
SMB_COM_WRITE_BULK	0xD9
SMB_COM_WRITE_BULK_DATA	0xDA

ภาคผนวก ข.
คำสั่งและโครงสร้างตัวแปรของแลนแมนเนจเจอร์ที่ใช้ในโครงงาน

NetServerGetInfo

หน้าที่

คำสั่งที่ใช้ในการอ่านรายละเอียดของเซิร์ฟเวอร์

รูปแบบ

NET_API_STATUS NetServerGetInfo (

LPWSTR *servername,*

DWORD *level,*

LPBYTE **bufptr*

);

พารามิเตอร์

servername

พอยน์เตอร์ชี้ไปที่สตริงซึ่งบรรจุชื่อของเครื่องคอมพิวเตอร์ที่ต้องการตรวจสอบ ถ้าต้องการตรวจสอบเครื่องตัวเอง (Local computer) ให้มีค่าเป็น **NULL**

level

ระดับรายละเอียดของข้อมูลที่ต้องการทราบ ดังนี้

100 คือ ให้พารามิเตอร์ *bufptr* ชี้ไปยังโครงสร้างข้อมูล **SERVER_INFO_100**

101 คือ ให้พารามิเตอร์ *bufptr* ชี้ไปยังโครงสร้างข้อมูล **SERVER_INFO_101**

102 คือ ให้พารามิเตอร์ *bufptr* ชี้ไปยังโครงสร้างข้อมูล **SERVER_INFO_102**

bufptr

พอยน์เตอร์ชี้ไปที่บัฟเฟอร์เก็บข้อมูลโดยชนิดของตัวแปรบัฟเฟอร์นี้ต้องเป็นชนิดเดียวกับที่ในพารามิเตอร์ *level* กำหนด

ค่าส่งกลับ

NERR_Success คือ ฟังก์ชันทำงานสำเร็จ

ERROR_ACCESS_DENIED คือ ไม่สามารถเข้าถึงข้อมูลตามที่ร้องขอมา

ERROR_INVALID_LEVEL คือ ไม่มีการกำหนดค่าในพารามิเตอร์ *level*

ERROR_INVALID_PARAMETER คือ พารามิเตอร์ขาดหาย

ERROR_NOT_ENOUGH_MEMORY คือ หน่วยความจำไม่พอ

SERVER_INFO_101

หน้าที่

โครงสร้างตัวแปรที่เก็บรายละเอียดของเซิร์ฟเวอร์ซึ่งบรรจุรายละเอียด คือ ชื่อ, แพลตฟอร์ม, ชนิดของเซิร์ฟเวอร์และซอฟต์แวร์อื่นๆ ที่มีส่วนร่วมในเซิร์ฟเวอร์

รูปแบบ

```
typedef struct _SERVER_INFO_101 {
        DWORD sv101_platform_id;
        LPTSTR sv101_name;
        DWORD sv101_version_major;
        DWORD sv101_version_minor;
        DWORD sv101_type;
        LPTSTR sv101_comment;
} SERVER_INFO_101, *PSERVER_INFO_101, *LPSERVER_INFO_101
```

ความหมายของสมาชิก

sv101_platform_id

รหัสหมายเลขของแพลตฟอร์ม

sv101_name

สตริงซึ่งบรรจุชื่อของเซิร์ฟเวอร์

sv101_version_major และ sv101_version_minor

เวอร์ชันของซอฟต์แวร์แลนแมนเนจเจอร์

sv101_type

หมายเลขแสดงชนิดของเซิร์ฟเวอร์ซึ่งมีความหมายดังนี้

SV_TYPE_WORKSTATION คือ เวิร์กสเตชันของแลนแมนเนจเจอร์ทุกชนิด

SV_TYPE_SERVER คือ เซิร์ฟเวอร์แลนแมนเนจเจอร์ทุกชนิด

SV_TYPE_DOMAIN_CTRL คือ ตัวควบคุมโดเมนหลัก

SV_TYPE_DOMAIN_BAKCTRL คือ ตัวควบคุมโดเมนสำรอง

SV_TYPE_NT คือ เซิร์ฟเวอร์รันวินโดวส์เอ็นที (ทั้งวินโดวส์เอ็นทีเซิร์ฟเวอร์และเวิร์กสเตชัน)

SV_TYPE_WFW คือ เซิร์ฟเวอร์ซึ่งรันวินโดวส์ฟอร์เวิร์กกรุป

SV_TYPE_WINDOWS คือ เซิร์ฟเวอร์ซึ่งรันวินโดวส์ 95/98

sv101_comment;

พอยน์เตอร์ชี้ไปยังสตริงซึ่งบรรจุหมายเหตุของเซิร์ฟเวอร์

NetShareEnum

หน้าที่

ฟังก์ชันซึ่งทำหน้าที่ในการตรวจสอบทรัพยากรที่แชร์ไว้ในเครื่องเซิร์ฟเวอร์

รูปแบบ

NET_API_STATUS NetShareEnum (

LPWSTR *servername*,

DWORD *level*,

LPTBYTE * *bufptr*,

DWORD *prefmaxlen*,

LPDWORD *entriesread*,

LPDWORD *totalentries*,

LPDWORD *resume_handle*

);

พารามิเตอร์

servername

พอยน์เตอร์ชี้ไปที่สตริงซึ่งบรรจุชื่อของเครื่องคอมพิวเตอร์ที่ต้องการตรวจสอบ ถ้าต้องการตรวจสอบเครื่องตัวเอง (Local computer) ให้มีค่าเป็น **NULL**

level

ระดับรายละเอียดของข้อมูลที่ต้องการทราบโดยมีระดับต่างๆ คือ 0, 1, 2 และ 502

bufptr

พอยน์เตอร์ชี้ไปที่บัฟเฟอร์เก็บข้อมูลโดยชนิดของตัวแปรบัฟเฟอร์นี้ต้องเป็นชนิดเดียวกับที่ในพารามิเตอร์ *level* กำหนด

prefmaxlen

กำหนดความยาวสูงสุดของผลลัพธ์ที่ส่งกลับมา

entrieread

จำนวนของทรัพยากรจริงที่ได้จากการตรวจสอบ

totalentries

จำนวนของทรัพยากรทั้งหมดที่อ่านเข้ามา

resume_handle

เก็บค่าแฮนเดิ้ลเพื่อใช้ในการค้นหาทรัพยากรที่แชร์ตัวอื่นต่อไป โดยกำหนดให้มีค่าเริ่มต้นเป็นศูนย์

SHARE_INFO_1

หน้าที่

เป็นโครงสร้างข้อมูลซึ่งใช้เก็บรายละเอียดของทรัพยากรที่เปิดแชร์ โดยมีรายละเอียดเกี่ยวกับชื่อ, ชนิดและหมายเหตุของทรัพยากรนั้น

รูปแบบ

```
typedef struct _SHARE_INFO_1 {
        LPWSTR shi1_netname;
        DWORD shi1_type;
        LPWSTR shi1_remark;
} SHARE_INFO_1, *PSHARE_INFO_1, *LPSHARE_INFO_1;
```

หน้าที่ของสมาชิก

shi1_netname

สตริงซึ่งบรรจุชื่อที่ใช้ในการแชร์ของทรัพยากร

shi1_type

ชนิดของทรัพยากรซึ่งมีรายละเอียด คือ

STYPE_DISKTREE	คือ ดิสก์หรือไดเรกทอรี
STYPE_PRINTQ	คือ เครื่องพิมพ์
STYPE_DEVICE	คือ อุปกรณ์สื่อสาร
STYPE_IPC	คือ Interprocess Communication (IPC)

shi1_remark

พอยน์เตอร์ชี้ไปที่สตริงซึ่งบรรจุหมายเหตุของทรัพยากรนั้น

WnetAddConnection2

หน้าที่

ฟังก์ชันซึ่งทำหน้าที่สร้างการเชื่อมต่อกับทรัพยากรในระบบเครือข่าย โดยเมื่อสร้างการเชื่อมต่อได้แล้วทรัพยากรนั้นจะเสมือนเป็นทรัพยากรในเครื่องตนเอง

รูปแบบ

DWORD WnetAddConnection2 (

LPNETRESOURCE *lpNetResource*,

LPCTSTR *lpPassword*,

LPCTSTR *lpUsername*,

DWORD *dwFlags*

);

พารามิเตอร์

lpNetResource

พอยน์เตอร์ชี้ไปที่ตัวแปรโครงสร้าง **NETRESOURCE** ซึ่งบรรจุรายละเอียดเกี่ยวกับทรัพยากรในระบบเครือข่ายที่ต้องการเชื่อมต่อ, ทรัพยากรท้องถิ่น (Local Device) และข้อจำกัดต่างๆของทรัพยากร โดยสมาชิกของตัวแปรโครงสร้าง **NETRESOURCE** มีดังนี้

dwType คือชนิดของทรัพยากรที่ต้องการติดต่อ ถ้าใน **lpLocalName** เป็นไม่สตริงว่าง (Non-empty String) ตัวแปรนี้มีค่าเป็น RESOURCETYPE_DISK หรือ RESOURCETYPE_PRINT แต่ถ้ามีค่าเป็นค่าว่าง (NULL) ตัวแปรนี้มีค่าเป็น RESOURCETYPE_DISK, RESOURCETYPE_PRINT หรือ RESOURCETYPE_ANY

lpLocalName คือชื่อของทรัพยากรท้องถิ่น (Local Device) เพื่อให้ทรัพยากรในเครือข่ายเชื่อมโยงการเชื่อมต่อด้วย เช่น ชื่อดิสก์ไดร์ฟ หรือ พอร์ตเครื่องพิมพ์ ("X:" หรือ "LPT1") ถ้าในตัวแปรนี้เป็นค่าว่าง จะไม่มีการเชื่อมโยงการเชื่อมต่อมายังทรัพยากรท้องถิ่น

lpRemoteName คือพอยน์เตอร์ชี้ไปยังสตริงซึ่งบรรจุชื่อของทรัพยากรบนระบบเครือข่ายที่ต้องการสร้างการเชื่อมต่อด้วย

lpProvider คือพอยน์เตอร์ชี้ไปที่สตริงซึ่งบรรจุข้อมูลเกี่ยวกับการจัดหาช่องทางในการไปเชื่อมต่อกับทรัพยากรในระบบเครือข่าย โดยในตัวแปรนี้กำหนดให้เป็นค่าว่างได้เพื่อให้ระบบปฏิบัติการตัดสินใจหาช่องทางในการเข้าถึงทรัพยากรเอง

lpPassword

พอยน์เตอร์ชี้ไปยังสตริงซึ่งบรรจุรหัสผ่านในการเข้าใช้ทรัพยากรในระบบเครือข่าย โดยรหัสผ่านนี้ต้องมีความสัมพันธ์กับชื่อผู้ใช้งานซึ่งอยู่ในพารามิเตอร์ *lpUsername* ถ้ารหัสผ่านเป็นค่าว่างคือให้ใช้รหัสผ่านที่ผูกติดกับชื่อผู้ใช้งาน (Default Password)

lpUsername

พอยน์เตอร์ชี้ไปยังสตริงซึ่งบ รรจุชื่อผู้ใช้เพื่อการเข้าใช้งานทรัพยากรในระบบเครือข่ายนั้นถ้าเป็นค่าว่าง คือการใช้ชื่อผู้ใช้งานระบบที่ใช้อยู่ในขณะนั้น (Default User)

dwFlags

คือการตั้งค่าใช้งานการเชื่อมต่อนั้น

ค่าส่งกลับ

ERROR_ACCESS_DENIED คือ ไม่สามารถเข้าถึงทรัพยากรนั้นได้

ERROR_ALREADY_ASSIGNED คือ ทรัพยากรท้องถิ่นใน **lpLocalName** มีการใช้งานแล้ว

ERROR_BAD_DEV_TYPE คือ ทรัพยากรท้องถิ่นและทรัพยากรที่เชื่อมต่อไม่สัมพันธ์กัน

ERROR_BAD_DEVICE คือ ไม่พบทรัพยากรท้องถิ่นที่ให้เชื่อมต่อด้วย

ERROR_BAD_NET_NAME คือ ไม่พบทรัพยากรบนระบบเครือข่าย

ERROR_BUSY คือ ระบบเครือข่ายมีการจราจรที่หนาแน่นมาก (Busy)

ERROR_INVALID_PASSWORD คือ รหัสผ่านผิด

ERROR_NO_NETWORK คือ ไม่พบระบบเครือข่าย

ภาคผนวก ค.

ค่าผิดพลาดที่ส่งกลับมาของอาร์พีซี

The following gives a brief description of the possible RPC call return codes.

Value	Meaning
RPC_SUCCESS	Successful completion.
RPC_CANT_ENCODE_ARGS XDR	error encoding arguments.
RPC_CANT_DECODE_RES XDR	error decoding results.
RPC_CANT_SEND	Socket transmit error.
RPC_CANT_RECV	Socket receive error.
RPC_TIMED_OUT No	response from server.
RPC_VERS_MISMATCH	Server does not support RPC version.
RPC_AUTH_ERROR	Permission denied.
RPC_PROG_UNAVAIL	Program not registered on server.
RPC_PROG_VERS_MISMATCH	Incompatible program version on server.
RPC_PROC_UNAVAIL	Procedure not registered.
RPC_CANT_DECODE_ARGS	Server unable to decode arguments.
RPC_SYSTEM_ERROR Local	internal RPC error (e.g. nomemory).
RPC_UNKNOWN_HOST	Specified host does not exist.
RPC_PMAP_FAILURE	Unable to obtain remote port from port mapper.
RPC_PROG_NOT_REGISTERED	Remote port for program not known.
RPC_UNABLE_TO_REGISTER	Unable to register service with por mapper
RPC_UNABLE_TO_REMOVE	Unable to remove service from port mapper
RPC_FAILED	General 'other' error.
RPC_CALL_INPROGRESS	Asynchronous RPC call is in progress
RPC_STALE_RACHANDLE	Rac handle is no longer valid