บทที่ 2
ระบบไฟล์ในลีนุกซ์

ระบบไฟล์ของระบบลีนุกซ์นั้น สำหรับผู้ที่คุ้นเคยกับระบบไฟล์ในยูนิกซ์ก็สามารถเข้าใจได้ง่าย ส่วนผู้ที่ไม่คุ้นเคยก็ไม่ใช่เรื่องยาก ระบบลีนุกซ์นั้นนับว่าเป็นระบบปฏิบัติการยูนิกซ์ชนิดหนึ่ง ซึ่งมีการอ้างถึงทุกสิ่งทุกอย่างในระบบเป็นไฟล์ทั้งสิ้น ไม่ว่าเป็นอุปกรณ์ฮาร์ดแวร์ โปรแกรม หรืออะไรก็ตามแต่เนื่องจากปริมาณของไฟล์นั้นมีมาก ดังนั้นการจัดโครงสร้างของระบบไฟล์ในลีนุกซ์จึงเป็นสิ่งสำคัญ

2.1 ระบบไฟล์

ระบบไฟล์คือโครงสร้างการจัดเก็บข้อมูลในฮาร์ดดิสก์ เพื่อให้ระบบปฏิบัติการสามารถอ่านเขียนใช้งานไฟล์ที่ต้องการได้อย่างมีประสิทธิภาพ และควบคุมสิทธิในการเข้าใจของผู้ใช้ เช่น ในระบบปฏิบัติการดอส (Disk Operating System) ที่มีการใช้งานมาเป็นระยะเวลายาวนานมีระบบไฟล์ที่เรียกว่า FAT (File Allocation Table) ซึ่งมีทั้งแบบ 12 และ 16 บิต และสำหรับระบบไฟล์ในไมโครซอฟต์วินโดวส์ 95 ของไมโครซอฟต์ และ OS/2 ของไอบีเอ็มสนับสนุนระบบไฟล์ทั้งแบบ FAT12, FAT16 และ FAT32 แต่ระบบปฏิบัติการทั้งสองแบบเป็นระบบปฏิบัติการแบบผู้ใช้เดียว ทำให้ระบบไฟล์ไม่มีส่วนควบคุมสิทธิการเข้าใช้ แต่ระบบปฏิบัติการใหม่ๆ เช่น ไมโครซอฟต์วินโดวส์ เอ็นที จะใช้ระบบไฟล์ที่เรียกว่า NTFS (NT File System) หรือระบบปฏิบัติการ IBM OS/2 จะใช้ระบบไฟล์ที่เรียกว่า HPFS (High Performance File System) ซึ่งระบบไฟล์ทั้งสองจะมีส่วนที่ควบคุมสิทธิการเข้าใช้ของผู้ใช้ มีส่วนที่สนับสนุนการเข้าใช้จากผู้ใช้หลายคน

ระบบปฏิบัติการ	ระบบไฟล์
DOS	FAT12, FAT16
MS Windows 95	FAT12, FAT16
MS Windows 95, MS Windows 98, MS Windows ME	FAT12, FAT16, FAT32
MS Windows NT, MS Windows 2000	FAT12, FAT16, NTFS, HPFS
IBM OS/2	FAT12, FAT16, HPFS
Linux	FAT12, FAT16, FAT32, HPFS, NTFS, Ext1, Ext2, UFS, VFAT, Minix, ISOFS, HFS, AFFS, ADFS, SYSV

ตารางที่ 2-1 ระบบไฟล์ของระบบปฏิบัติการชนิดต่างๆ

2.2 ระบบชื่อไฟล์ในระบบลีนุกซ์

ระบบการตั้งชื่อไฟล์ในระบบปฏิบัติการระบบลีนุกซ์นั้นต่างจากดอสคือ ดอสจะให้มีการตั้งชื่อไฟล์ด้วยชื่อขนาด 8 ตัวอักษร และสกุลอีก 3 ตัวอักษร ส่วนระบบไฟล์ของระบบลีนุกซ์ในรุ่นแรกๆ นั้นจะใช้ระบบไฟล์ของ Minix (Minix File System) ซึ่งเป็นระบบไฟล์ที่ถูกใช้ในระบบปฏิบัติการ Minix นี้จะมีขีดความสามารถจำกัดในการเก็บชื่อไฟล์ได้สูงสุดเพียง 14 ตัวและมีข้อจำกัดมากมายในระบบไฟล์นี้ต่อมาได้มีการปรับปรุงและพัฒนาเพิ่มเพื่อให้สามารถเก็บชื่อไฟล์ที่มีขนาดเพิ่มขึ้นได้ถึง 30 ตัวอักษรสำหรับลีนุกซ์ในปัจจุบันนี้ใช้ระบบไฟล์ที่เรียกว่า Ext2 (Extended Files System 2) ซึ่งสามารถตั้งชื่อไฟล์ได้ถึง 255 ตัวอักษร

การตั้งชื่อไฟล์ในระบบลีนุกซ์นั้นสามารถใช้ตัวอักษร ตัวเลข ขีดเส้นใต้ และยังให้ความแตกต่างกันระหว่างอักษรเล็กกับอักษรตัวใหญ่ในรูปแบบที่เรียกว่า เคสเซนซิทีฟ (Case Sensitive) ด้วย เช่น FILE1, file1 และ File1 นั้นชื่อไฟล์ทั้ง 3 มีความแตกต่างกัน

สำหรับการตั้งชื่อไฟล์นั้นควรหลีกเลี่ยงการใช้เครื่องหมายพิเศษต่างๆ เช่น ^ " ' , - ? [ ] ( ) ~ ! $ { } < > # @ & / และยิ่งกว่านั้นถ้าไฟล์ใดที่มีชื่อขึ้นต้นด้วย “ . ” (dot) จะกลายเป็นไฟล์ที่ถูกซ่อนไว้ (hidden file) ซึ่งจะไม่แสดงรายชื่อไฟล์ออกมาให้เห็น หากต้องการเรียกไฟล์ที่ถูกซ่อนขึ้นมาดูต้องเพิ่มพารามิเตอร์พิเศษให้กับคำสั่งในการขอดูรายชื่อไฟล์

ระบบไฟล์	ขนาดของชื่อไฟล์(ตัวอักษร)	ขนาดไฟล์ (เมกกะไบต์)
Minix File System	14	64
Extended File System(Ext1)	255	2048
Extended File System(Ext2)	255	2048

ตารางที่ 2-2 ระบบชื่อและขนาดของไฟล์ในรูปแบบต่างๆ

2.3 ระบบไฟล์ในลีนุกซ์

ขนาดของไฟล์ในระบบลีนุกซ์รุ่นแรกๆ นั้นไม่สามารถเก็บไฟล์ที่มีขนาดใหญ่เกินกว่า 64 เมกะไบต์ได้ทำให้ผู้ใช้งานที่ต้องการจะใช้ไฟล์ขนาดใหญ่บนลีนุกซ์ไม่สามารถทำได้ จึงได้มีการออกแบบและสร้างระบบไฟล์ขึ้นใหม่เรียกว่า Ext (Extended File System) เมื่อเดือนเมษายน ปี 2535 เป็นระบบไฟล์ที่สองที่ระบบปฏิบัติการลีนุกซ์สนับสนุนการทำงานด้วย โดยระบบไฟล์นี้ถูกสร้างโดย "เรมี คาร์ด" (Remy Card) เพื่อให้ระบบสามารถสนับสนุนการใช้งานไฟล์ขนาดใหญ่ โดยระบบไฟล์นี้มีพื้นฐานมาจากระบบไฟล์ใน Minix ทำให้มีข้อจำกัดในระบบไฟล์ของระบบลีนุกซ์ลดลง แต่ผลจากการออกแบบส่วนจัดการพื้นที่ว่างที่ยังไม่ดีพอทำให้ระบบจะเกิดพื้นที่ว่างที่ไม่ต่อเนื่องกันได้ง่ายทำให้เกิดปัญหาที่เรียกว่าแฟรก

เมนต์เตชั่น(fragmentation) ส่งผลให้ระบบทำงานช้ากว่าระบบไฟล์ของ Minix ที่เป็นระบบไฟล์ต้นแบบ ต่อมา "เรมี คาร์ด" และ "เวย์เน่ ดาวิสัน" (Wayne Davison) ได้ทำมีการออกแบบระบบไฟล์ใหม่เรียกว่า ระบบไฟล์แบบ Ext2 (Extended Files System 2) ในเดือนมกราคม ปี 2536 หลังจากที่เก็บข้อมูลความต้องการจากผู้ใช้ที่ต้องการระบบไฟล์ที่มีประสิทธิภาพ เป้าหมายในการออกแบบระบบไฟล์นี้จึงมุ่งไปที่ประสิทธิภาพและความถูกต้องของข้อมูล และสามารถรองรับขนาดของพาร์ทิชันที่สูงถึง 4 เทราไบต์ ระบบไฟล์นี้มีพื้นฐานมาจากระบบไฟล์ Ext ระบบไฟล์นี้เป็นที่ยอมรับใช้งานจนถึงปัจจุบัน

2.4 การจัดแบ่งพาร์ทิชันของระบบไฟล์

ในการติดตั้งระบบปฏิบัติการเรดแฮ็ตลีนุกซ์แบบเซิร์ฟเวอร์ (Server) โปรแกรมการติดตั้งจะสร้างพาร์ทิชันเพื่อที่จะใช้ในการติดตั้งระบบปฏิบัติการลีนุกซ์จะแตกต่างกับระบบปฏิบัติการดอสและวินโดว์ที่เราคุ้นเคย ด้วยระบบปฏิบัติการดอสและวินโดว์จะใช้วิธีการอ้างถึงพาร์ทิชันต่างๆ โดยการอาศัยตัวอักษรประจำพาร์ทิชัน เช่น C:,D: หรือ F: เป็นต้น แต่ในระบบปฏิบัติการลีนุกซ์ จะมองระบบไฟล์ที่จัดเก็บไว้ในแต่ละพาร์ทิชันต่อเนื่องกันตลอดทุกพาร์ทิชัน โดยเริ่มจากไดเรกทอรีราก (root directory) ต่อเนื่องไปยังไดเรกทอรีต่างๆ ที่ถูกจัดเก็บไว้ในแต่ละพาร์ทิชัน แต่อย่างไรก็ตามระบบปฏิบัติการจะไม่ได้มองพื้นที่ว่างทั้งหมดในแต่ละพาร์ทิชันเป็นพื้นที่ว่างผืนเดียวกันหมด

การจัดระบบไฟล์ของลีนุกซ์แบบนี้ทำให้การจัดเก็บไฟล์ในแต่ละไดเรกทอรีขึ้นกับว่าไดเรกทอรีนั้นเก็บในพาร์ทิชันใด ดังนั้นไฟล์ที่จัดเก็บจึงต้องเก็บในพาร์ทิชันเดียวกันด้วย ผลจากการจัดการไฟล์ด้วยวิธีการนี้จึงต้องมีการกำหนดจุดเชื่อมต่อ (Mount Point) ให้กับแต่ละพาร์ทิชันต่างๆ เพื่อให้เข้าถึงพาร์ทิชันทั้งหมดได้

ไดเรกทอรีรากจะเก็บไฟล์โปรแกรมและไฟล์คุณสมบัติที่จำเป็นในการบูตระบบ และไดเรกทอรีที่ใช้ในการเชื่อมต่อกับพาร์ทิชันอื่นๆ ดังนั้นพาร์ทิชันหลัก (root partition หรือ main partition) จะถูกเชื่อมต่อเข้ากับระบบปฏิบัติการลีนุกซ์เป็นลำดับแรก ถ้าพาร์ทิชันหลักเสียหายจะทำให้ไม่สามารถบูตหรือเริ่มต้นการทำงานของระบบปฏิบัติการได้ ด้วยวิธีการนี้ระบบไฟล์ของระบบลีนุกซ์จึงมีเพียงหนึ่งไดเรกทอรีแต่สามารถทำงานบนระบบที่มีฮาร์ดดิสก์กี่ตัวก็ได้

การที่ระบบปฏิบัติการลีนุกซ์ทำการแยกย่อยฮาร์ดดิสก์ออกเป็นหลายๆ พาร์ทิชันแทนการให้พาร์ทิชันขนาดใหญ่เพียงหนึ่งหรือสองพาร์ทิชัน เพราะต้องการให้แต่ละพาร์ทิชันมีการจองขนาดการใช้พื้นที่ใช้สอยไว้ล่วงหน้า เพื่อป้องกันการใช้พื้นที่ว่างจนหมดแล้วทำให้ระบบปฏิบัติการไม่สามารถทำงานต่อไปได้ ยกตัวอย่างเช่น ถ้ามีการติดตั้งโปรแกรมประยุกต์จนเต็มพื้นที่ใช้สอยทั้งหมดจะทำให้ระบบปฏิบัติการลีนุกซ์ไม่สามารถทำงานต่อไปได้

การจัดเก็บไฟล์ในลีนุกซ์จะเก็บในรูปของบล็อก (Block) โดยจะมีขนาดเป็นจำนวนเท่าของ 512 ไบต์ ขึ้นอยู่กับการกำหนดขนาดของระบบนั้นๆ ซึ่งการกำหนดขนาดเล็กหรือใหญ่ก็มีข้อดี ข้อเสียแตกต่างกันไป หากบล็อกมีขนาดใหญ่จะทำให้การแลกเปลี่ยนข้อมูลระหว่างดิสก์กับหน่วยความจำทำได้รวดเร็ว

แต่อาจจะทำให้เกิดการสูญเสียเนื้อที่ (fragmentation) ไปได้โดยง่ายการจัดเก็บโดยปกติจะเก็บเป็นบล็อกที่ติดต่อกัน แต่ถ้ามีไฟล์อื่นมาแทรกก็สามารถกระโดดข้ามบล็อกนั้นไปได้ เนื่องจากมีตารางบอกว่าไฟล์นั้นเก็บข้อมูลอยู่ที่บล็อกใดบ้างโดยเก็บไว้ในส่วนของไอโหนด (inode) ซึ่งจะกล่าวในลำดับต่อไป

2.5 การจัดไดเรกทอรีของระบบไฟล์ลีนุกซ์

ลีนุกซ์นั้นใช้โครงสร้างของไฟล์และไดเรกทอรีแบบของ FSSTND (Linux File System Standard) ซึ่งต่อมาได้มีระบบหลายระบบนำรุปแบบการจัดการไฟล์และไดเรกทอรีแบบนี้ไปใช้และพัฒนาเป็นระบบโครงสร้างใหม่ที่เรียกว่า FHS (Filesystem Hierarchy Standard) แทน โดยมีการจัดเป็นแบบลำดับชั้น โดยมีลักษณะคล้ายกับต้นไม้กลับหัว โดยจุดเริ่มต้นหรือชั้นแรกจะเรียกว่าเป็นราก (root) ซึ่งจะเขียนแทนด้วยเครื่องหมาย / ไฟล์แต่ละไฟล์นั้นอาจจะเป็นข้อมูลที่สร้างขึ้น หรือโปรแกรมก็ได้ หรือแม้แต่ใช้เป็นที่เก็บไฟล์เองก็ได้ ซึ่งไฟล์ลักษณะนี้จะจัดว่าเป็นไฟล์ประเภทไดเรกทอรีรูปแบบการจัดเก็บไฟล์แบบลำดับชั้นนี้จะทำให้การจัดเป็นไฟล์เป็นระบบ ทำให้ง่ายต่อการใช้งานและดูแลโครงสร้างหลักของการจัดระบบไฟล์

2.6 ไอโหนด (Inode)

เคอร์เนลจะไม่สามารถติดต่อหรือรู้จักไฟล์โดยตรงแต่เคอร์เนลจะรับรู้สิ่งต่างๆ เกี่ยวกับไฟล์ที่มีอยู่ได้จากโครงสร้างข้อมูลที่เรียกว่าไอโหนด โดยไอโหนดจะประกอบด้วยข้อมูลต่างๆ เกี่ยวกับไฟล์เป็นจำนวนมาก เช่น การแสดงสิทธิต่างๆ ของไฟล์

- การบอกถึงชนิดของไฟล์
- แสดงถึงเจ้าของ และกลุ่มของเจ้าของไฟล์
- วัน เวลา ที่สร้างไฟล์ เปลี่ยนแปลงไฟล์ หรือแอกเซสไฟล์
- จำนวนลิงค์ที่ไฟล์เชื่อมต่อด้วย
- เก็บตำแหน่งที่อยู่ของข้อมูลในไฟล์

ไอโหนดนั้นจะถูกสร้างขึ้นมาโดยอัตโนมัติตอนสร้างระบบไฟล์ และถ้าหากไอโหนดนั้นเต็มจะไม่สามารถสร้างไฟล์ขึ้นมาได้เลย ถึงแม้ว่าจะยังมีเนื้อที่ว่างอยู่

2.7 ประเภทของไฟล์

ไฟล์ในระบบลีนุกซ์ได้มีการแบ่งประเภทออกไปหลายรูปแบบ และรูปแบบนั้นยังขึ้นอยู่กับกลุ่มของผู้สร้างลีนุกซ์ด้วย

- ไฟล์ทั่วไป (Regular file)
- ไดเรกทอรี (Directories)
- ไฟล์พิเศษ (Character and Block device file)
- ไฟล์ที่เชื่อมต่อ (Link File)

2.7.1 ไฟล์ทั่วไป (Regular file)

เป็นไฟล์ธรรมดาที่อาจสร้างขึ้นเองจากเอดิเตอร์ หรืออาจจะทำการสำเนามาจากไฟล์อื่น หรืออาจจะเป็นโปรแกรมใช้งานต่างๆ ก็ได้ โดยลักษณะการรายงานละเอียดต่างๆ ของไฟล์ เช่น สิทธิไฟล์ เจ้าของไฟล์ กลุ่มเจ้าของไฟล์ ขนาดหรือวันเวลาที่ทำการสร้าง เปลี่ยนแปลงแก้ไขไฟล์ ซึ่งจะได้ข้อมูลต่างๆ นี้มาจากไอโหนด (ซึ่งจะกล่าวในลำดับต่อไป) นั่นเอง

```
[arnut@database / sbin]$ ls -al | more
total 4137
drwxr - xr -x 3 root root 2048 Oct 4 11:34 .
drwxr - xr -x 17 root root 1024 Oct 2 03:52 ..
-rwxr - xr -x 1 root root 29544 Jun 18 22:05 arp
-rwxr - xr -x 1 root root 8244 Mar 22 1999 badblocks
-rwsr - xr -x 1 root root 10708 Jun 2 19:51 cardctl
-rwxr - xr -x 1 root root 35096 Jun 2 19:51 cardmgr
-rwxr - xr -x 1 root root 56728 Apr 17 1999 cfdisk
-rwxr - xr -x 1 root root 21192 Apr 19 1999 chkconfig
-rwsr - sr -x 1 root root 147 Aug 16 11:05 chkyppasswdd
-rwxr - xr -x 1 root root 4276 Apr 17 1999 ctrlaltdel
-rwxr - xr -x 1 root root 33364 Mar 22 1999 debugfs
-rwxr - xr -x 1 root root 27316 Apr 20 1999 depmod
-rwxr - xr -x 1 root root 6620 Mar 22 1999 dumpe2fs
-rwxr - xr -x 1 root root 410716 Mar 22 1999 e2fsck
-rwxr - xr -x 1 root root 4916 Mar 22 1999 e2label
-rwxr - xr -x 1 root root 93980 Apr 17 1999 fdisk
--More—
```

2.7.2 ไดเร็กทอรี (Directory)

เป็นไฟล์ชนิดหนึ่งซึ่งใช้เก็บบรรดาไฟล์ต่างๆ หรือไดเรกทอรีด้วยกันเอง ปริมาณไฟล์ในระบบลีนุกซ์ นั้นมีจำนวนมาก จึงจำเป็นต้องมีรูปแบบการจัดเก็บไฟล์ให้อยู่ในไดเรกทอรี่ที่เหมาะสม เพื่อให้ผู้ใช้สามารถทำนายได้ว่าไฟล์ควรอยู่ในไดเร็กทอรีใด

2.7.3 ไฟล์พิเศษ(Block&Character device files)

ลีนุกซ์มีไฟล์ชนิดพิเศษอยู่ 2 ชนิดคือ แบบบล็อก (block device file) และแบบตัวอักษร (Character device file) โดยลักษณะความแตกต่างกันของแบบบล็อกกับแบบคาเรกเตอร์ คือ ลักษณะการรับส่งข้อมูล ซึ่งถ้าเป็นไฟล์แบบตัวอักษรจะมีการรับส่งทีละ 1 ตัวอักษร การส่งแบบบล็อกนั้นจะมีการรับส่งข้อมูลทีละบล็อก ซึ่งอาจเป็น 512, 1024 หรือ 2048 ตัวอักษรก็ได้ ไฟล์ชนิดพิเศษเหล่านี้จะอยู่ในไดเรกทอรี “dev” ซึ่งเป็นไดเรกทอรีที่เก็บอุปกรณ์ต่างๆ ของระบบ เช่น ฮาร์ดดิสก์ ดิสก์ไดรฟ์ จอภาพ และ หน่วยความจำ เป็นต้น

2.7.4 ไฟล์ที่เชื่อมต่อ (Link Files)

ลักษณะการเชื่อมต่อไฟล์นั้นมีอยู่ 2 แบบคือการเชื่อมต่อทางกายภาพ (Hard Link) กับการเชื่อมต่อโดยนัย (Symbolic Link)

2.7.4.1 การเชื่อมต่อกายภาพ (Hard Link) การเชื่อมต่อแบบนี้จะมีการใช้ไอโหนด เดียวกันกับไฟล์ต้นฉบับ เสมือนกับการสร้างไฟล์ใหม่ขึ้นมา เพียงแต่ค่าไอโหนดนั้นซ้ำกับของเดิม และ ไอโหนด จะมีตัวนับจำนวนไฟล์ที่เชื่อมต่อด้วย

2.7.4.2 การเชื่อมต่อโดยนัย (Symbolic Link) เป็นลักษณะการเชื่อมต่อเช่นเดียวกันแต่จะมีการสร้างไอโหนด ของไฟล์ตัวเองขึ้นมาใหม่ โดยไฟล์ใหม่ที่ทำการสร้างขึ้นจะเก็บไดเร็กทอรีและชื่อไฟล์ที่ต้องการเชื่อมต่อด้วย (เหมือนกับการสร้าง Shortcut ใน Windows) หากมีการเปลี่ยนแปลงไฟล์ต้นฉบับจะส่งผลกระทบต่อไฟล์ที่เชื่อมต่อด้วย

2.8 การจัดการไฟล์และไดเร็กทอรี

- **การคัดลอก (COPY)**

เราสามารถทำการสำเนาแฟ้มข้อมูล (หรืออาจจะเป็นไดเรกทอรีด้วยก็ได้) โดยใช้คำสั่ง

```
# cp file1 file2
```

คำสั่งข้างบนจะทำการสำเนาแฟ้มข้อมูล (copy) จาก file ไปเป็นชื่อ file2

- **การลบ (REMOVE)**

  หากต้องการทำการลบแฟ้มข้อมูล (หรืออาจจะเป็นไดเรกทอรี) ก็สามารถใช้คำสั่ง

  # rm filename

- **การย้าย/เปลี่ยนชื่อ (MOVE)**

  เราสามารถใช้คำสั่ง mv เพื่อทำการย้ายแฟ้มจากไดเรกทอรีหนึ่ง ไปสู่อีกไดเรกทอรีหนึ่ง

  # mv filename directory

  คำสั่งข้างบนจะทำการย้ายแฟ้มชื่อ filename จากไดเรกทอรีปัจจุบัน ไปสู่ไดเรกทอรีที่ ชื่อdirectory นอกจากนี้เราสามารถใช้คำสั่ง mv เพื่อใช้ทำการเปลี่ยนชื่อไฟล์ได้

  # mv file1 file2

  คำสั่งนี้จะทำการเปลี่ยนชื่อจาก file1 ไปเป็นชื่อ file2

- **การ mount ระบบไฟล์**

  การ mount จะเป็นการนำระบบไฟล์เข้าไปเชื่อมต่อกับระบบไฟล์เดิม โดยการนำระบบไฟล์ที่ต้องการเชื่อมต่อมาทับตรงจุดที่ต้องการเชื่อม ซึ่งเรียกว่า mount point หลังจากทำการเชื่อมต่อแล้วระบบไฟล์ที่นำมา mount นั้นจะเป็นเสมือนส่วนหนึ่งของระบบไฟล์ การ mount ในระบบลีนุกซ์นั้นจะอนุญาตเฉพาะผู้ดูแลระบบและนอกจากการ mount ระบบไฟล์แล้วยังสามารถ mount อุปกรณ์ให้เป็นเสมือนกับไฟล์หนึ่งในระบบก็กระทำได้อย่างเช่น

  การ Mount CD-ROM

  # mount /dev/cdrom /mnt/cdrom

  การ Mount Floppy Disk

  # mount /dev/fd0 /mnt/floppy

- **การยกเลิกการ Mount**

  สำหรับการยกเลิกการ Mount นั้นสามารถยกเลิกโดยใช้คำสั่ง unmount

  # cd /root

  # umount /dev/cdrom

- **การกำหนดสิทธิการใช้งานไฟล์ (File Permission)**

  ลีนุกซ์เป็นระบบปฏิบัติการที่มีการใช้งานไฟล์ต่างๆ ร่วมกันในกลุ่มของผู้ใช้งาน หากให้สิทธิแก่ผู้ใช้ทั้งหมดในการจัดการหรือกระทำใดได้กับไฟล์นั้นย่อมก่อให้เกิดความวุ่นวายและสับสนได้ ดังนั้นลีนุกซ์จึงได้มีการจำกัดสิทธิการใช้งานไฟล์โดยกำหนดประเภทของผู้ใช้งานไฟล์ออกเป็น 3 กลุ่มใหญ่ๆ คือ

  - เจ้าของไฟล์หรือผู้สร้างไฟล์ (owner)
  - ผู้ใช้ในกลุ่มเดียวกับเจ้าของไฟล์ (group)

- บุคคลอื่น (other or universe)

สิทธิในไฟล์นั้นประกอบไปด้วยการให้สิทธิในการอ่านไฟล์ "r" หรือ "read" เขียนหรือแก้ไขเปลี่ยนแปลงไฟล์ "w" หรือ "write" และเรียกปฏิบัติงานหรือค้นหา "x" หรือ "execute" , "search" และการไม่มีสิทธิจะใช้สัญลักษณ์ขีด "-"

รูปแบบการแสดงสิทธิในไฟล์นั้นจะใช้ตัวเลข 9 บิต หรือเรียกว่า nine permission bits โดย 3 บิตแรกเป็นการแสดงสิทธิของเจ้าของไฟล์ 3 บิตถัดมานั้นเป็นการแสดงสิทธิของกลุ่มเจ้าของไฟล์และ 3 บิตสุดท้ายเป็นการแสดงสิทธิของบุคคลอื่น ดังตัวอย่างในตารางที่ 2-4

สิทธิ์การใช้งาน	รหัส
-rwxrwxrwx	777
-rwxrwx---	770
-rwx------	700
-rw-rw-rw-	666
-rw-rw----	660
-rw-------	600
-r--------	400

ตารางที่ 2-3 สิทธิในการเข้าใช้งาน

ตัวอย่างเช่น เมื่อใช้คำสั่งขอดูไฟล์ชื่อ file1 จะพบรหัสแสดงสิทธิดังนี้

[arnut@database arnut]$ ls -l file1

-rw-r- -r- - 1 arnut lecturer 0 Oct 22 17:58 file1

[arnut@database arnut]$

ตำแหน่งแรกที่ปรากฏทางซ้ายของบรรทัดนั้นใช้บ่งประเภทของไฟล์ในที่นี้เป็นรหัส "-" หมายถึงไฟล์ทั่วไป ค่าสิทธิของเจ้าของแฟ้มเป็น "rw-" คืออ่านและแก้ไขไฟล์ได้ ส่วนผู้ใช้ในกลุ่มเจ้าของไฟล์และบุคคลอื่นนั้นมีสิทธิเป็น " r- - " คือสามารถอ่านไฟล์ได้อย่างเดียวนั้น

การใช้งาน	สิทธิ์ขั้นต่ำ	ตัวอย่างการใช้งาน
พิมพ์ไฟล์	-r	$ cat sample_file
แก้ไขไฟล์	-rw	$ vi sample_file
ลบไฟล์	-w	$ rm sample_file
เพิ่มข้อมูล	-w	$ cat >> sample_file
เรียกใช้โปรแกรมไบนารี่	-x	$ run binary
เรียกใช้สคริปต์	-rw	$ run script

ตารางที่ 2-4 สิทธิขั้นต่ำในการใช้งานไฟล์

- **การเปลี่ยนแปลงสิทธิต่างๆ ในไฟล์**

การเปลี่ยนแปลงสิทธิต่างๆ ในไฟล์จะทำได้นั้นมีเพียง 2 กรณีคือผู้ที่ทำการเปลี่ยนแปลงนั้นเป็นเจ้าของไฟล์หรือเป็นผู้ดูแลระบบ การเปลี่ยนแปลงสิทธิต่างๆ ของไฟล์สามารถกระทำได้โดยใช้คำสั่ง chmod ตามด้วยสิทธิที่ต้องการให้เป็นและต่อท้ายด้วยชื่อไฟล์ที่ต้องการ

การเปลี่ยนแปลงสิทธิโดยให้ผู้ใช้ทุกกลุ่มนั้นสามารถใช้คำสั่ง chmod ตามด้วยเลขฐานแปดหรือกลุ่มตัวอักษรก็ได้ดังตารางที่ 4.5

กลุ่มผู้ใช้	เครื่องหมายการจัดการ	รูปแบบของสิทธิ์
-u (เจ้าของไฟล์)	+ (เพิ่มสิทธิ์)	-r(การอ่าน)
-g (บุคคลในกลุ่มเจ้าของไฟล์)	= (กำหนดสิทธิ์)	-w(การเขียน)
-o(บุคลทั่วไป)	- (ลดสิทธิ์)	-x(การเรียกใช้หรือค้นหา)
-a(ทุกคนทุกกลุ่ม)		

ตารางที่ 2-5 การเปลี่ยนแปลงสิทธิโดยใช้ตัวอักษร

และการกำหนดสิทธิในส่วนของกลุ่มผู้ใช้งานและรูปแบบของสิทธินั้น เราสามารถกำหนดได้มากกว่าหนึ่งกลุ่มได้ เช่น g+w จะหมายถึงกำหนดเพิ่มให้บุคคลในกลุ่มเจ้าของไฟล์สามารถแก้ไขไฟล์ได้

```
[arnut@database arnut]$ Is - I test_sql
-rw-r- - r- - 1 arnut lecturer 0 Oct 22 17:58 test_sql1
[arnut@database arnut]$ chmod g+w test_sql
[arnut@database arnut]$ Is - I test_sql
-rw-rw-r- - 1 arnut lecturer 0 Oct 22 17:58 test_sql
[arnut@database arnut]$
```

นอกจากนี้ยังสามารถเปลี่ยนผู้เป็นเจ้าของไฟล์ และกลุ่มของเจ้าของไฟล์ได้เช่นกัน โดยใช้คำสั่ง chown (change owner) และ chgrp (change group)

- **การกำหนดสิทธิต่างๆ ในไฟล์โดยอัตโนมัติ**

ในเชลล์จะมีคำสั่ง umask ไว้กำหนดค่าสิทธิต่างๆ ในไฟล์โดยอัตโนมัติเก็บไว้ที่ .profile สำหรับบอร์นเชลล์และคอร์นเชลล์ หรือ .chrc สำหรับซีเซลล์ ชื่อไฟล์ทั้งสองนี้เป็นไฟล์เริ่มต้นในการใช้งานทุกครั้ง

การกำหนดเช่นนี้เพื่อที่จะได้ไม่ต้องทำการกำหนดสิทธิหรือเปลี่ยนแปลงสิทธิของไฟล์ทุกครั้งที่มีการสร้างขึ้นใหม่ โดยรูปแบบของคำสั่ง umask จะตามด้วยตัวเลขฐานแปดจำนวน 3 หลัก ซึ่งตัวเลขจะแทนสิทธิต่างๆ ในทางตรงข้ามกับการกำหนดสิทธิของไฟล์ ดังตารางที่ 2 – 6

โดยทั่วไปแล้ว ค่าของ umask จะถูกกำหนดให้เป็น 022 คือให้เจ้าของไฟล์นั้นมีสิทธิทุกอย่างในการดำเนินการกับไฟล์ ส่วนสมาชิกในกลุ่มและผู้ใช้อื่นนั้นกำหนดให้เพียงสามารถอ่านและเอ็กซีคิวต์ไฟล์ได้เท่านั้น

umask	**สิทธิของไฟล์**	**สิทธิของไดเรกทอรี**
077	600(rw - - - - - - -)	700 (rwx - - - - -)
067	600 (rw - - - - - - -)	710 (rwx - -x - - -)
066	600 (rw - - - - - - -)	711 (rwx - -x - - x)
027	640 (rw - r - - - - -)	750 (rwxr - x - - -)
026	640 (rw - r - - - - -)	751 (rwxr - x - - x)
022	644 (rw - r - - r - -)	755 (rwxr - x r - x)
000	666 (rw - r w - r w -)	777 (rwxrwxrwx)

ตารางที่ 2-6 การกำหนดสิทธิของไฟล์และไดเรกทอรีโดยใช้ umask

2.9 ไฟล์สำคัญในการใช้งาน

2.9.1 ไฟล์ที่มองไม่เห็น (Hidden Files)

ในระบบปฏิบัติการลีนุกซ์ ชื่อไฟล์ และไดเร็กทอรีที่ขึ้นต้นด้วยจุด (.) ระบบจะถือว่าเป็นไฟล์ชนิดพิเศษโดยจะไม่สามารถมองเห็นได้จากมุมมองปกติ เป็น ไฟล์ซ่อน (Hidden Files) จะต้องทำการกำหนดออปชั่นพิเศษเพื่อทำการแสดงรายชื่อไฟล์ เช่น ls -a

2.9.2 ไฟล์สำคัญในระดับผู้ใช้

เมื่อผู้ใช้แต่ละคนล็อกอินเพื่อเข้าใช้ระบบ จะมีไฟล์สำคัญหลายๆ ไฟล์ที่เกี่ยวข้องกับการทำงานของผู้ใช้แต่ละคน โดยไฟล์เหล่านี้จะมีอยู่ในโฮมไดเรกทอรีของผู้ใช้แต่ละคน เพื่อเหตุผลในการทำงานต่างๆ ซึ่งมีรายการดังนี้

ไฟล์ .bash_history	ทำหน้าที่เก็บรายการคำสั่ง ที่ผู้ใช้ทำการเรียกใช้เอาไว้ ทำให้สามารถเรียกใช้คำสั่งเก่าได้
ไฟล์ .bashrc	จะถูกประมวลผลทุกครั้ง ที่ผู้ใช้ทำการสร้างเชลล์ขึ้นมาใหม่หรือเรียกใช้คำสั่งที่ไม่ใช่คำสั่งภายในของเชลล์
ไฟล์ .bash_profile	จะถูกประมวลผลทุกครั้ง เมื่อผู้ใช้ทำการ login เพื่อเข้าใช้ระบบ
ไฟล์ .bash_logout	จะถูกประมวลผลทุกครั้ง เมื่อผู้ใช้ทำการ logout เพื่อออกจากระบบ

ตารางที่ 2-7 ไฟล์สำคัญในระดับผู้ใช้

- ไฟล์สำคัญระดับการใช้งาน

ไฟล์ /etc/lilo.conf	เป็นไฟล์ที่เก็บค่าคอนฟิกูเรชั่นของโปรแกรม Lolo (Linux Loader)
ไฟล์ /etc/inittab	เป็นไฟล์ที่เก็บค่าคอนฟิกูเรชั่นของการเริ่มงานระบบ
ไฟล์ /etc/profile	เป็นไฟล์ที่เก็บคำสั่งเริ่มต้นของระบบที่ต้องการสั่งให้ทำงานตอนบู๊ตระบบ
ไฟล์ /etc/fstab	เป็นไฟล์ที่เก็บค่าคอนฟิกูเรชั่นของการเชื่อมต่อระบบไฟล์ในขณะบู๊ตระบบ
ไฟล์ /tec/X11/XF86Config-4	เป็นไฟล์ที่เก็บค่าคอนฟิกูเรชั่นของระบบ X Windows

ตารางที่ 2-8 ไฟล์สำคัญระดับใช้งาน

2.10 สรุป

ในบทนี้เราก็ได้เรียนรู้เกี่ยวกับระบบไฟล์ของระบบปฏิบัติการลีนุกซ์ ไม่ว่าจะเป็นระบบไดเรกทอรีทั้งหมดที่มีในลีนุกซ์, ประเภทของระบบไฟล์, การจัดการกับระบบไฟล์ รวมทั้งการกำหนดสิทธิการเข้าใช้งานไฟล์ ซึ่งถือเป็นเรื่องสำคัญสำหรับผู้ที่ต้องการศึกษาระบบปฏิบัติการลีนุกซ์ ต้องเรียนรู้การใช้งานระบบไฟล์อย่างถูกต้องเพื่อความเสถียรภาพของระบบและนำไปใช้ในส่วนอื่นๆ