เอกสารประกอบการสอน

วิชา ระบบปฏิบัติการ 2 (Operating Systems 2) รหัส 4121402
บทที่ 4 การจัดการเกี่ยวกับไฟล์ (File Management)
หลักสูตรระดับปริญญาตรี
พุทธศักราช 2551 (ปรับปรุง 2554)

โดย

จุฑาวุฒิ จันทรมาลี

สาขาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสวนดุสิต

บทที่ 4
การจัดการเกี่ยวกับไฟล์
(File Management)

สำหรับผู้ใช้งาน การจัดการเกี่ยวกับไฟล์ (File Management) จะเกี่ยวข้องกับหน้าที่ของระบบปฏิบัติการโดยตรงในการสนับสนุนการจัดเก็บและการัดการไฟล์ ซึ่งระบบไฟล์ (File System) อาจจะประกอบไปด้วย ข้อมูล (Data) โครงสร้างไดเร็กทรอรี (Directory Structure) การแบ่งไฟล์ (Partitions) การป้องกันไฟล์ (File Protection) ร่วมถึงการใช้ไฟล์ร่วมกัน (Share File) ซึ่งจะแสดงได้จากหัวข้อต่างๆ ดังนี้

4.1 แนวคิดเกี่ยวกับไฟล์ (File Concept)

ไฟล์ (File) คือ กลุ่มข้อมูลหรือสารสนเทศที่มีความสัมพันธ์กันและถูกจัดเก็บไว้บนหน่วยความจำรอง (Secondary Storage) เพราะข้อมูลไม่สามารถจัดเก็บลงบนหน่วยความจำได้โดยตรงจะต้องจัดเก็บไว้ในรูปแบบของไฟล์ โดยทั่วไปไฟล์จะจัดเก็บโดยลำดับของบิต (Bits) ไบต์ (Bytes) หรือ เรคคอร์ด (Recodes) โดยผู้ใช้เป็นผู้สร้างขึ้น แบ่งออกเป็น 2 ประเภท เช่น

1.ไฟล์โปรแกรม (Program File) เช่น ไฟล์ข้อความ (Text File) ไฟล์เฉพาะ (Specific File) ไฟล์สำหรับประมวลผล (Execute File) ไฟล์ไลบรารี (Library File)

2.ไฟล์ข้อมูล (Data File) เช่น ตัวเลข (Numeric) ตัวอักษร (Alphabetic) สัญลักษณ์ (Alphanumeric) หรือเลขฐานสอง (Binary)

การจัดเก็บไฟล์จะมีรูปแบบที่แตกต่างกัน ขึ้นอยู่กับผู้ใช้เป็นผู้สร้างขึ้น โดยไฟล์แต่ละประเภทจะมีการกำหนดโครงสร้าง (Structure) ที่แน่นอน โดยใช้วิธีการจัดเก็บทางตรรกะ (Logical Secondary Storage) หรือวิธีการจัดเก็บข้อมูลที่มีความสัมพันธ์กันเก็บอยู่ในที่เดียวกัน เช่น

- ไฟล์ข้อความ (Text File) เป็นไฟล์ที่เก็บรวบรวมกลุ่มของข้อความที่ประกอบขึ้นจากหลายบรรทัดเอาไว้ที่เดียวกัน
- ไฟล์โปรแกรม (Source File) เป็นไฟล์ที่เก็บรวบโปรแกรมย่อย (Subroutines) และฟังก์ชันการทำงานเอาไว้ที่เดียวกัน
- ไฟล์วัตถุ (Object File) เป็นไฟล์ที่เก็บรวบรวมกลุ่มข้อมูลของไบต์ (Bytes) และจัดเก็บไว้ในรูปบล็อก (Blocks) เอาไว้ที่เดียวกัน
- ไฟล์ประมวลผล (Executable File) เป็นไฟล์ที่เก็บรวบรวมชุดคำสั่งเพื่อที่จะสามารถโหลดเข้าไปประมวลผลยังหน่วยความจำได้

- **คุณลักษณะของไฟล์ (File Attributes)**

1.ชื่อไฟล์ (Name) เป็นกลุ่มของตัวอักษรหรือสัญลักษณ์ที่มนุษย์สร้างขึ้นเพื่อใช้สื่อความหมายต่างๆ และแสดงคุณสมบัติของไฟล์นั้น

2. ลักษณะเฉพาะของไฟล์ (Identifier) เป็นแท็กเฉพาะ (Unique Tag) ใช้ตัวเลขระบุโดยรวมเข้า

กับไฟล์ภายในระบบไฟล์ (File System)

3. ประเภทของไฟล์ (Type) เป็นข้อมูลเพื่อใช้ระบุว่าเป็นไฟล์ชนิดอะไร โดยที่ระบบปฏิบัติการเป็นตัวจัดการเพื่อรองรับความแตกต่างของไฟล์แต่ละชนิดกัน

4. ตำแหน่ง (Location) เป็นข้อมูลที่ใช้อ้างอิงเครื่องมือและตำแหน่งของไฟล์ที่ใช้จัดเก็บบนอุปกรณ์ต่างๆ

5. ขนาด (Size) เป็นขนาดปัจจุบันของไฟล์ที่จัดเก็บอยู่ เช่น ไบต์ (Bytes) เวิร์ด (Words) หรือ บล็อค (Block) เป็นต้น โดยขนาดสูงสุดที่จัดเก็บได้จะระบุไว้ในคุณลักษณะของไฟล์

6. การป้องกัน (Protection) เป็นการควบคุมการเข้าถึง (Access-control) ข้อมูล การกำหนดสิทธิในการอ่าน (Reading) การเขียน (Writing) การประมวลผล (Executing) เป็นต้น

7. เวลา (Time) วันที่ (Date) และการระบุคุณลักษณะผู้ใช้ (User Identification) ซึ่งข้อมูลเหล่านี้จะใช้เพื่อระบุ ควบคุม ป้องกันและรักษาความปลอดภัยในการใช้งานได้ เช่น การสร้าง (Creation) การเปลี่ยนแปลงครั้งล่าสุด (Last Modify) การใช้งานครั้งล่าสุด (Last Use) เป็นต้น

- **การจัดการเกี่ยวกับไฟล์ (File Operating)**

ระบบปฏิบัติการทุกประเภทมีคำสั่งพื้นฐานที่ใช้เพื่อใช้จัดการเกี่ยวกับไฟล์ผ่านคำสั่งใน System Call พื้นฐาน 6 คำสั่งดังนี้

1. การสร้างไฟล์ (Creating a file) ประกอบด้วย 2 ขั้นตอน คือ

1.1 หาที่ว่างสำหรับสร้างไฟล์

1.2 ระบุไฟล์ที่สร้างขึ้นมาใหม่ไว้ในไดเร็กทรอรี ที่บรรจุ ชื่อไฟล์ ตำแหน่ง หรือข้อมูลด้านอื่นๆ

2. การเขียนไฟล์ (Writing a file) เป็นการใช้คำสั่งใน System Call ในการเขียนชื่อและข้อมูลต่างๆ ของไฟล์ โดยระบบปฏิบัติการจะทำการค้นหาไดเร็กทรอรีที่ระบุตำแหน่งไฟล์ โดยระบบจะเก็บพอยเตอร์ (Pointer) สำหรับระบุตำแหน่งที่ต้องการเขียนลงบนไฟล์ และมีการปรับปรุงตำแหน่งพอยเตอร์ใหม่ทุกครั้งเมื่อมีการเขียนไฟล์เกิดขึ้น

3. การอ่านไฟล์ (Reading a file) เป็นการใช้คำสั่งใน System Call ในการระบุชื่อและที่อยู่เพื่อเพื่อที่จะค้นหาไฟล์ที่ต้องการจากไดเร็กทรอรี และทำการจัดเก็บพอยเตอร์ของไฟล์ที่ถูกอ่านหรือเขียนไว้ในพอยเตอร์ ณ ตำแหน่งไฟล์ปัจจุบัน (Current-File-Position Pointer) โดยไฟล์ที่ถูกอ่านหรือเขียนจะจัดเก็บไว้เพียงพอยเตอร์เดียวเท่านั้น ซึ่งทำให้ช่วยประหยัดพื้นที่และลดความซ้ำซ้อนของพอยเตอร์ในหน่วยความจำ

4. การย้ายตำแหน่งภายในไฟล์ (Repositioning Within a File) เป็นการค้นหาไฟล์ในไดเร็กทรอรี และกำหนดค่าพอยเตอร์ให้ชี้ไปยังตำแหน่งไฟล์ปัจจุบัน (Current-File-Position) เพื่อย้ายไฟล์ไปยังตำแหน่งใหม่ที่ต้องการ เราเรียกวิธีการแบบนี้ว่า การค้นหา (Seek)

5. การลบไฟล์ (Deleting a file) เป็นการใช้คำสั่งใน System Call ในการลบไฟล์ที่ต้องการและคืนพื้นที่ว่างที่เกิดหลังจากการลบไฟล์ให้กับระบบปฏิบัติการเพื่อดำเนินการเพื่อจัดสรรพื้นที่ว่างให้กับไฟล์อื่นต่อไป

6. การตัดทอนไฟล์ (Truncating a file) ผู้ใช้อาจจะต้องการลบเนื้อหาบางส่วนของไฟล์ออกแต่ยังคงเก็บคุณลักษณะของไฟล์เดิมไว้ โดยฟังก์ชั่นนี้จะทำให้คุณสมบัติเดิมของไฟล์ไม่เปลี่ยนแปลง ยกเว้นความยาวของไฟล์ นอกจาก 6 พื้นฐานในการดำเนินการเกี่ยวกับไฟล์แล้ว ยังมีวิธีการดำเนินการอื่นอีกในการจัดการกับไฟล์ เช่น การต่อท้าย (Appending) การเปลี่ยนชื่อไฟล์ (Renaming) และการคัดลอกไฟล์ (Copy) เป็นต้น

การดำเนินการค้นหาไฟล์ในไดเร็กทรอรีเป็นหน้าที่ของระบบปฏิบัติการ โดยมีการเรียกใช้คำสั่งใน System Call ในการเปิด (Open) ไฟล์เพื่อทำการค้นหาไฟล์ในตำแหน่งที่ระบุชื่อไฟล์นั้นไว้ ในขณะเดียวกันก็จะมีการสร้างตารางสำหรับจัดเก็บข้อมูลเกี่ยวกับการเปิดไฟล์ทั้งหมดไว้ใน Open File Table เพื่อใช้เป็นดัชนีให้กับตัวดำเนินการเกี่ยวกับไฟล์ในการค้นหาไฟล์ในครั้งต่อไป ทำให้ไม่ต้องเสียเวลาค้นหาไฟล์ในไดเร็กทรอรีอีก และในกรณีที่ไม่ต้องการใช้งานไฟล์นั้นแล้ว ระบบปฏิบัติการก็จะทำการปิด (Close) ไฟล์นั้นทันทีแล้วนำข้อมูลออกจาก Open File Table โดยการเปิดไฟล์ทุกครั้งจะเกี่ยวข้องกับไฟล์ต่อไปนี้

- ตัวชี้ตำแหน่งไฟล์ (File Pointer) ใช้สำหรับชี้ไปยังตำแหน่งที่ถูกอ่านหรือเขียนเป็นครั้งสุดท้าย
- ตัวนับการเปิดไฟล์ (File Open Count) ใช้สำหรับนับจำนวนครั้งของการเปิดไฟล์
- ตำแหน่งของไฟล์บนดิสก์ (Disk location of the File) ใช้ระบุตำแหน่งไฟล์บนดิสก์เมื่อมีการเปลี่ยนแปลงข้อมูลภายในไฟล์
- การให้สิทธิในการเข้าถึง (Access rights) แต่ละโปรเซสที่ต้องการการเข้าถึงเพื่อเปิดไฟล์ ระบบปฏิบัติการเป็นตัวจัดการว่าจะอนุญาตหรือปฏิเสธในการเข้าถึงไฟล์หรืออุปกรณ์ I/O ได้

- **ประเภทของไฟล์ (File Type)**

  โดยทั่วไปแล้วทุกๆ ระบบปฏิบัติการควรจะรู้จำ (Recognize) และสนับสนุนประเภทของไฟล์หลังจากมีการสร้างปละจัดเก็บไฟล์เกิดขึ้นจำเป็นต้องมีการกำหนดประเภทของไฟล์ ซึ่งแบ่งออกเป็น 2 ส่วนคือ ชื่อ (Name) และนามสกุล (Extention) และทั้งสองส่วนจะมี “ . ” (Period) ขั้นระหว่างกัน เช่น work.pdf, os.doc, test.txt เป็นต้น

ตารางที่ 4.1 แสดงประเภทของไฟล์

ประเภทของไฟล์ (File Type)	นามสกุล (Extention)	หน้าที่ (Function)
executable	exe, com, bin or none	ไฟล์พร้อมที่จะถูกประมวลผลโดยภาษาเครื่อง
object	obi, o	ไฟล์ที่คอมไพล์เป็นภาษาเครื่องแล้วแต่ยังไม่พร้อมที่จะประมวลผล (not link)
source code	c, cc, java, pas, asm, a	ซอร์ซโค้ดของภาษาโปรแกรมต่างๆ
batch	bat, sh	ไฟล์ชุดคำสั่ง (command)

ประเภทของไฟล์ (File Type)	นามสกุล (Extention)	หน้าที่ (Function)
text	txt, doc	ไฟล์ที่เป็นข้อความหรือเอกสารทั่วไป
word processor	wp, tex, rtf, doc	ไฟล์ที่ได้จากการประมวลผลข้อความรูปแบบต่างๆ
library	lib, a, sof, dll	ไฟล์ไลบรารีหรือโปรแกรมย่อย
print of view	ps, pdf, jpg	ไฟล์ที่จัดเก็บข้อมูลในรูปแบบเอสกี (ASCII) หรือรูปแบบไบนารีสำหรับพิมพ์หรือแสดงผล
archive	arc, zip, tar	ไฟล์ที่รวบรวมไฟล์ทั้งหมดให้เป็นหนึ่งไฟล์หรือไฟล์ที่เกิดจากการบีบอัด (compressed) เพื่อประหยัดพื้นที่ในการจัดเก็บ
multimedia	mpeg, mov, rm	ไฟล์ไบนารีที่ภายในสามารถบรรจุเสียงหรือภาพ (A/V information)

- **โครงสร้างของไฟล์ (File Structure)**

ชนิดของไฟล์อาจจะใช้แสดงถึงโครงสร้างที่อยู่ภายในไฟล์ ซึ่งระบบปฏิบัติการจะสามารถประมวลผลไฟล์ได้ก็ต่อเมื่อเข้าใจโครงสร้างของไฟล์ก่อน เป็นการกำหนดว่าระบบปฏิบัติการจะทำการโหลดไฟล์ไปเก็บไว้ที่ตำแหน่งใดและทำการตรวจสอบว่าคำสั่งเริ่มต้นอยู่ ณ ตำแหน่งใดในหน่วยความจำหลัก ดังนั้นระบบปฏิบัติการโดยทั่วไปจะต้องสนับสนุนโครงสร้างของไฟล์อย่างน้อยหนึ่งโครงสร้าง คือ ไฟล์สำหรับประมวลผล (Executable File) ซึ่งเป็นไฟล์ที่ช่วยให้ระบบปฏิบัติการสามารถโหลด (Load) และประมวลผลโปรแกรม (Run Programs) ในตำแหน่งที่ถูกต้องได้

ระบบปฏิบัติการแมคอินทอช (Macintosh) สนับสนุนโครงสร้างไฟล์ที่มีขนาดเล็กได้ โดยแบ่งโครงสร้างของไฟล์ออกเป็น 2 ส่วน คือ

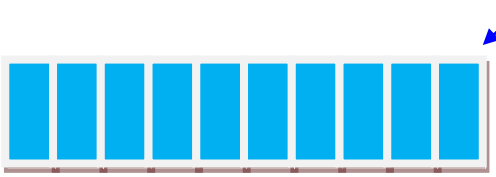
1.ส่วนของทรัพยากร (Resource Fork) ใช้บรรจุสารสนเทศ (Information) ที่ผู้ใช้ต้องการ

2.ส่วนข้อมูล (Data Fork) ใช้บรรจุโค้ดโปรแกรม (Program Code) และข้อมูล (Data)

การจัดโครงสร้างไฟล์ที่ใช้กันโดยทั่วไปมี 3 วิธี

1. แบบไบต์เรียงต่อกัน

มีการเก็บเป็นไบต์เรียงต่อ ๆ กันไป ดังเช่นในระบบปฏิบัติการของ UNIX และ Windows การเก็บไฟล์ในลักษณะนี้เป็นแบบที่ไม่มีโครงสร้างในการจัดเก็บ ไฟล์ที่ถูกสร้างใหม่จะถูกนำมาเรียงต่อกันไปเรื่อย ๆ จนเต็มเนื้อที่ โดยที่ตัวระบบปฏิบัติการแทบจะไม่ทำหน้าที่อะไรเลย แสดงดังรูปที่ 4.1



รูปที่ 4.1 แสดงไฟล์ที่ถูกจัดเก็บแบบเรียงลำดับ

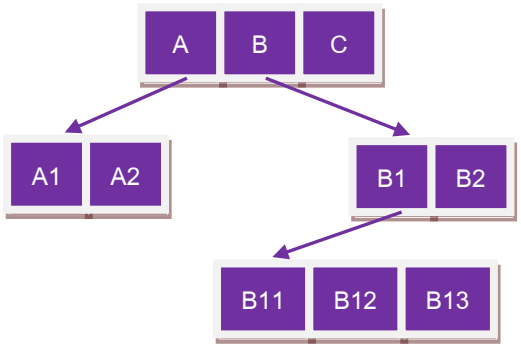
2. แบบเรกคอร์ดเรียงต่อกัน

โดยมีขนาดของเรกคอร์ดคงที่ ในแต่ละไฟล์จะถูกจัดเก็บอยู่ในรูปของเรกคอร์ดจัดเรียงกันไปตามลำดับจนถึงเรกคอร์ดสุดท้าย ซึ่งในเรกคอร์ดสุดท้ายอาจจะไม่เต็มเรกคอร์ดก็ได้ ในการอ่านและเขียนจะทำไปทีละเรกคอร์ด ในบางระบบอาจจะกำหนดให้แต่ละเรกคอร์ดมีขนาดเท่ากับ 80 อักษร ซึ่งเท่ากับ 1 บรรทัดพอดี เช่นในระบบปฏิบัติการ CP/M แสดงดังรูปที่ 4.2

รูปที่ 4.2 แสดงไฟล์ที่ถูกจัดเก็บแบบเรกคอร์ด

3. แบบต้นไม้

แต่ละบล็อกจะประกอบไปด้วยเรกคอร์ด โดยมีขนาดของเรกคอร์ดขึ้นอยู่กับปัจจัยหลายประการ เช่น ขนาดของไฟล์ข้อมูล เวลาที่ใช้ในการเข้าถึงข้อมูล (Access time) เป็นต้น วิธีนี้ใช้ในระบบปฏิบัติการหลายเครื่องด้วยกัน แสดงดังรูปที่ 4.3



รูปที่ 4.3 แสดงไฟล์ที่ถูกจัดเก็บแบบต้นไม้

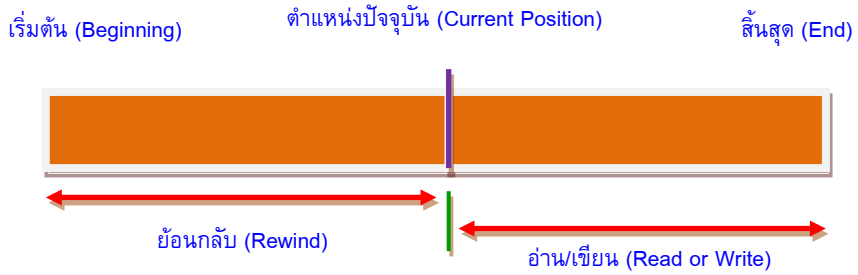
- **โครงสร้างภายในไฟล์ (Internal File Structure)**

ซึ่งแต่ละระบบปฏิบัติจะมีโครงสร้างภายในไฟล์ที่ซับซ้อนแตกต่างกันไป การจัดเก็บไฟล์ลงบนพื้นที่ว่างในดิสก์ขนาดของบล็อก (Block size) จะถูกกำหนดโดยขนาดของเซกเตอร์ (Sector) โดยที่ขนาดความยาวของการบันทึกทางตรรกะ (Logical Recode) จะต้องมีขนาดความยาวเท่ากับของการบันทึกทางกายภาพ (Physical Record) โดยส่วนใหญ่แล้วจำนวนขนาดความยาวของการบันทึกทางตรรกะ (Logical Recode) มักจะจัดเก็บเป็นชุด (Packing) และสัมพันธ์กับบล็อกในของขนาดความยาวทางกายภาพด้วย (Physical Block) ซึ่งชุด (Packing) ข้อมูลทางกายภาพอาจจะถูกกำหนดโดยผู้ใช้งานโปรแกรมหรือระบบปฏิบัติการก็ได้

4.2 วิธีการเข้าถึงไฟล์ (File Access Methods)

การจัดเก็บไฟล์ข้อมูล เมื่อมีการเรียกใช้จะต้องมีการเข้าถึงและอ่านไฟล์นั้นเข้ามาเก็บยังหน่วยความจำหลัก ซึ่งข้อมูลที่อยู่ในไฟล์สามารถเข้าถึงได้หลายวิธี ดังนั้นระบบปฏิบัติการจะทำการเลือกวิธีที่เหมาะสมเพียงหนึ่งวิธีเท่านั้นในการเข้าถึงไฟล์และเลือกโปรเซสที่เหมาะสมเพื่อใช้ในการแก้ปัญหาการเข้าถึงไฟล์ได้ดังนี้

- **การเข้าถึงแบบเรียงลำดับ (Sequential Access)** เป็นการอ่านและเขียนไฟล์แบบเรียงลำดับ โดยการอ่านจะอ่านส่วนถัดไปของไฟล์และทำการปรับเปลี่ยนค่าพอยเตอร์โดยอัตโนมัติตามตำแหน่ง (Track) ของอุปกรณ์ I/O นั้น โดยวิธีนี้คล้ายกับการเขียนไปยังท้ายไฟล์และสามารถย้อยกลับไปยังตำแหน่งเริ่มต้นของไฟล์ได้ ดังนั้น การเข้าถึงไฟล์สามารถที่จะย้อนกลับไปมา (Forward or Backward) ระหว่างจุดเริ่มต้นและจุดสิ้นสุดได้ แสดงดังรูปที่ 4.4 และการเข้าถึงแบบลำดับขึ้นอยู่กับรูปแบบการจัดเก็บไฟล์ลงเทปซึ่งทำงานได้ดีในอุปกรณ์ที่เข้าถึงแบบเรียงลำดับมากกว่าการเข้าถึงแบบสุ่ม



รูปที่ 4.4 การเข้าถึงแบบเรียงลำดับ (Sequential Access)

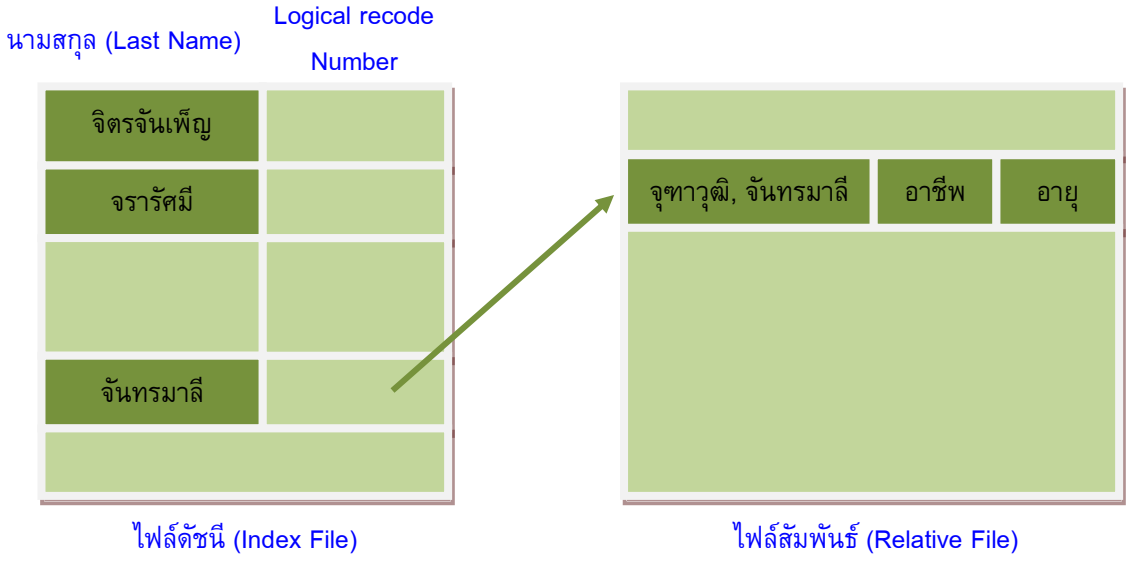
- **การเข้าถึงแบบทางตรง (Directl Access)** เป็นการเข้าถึงไฟล์ที่จัดเก็บในรูปแบบของเรกคอร์ดทางตรรกะ (Logical Records) ที่มีขนาดความยาวคงที่ทำให้โปรแกรมสามารถอ่านหรือเขียนเรกคอร์ดได้อย่างรวดเร็วเพราะไม่ต้องสนใจลำดับในการเข้าถึง อาจจะเรียกวิธีการเข้าถึงแบบนี้ว่า “ การเข้าถึงแบบสัมพันธ์ (Lelative Access) ” เป็นลักษณะพื้นฐานของการทำงานบนดิสก์ซึ่งอนุญาตให้เข้าถึงไฟล์ในบล็อกแบบสุ่ม (Random) ได้ ดังนั้นการเข้าถึงแบบทางตรงระบบปฏิบัติการจะยอมให้อ่านหรือเขียนไฟล์ ณ ตำแหน่งใดก็ได้ในบล็อกโดยไม่มีข้อจำกัดมาควบคุมการเข้าถึง ทำให้การเข้าถึงแบบทางตรงสามารถค้นหาและดำเนินการต่างๆ บนไฟล์เร็วกว่าการเข้าถึงแบบลำดับโดยที่ไฟล์ทั้งหมดจะมีการปรับปรุงข้อมูลอยู่เสมอ โดยมีการรวมหมายเลขบล็อกเข้าไว้ด้วยกัน เพื่อใช้เป็นพารามิเตอร์ในการถึงไฟล์ โดยผู้ใช้เป็นผู้กำหนดหมายเลขบล็อกที่สัมพันธ์กัน (Relative Block Number) ให้กับระบบปฏิบัติการ ซึ่งใช้เป็นดัชนีที่สัมพันธ์กับจุดเริ่มต้นของไฟล์ ซึ่งมีการเรียงลำดับเป็น 0, 1, 2, …. โดยระบบปฏิบัติการจะเป็น

ตัวตัดสินใจว่าจะนำไฟล์ไปไว้ ณ ตำแหน่งใดที่สัมพันธ์กับหมายเลขบล็อก (มักจะเริ่มที่หมายเลขบล็อกที่ 0 หรือ บล็อกที่ 1) ร่วมถึงสามารถป้องกันไม่ให้ผู้ที่ไม่ได้รับสิทธิเข้ามาใช้งานไฟล์

ตารางที่ 4.2 แสดงการจำลองสถานการณ์ของการเข้าถึงแบบลำดับให้เป็นการเข้าถึงแบบทางตรง

การเข้าถึงแบบลำดับ (Sequential access)	การนำไปใช้งานกับการเข้าถึงแบบทางตรง (Implement for durectl access)
reset	cp = 0;
read next	read cp; cp = cp + 1;
write next	write cp; cp = cp + 1;

- **การเข้าถึงแบบดัชนี (Index and Relative Files)** เป็นวิธีการใช้ดัชนี (Index) ในเข้าถึงไฟล์ โดยภายในจะบรรจุพอยเตอร์และจัดเก็บไว้ในบล็อกที่แตกต่างเพื่อใช้ค้นหาเรกคอร์ดที่อยู่ภายในไฟล์ โดยลำดับแรกจะทำการค้นหาดัชนีก่อน หลังจากนั้นจะใช้พอยเตอร์ในการเข้าถึงไฟล์โดยตรงเพื่อทำการค้นหาเรกคอร์ดที่ต้องการได้ แสดงดังรูปที่ 4.5



รูปที่ 4.5 ตัวอย่างการใช้ดัชนีและความสัมพันธ์ของไฟล์
(Example of Index and Relative Files Access)

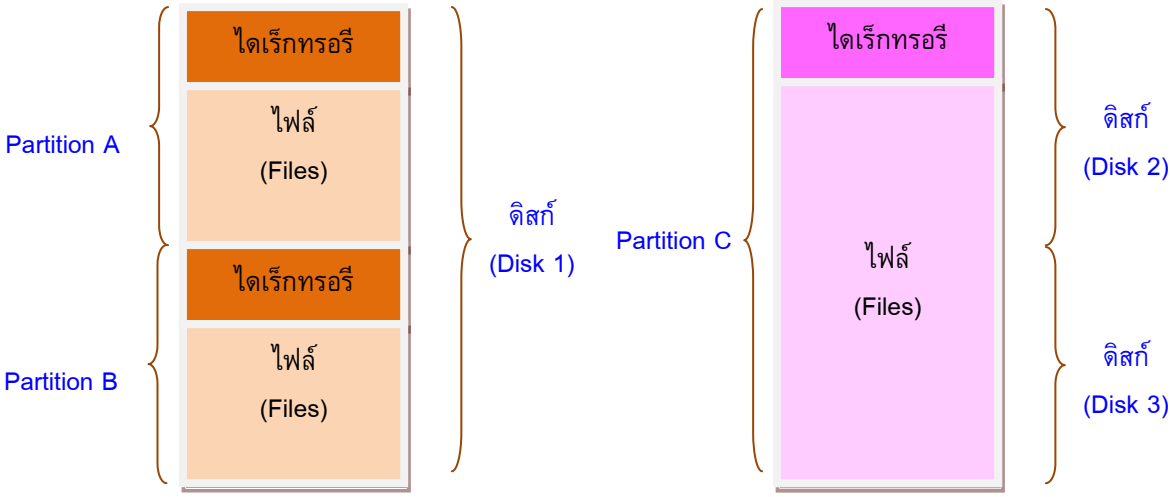
4.3 โครงสร้างไดเรกทรอรี (Directory Structure)

การจัดเก็บไฟล์ลงบนของคอมพิวเตอร์ เป็นหน้าที่ขอระบบปฏิบัติการ ในการจัดเก็บลงบนพื้นที่หน่วยความจำสำรอง (Seconder Memory) โดยที่รู้จักดีคือจัดเก็บลงบนพื้นที่ดิสก์ ซึ่งบางระบบอาจจะ

สามารถเก็บไฟล์ขนาดเทราไบต์ (Terabytes) ลงบนพื้นที่ดิสก์ได้ ดังนั้นการจัดเก็บไฟล์ทั้งหมดจำเป็นจะต้องจัดโครงสร้างไดเร็กทรอรี (Directory) ให้เหมาะสม กับพื้นที่ในดิสก์ โดยการจัดโครงสร้างของพื้นที่ดิสก์สามารถแบ่งออกเป็น 2 ส่วน คือ

- ส่วนที่ 1 ดิสก์จะถูกแบ่งออกเป็น Partition เดียวหรือมากกว่าหนึ่ง Partition ซึ่งบางครั้งอาจจะเรียกพื้นที่ๆ แบ่งออกมานี้ตามชนิดของเครื่องคอมพิวเตอร์ เช่น IBM เรียกว่า ดิสก์ขนาดเล็ก (Minidisks) เครื่อง PC เรียกว่า ความจุ (Volume) เครื่อง Macintosh เรียกว่า อาณาเขต (Arenas)
- ส่วนที่ 2 แต่ละ Partition จะบรรจุข้อมูลที่เกี่ยวข้องกับไฟล์ที่จัดเก็บไว้ภายใน โดยข้อมูลจะจัดเก็บไว้ในอุปกรณ์ไดเร็กทรอรี (Device Directory) หรือ ตารางแสดงรายการความจุ (Volume Table of Content) และมักจะเรียกชื่อสั้นๆ ว่า ไดเร็กทรอรี (Directory) โดยข้อมูลที่บันทึกอยู่ในไฟล์บน Partition จะประกอบด้วย ชื่อ ตำแหน่ง ขนาด และชนิด เป็นต้น

การจัดโครงสร้างของระบบไฟล์ที่แตกต่างกัน แสดงได้ดังรูปที่ 4.6



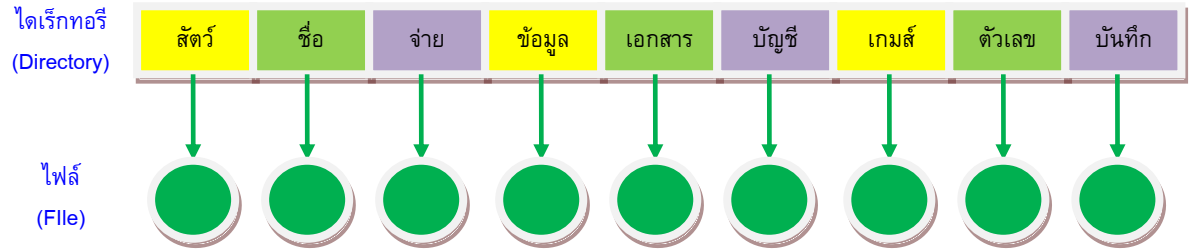
รูปที่ 4.6 การจัดโครงสร้างของระบบไฟล์โดยทั่วไป (A Typical File-System Organization)

การดำเนินการต่างๆ บนไดเร็กทรอรี

1. การค้นหาไฟล์ (Search for a File) เป็นการค้นหาไฟล์ ณ ตำแหน่งต่างๆ ที่อยู่ในไดเร็กทรอรี
2. การสร้างไฟล์ (Create a File) เป็นการสร้างไฟล์และมีการจัดเก็บลงในไดเร็กทรอรี
3. การลบไฟล์ (Delete a File) เป็นการลบไฟล์ออกจากไดเร็กทรอรี
4. การแสดงรายชื่อไฟล์ (List a Directory) เป็นการแสดงรายชื่อและรายละเอียดทั้งหมดของแต่ละไฟล์ในไดเร็กทรอรี
5. การเปลี่ยนชื่อไฟล์ (Rename a File) ชื่อไฟล์ต้องเปลี่ยนแปลงเพื่อให้สอดคล้องกับเนื้อหาและรายละเอียดของไฟล์เปลี่ยนไปด้วย
6. การท่องไปของระบบไฟล์ (Traverse The File System) เป็นเทคนิคการค้นหาหรือทำการคัดลอกไฟล์ไปยังตำแหน่งหรืออุปกรณ์ต่างๆ โดยสามารถเข้าถึงทุกๆ ไดเร็กทรอรี หรือทุกๆ ไฟล์ภายในโครงสร้างของไดเร็กทรอรีได้

การจำแนกลักษณะต่างๆ โครงสร้างทางตรรกะของไดเร็กทรอรี แบ่งออกได้ดังนี้

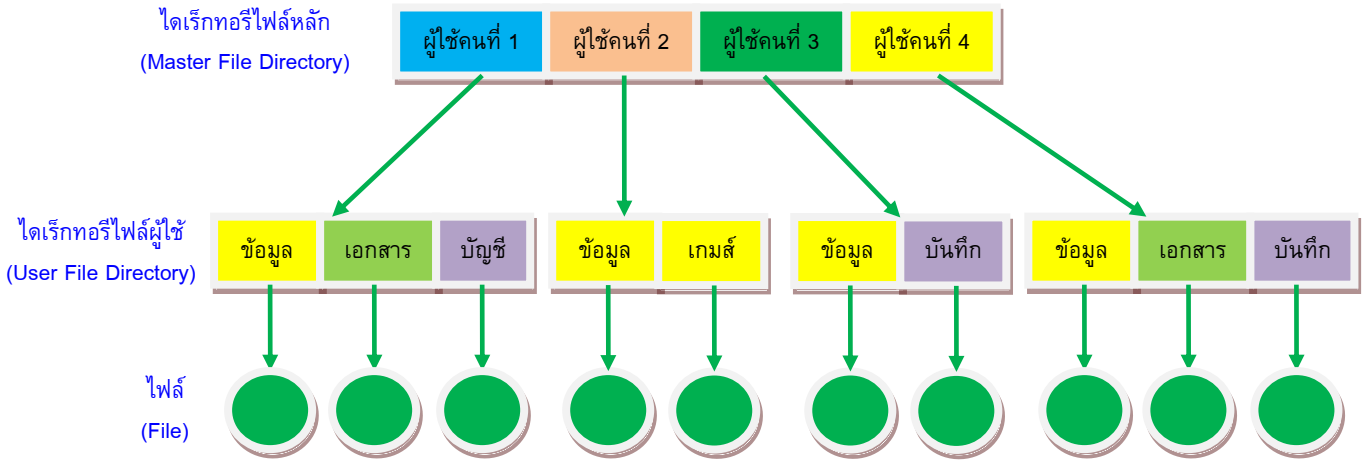
1. โครงสร้างแบบไดเร็กทรอรีเดียว (Single-Level Directory) เป็นโครงสร้างไดเร็กทรอรีที่ธรรมดาที่สุดเพราะทุก ๆ ไฟล์บรรจุอยู่ในไดเร็กทรอรีเดียวกัน สามารถจัดการและทำความเข้าใจได้ง่าย โดยโครงสร้างข้อมูลแบบนี้มีข้อจำกัดตรงที่เมื่อมีจำนวนไฟล์เพิ่มมากขึ้นหรือในระบบมีผู้ใช้จำนวนมากและไฟล์แต่ละประเภทต้องอยู่รวมกัน หากผู้ใช้สร้างไฟล์ที่มีชื่อเดียวกันอาจจะเกิดข้อผิดพลาดขึ้นเพราะไฟล์ใหม่ที่สร้างขึ้นไปทับไฟล์เก่าเดิมที่มีอยู่ อีกทั้งข้อจำกัดการกำหนดความยาวตัวอักษรในการตั้งชื่อไฟล์ของแต่ละระบบปฏิบัติด้วย เช่น MS-DOS อนุญาตให้ไม่เกิน 11 ตัวอักษร UNIX อนุญาตให้ไม่เกิน 255 ตัวอักษร โดยโครงสร้างแบบไดเร็กทรอรีเดียวแสดงดังรูปที่ 4.7



รูปที่ 4.7 แสดงโครงสร้างแบบไดเร็กทรอรีเดียว (Single-Level Directory)

2. โครงสร้างแบบสองไดเร็กทอรี (Two-Level Directory) เป็นโครงสร้างที่แบ่งไดเร็กทอรีให้กับผู้ใช้แต่ละคน ออกเป็น 2 ส่วน คือ

1. ไดเร็กทอรีของผู้ใช้ (User File Directory: UDF) ใช้เก็บไฟล์แต่ละคนของผู้ใช้
2. ไดเร็กทอรีหลัก (Master File Directory: MFD) ซึ่งจะมีดรรชนีที่ชี้ไปยังผู้ใช้ (User Name)หรือหมายเลขบัญชี (Account Name) ของผู้ใช้แต่ละคน โดยโครงสร้างแบบสองไดเร็กทอรีแสดงดังรูปที่ 4.8



รูปที่ 4.8 แสดงโครงสร้างแบบสองไดเร็กทอรี (Two-Level Directory)

3. โครงสร้างไดเร็กทอรีแบบต้นไม้ (Tree-Structure Directory) เป็นโครงสร้างที่แบ่งไดเร็กทอรีออกเป็นหลายระดับในลักษณะต้นไม้ที่แบ่งออกเป็นระดับชั้นและอนุญาตให้ผู้ใช้แต่ละคนสามารถที่จะสร้างไดเร็กทอรีย่อยและจัดการไฟล์แต่ละไฟล์ในภายในโครงสร้างไดเร็กทอรี โดยต้นไม้จะมีไดเร็กทอรีราก (Root) และทุกๆ ไฟล์ในระบบจะมีเส้นทางเฉพาะเป็นของตัวเอง (Unique Path Name) ซึ่งเชื่อมต่อไปยังไดเร็กทอรีย่อยที่เก็บไฟล์ที่ต้องการโดยเฉพาะ ไดเร็กทรอรีรากจะจัดเก็บไฟล์หรือไดเร็กทรอรีรากต่างๆ ที่มีโครงสร้างภายในเหมือนกัน และมีการจัดเก็บบิตเพื่อบอกสถานะ คือ บิต 0 กรณีเป็นไฟล์ หรือบิต 1 เป็นไดเร็กทอรีย่อย และใช้คำสั่ง System Call ในการสร้างและลบไดเร็กทอรี

ซึ่งโดยทั่วไปผู้ใช้แต่ละคนจะมี “ไดเร็กทอรีปัจจุบัน (Current Directory)” สำหรับจัดเก็บไฟล์ที่ผู้ใช้ต้องการใช้ในปัจจุบัน เมื่อผู้ใช้อ้างอิงถึงไฟล์ที่ต้องการ ระบบปฏิบัติการจะทำการค้นหาไฟล์ในไดเร็กทอรีในปัจจุบัน กรณีที่ไม่พบไฟล์ที่ต้องการ ผู้ใช้จะต้องมีการระบุชื่อของเส้นทาง (Path Name) ซึ่งแบ่งเป็น 2 ประเภท

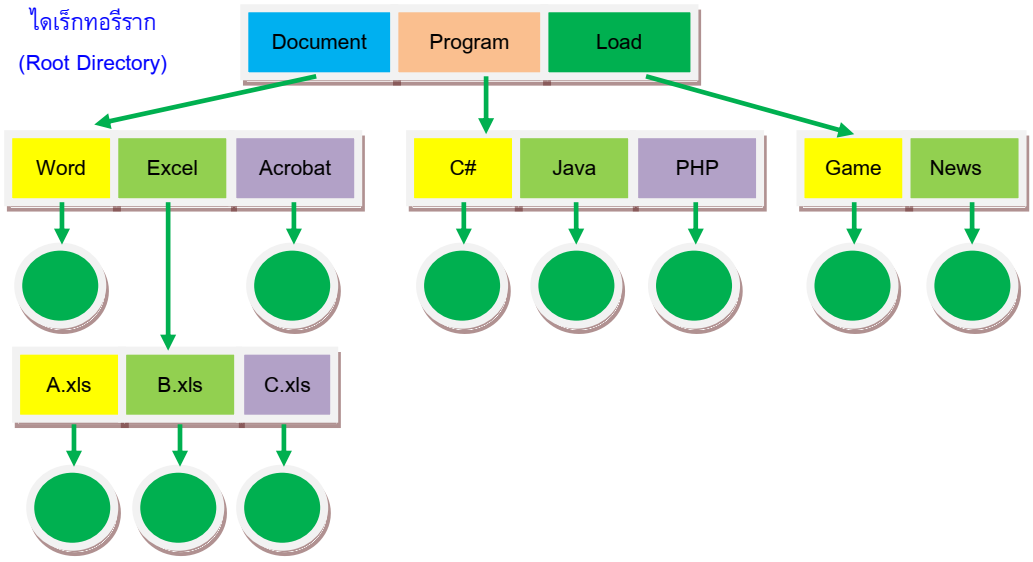
1.ชื่อเส้นทางแบบสมบูรณ์ (Absolute Path Name) เป็นเส้นทางการค้นหาไฟล์ที่ต้องการจากราก (Root) ลงมายังไฟล์ที่ต้องการ โดยมีการระบุชื่อไดเร็กทอรีที่ต้องการค้นหา

2.ชื่อเส้นทางแบบสัมพันธ์ (Relative Path Name) เป็นเส้นทางที่กำหนดจากไดเร็กทอรีปัจจุบัน

ตัวอย่าง ในโครงสร้างไดเร็กทอรีแบบทรี ในการจัดการระบบไฟล์ (โดยการอ้างอิงถึงชื่อไฟล์เดียวกัน) แสดงดังรูปที่ 4.9 ถ้าไดเร็กทอรีปัจจุบันคือ Root --> Document --> Excel จะได้ว่า

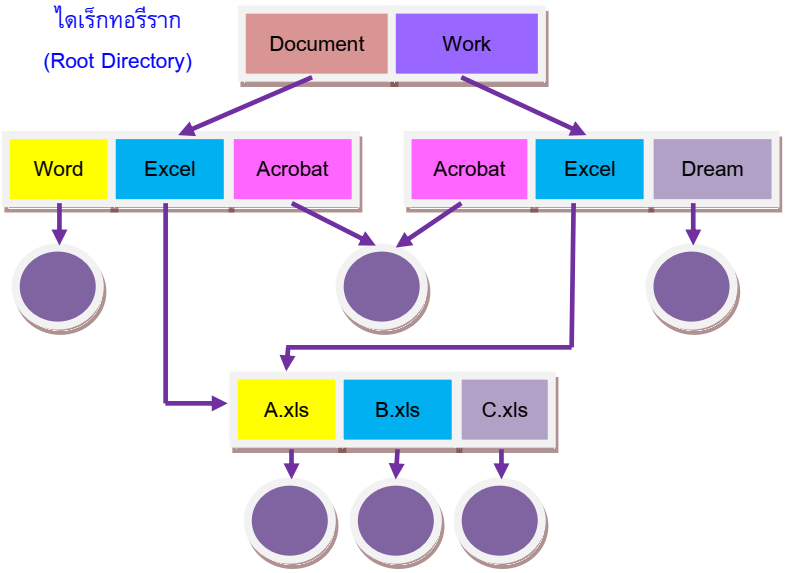
- ชื่อเส้นทางแบบสัมพันธ์ (Relative Path Name) คือ Excel --> B.xls
- ชื่อเส้นทางแบบสมบูรณ์ (Absolute Path Name) คือ Root --> Document --> Excel --> B.xls

ในส่วนของการลบไดเร็กทอรีของโครงสร้างแบบต้นไม้นั้นขึ้นอยู่กับระบบปฏิบัติการที่เลือกใช้ เช่น ถ้าเป็นระบบปฏิบัติการ MS-DOS จะใช้วิธีการลบไดเร็กทอรีย่อยให้หมดก่อนแล้วจึงไล่ขึ้นมาจนถึงไดเร็กทรอรีที่ต้องการลบ ส่วนในระบบปฏิบัติการ UNIX จะใช้ชุดคำสั่ง rmdir (remove directory) ใช้ลบไดเร็กทอรี ซึ่งทั้งไฟล์และไดเร็กทอรีย่อยจะถูกลบทั้งหมด



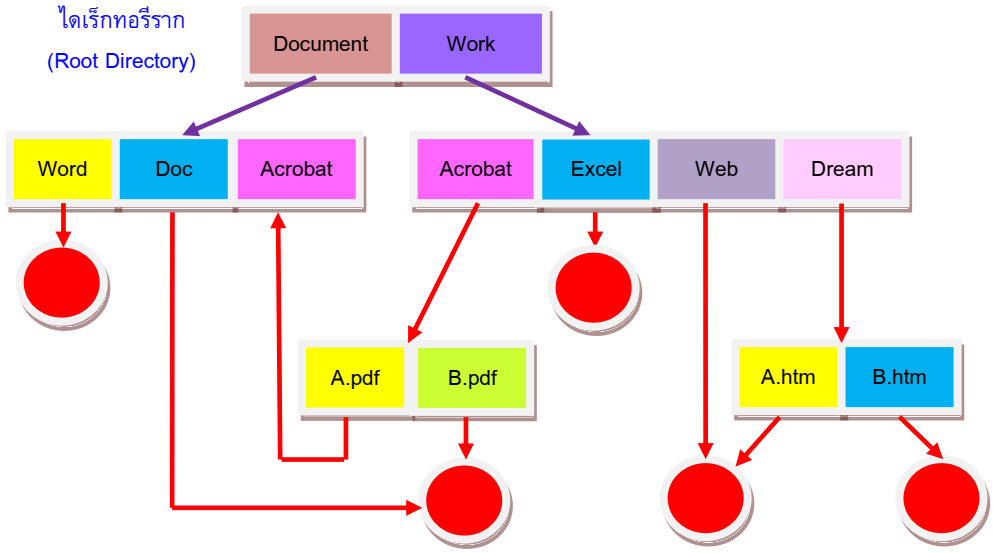
รูปที่ 4.9 แสดงโครงสร้างไดเร็กทอรี (Tree-Structure Directory)

4. ไดเร็กทอรีกราฟแบบไม่เป็นวงจร (Acyclic-Graph Directory) เป็นโครงสร้างต้นไม้ต้นไม้ที่มีการเชื่อมไดเร็กทอรีต่างๆ เข้าด้วยกันเพื่อจะได้สามารถทำงาน หรือใช้ไฟล์ร่วมกันได้ ทำให้เกิดไดเร็กทอรีที่มีลักษณะเป็นกราฟไม่เป็นวงจร (Acyclic Graph) โดยที่ไฟล์หรือไดเร็กทอรีย่อยเดียวกัน อาจอยู่ในไดเร็กทอรีที่ต่างกันได้ เช่น ระบบปฏิบัติการ UNIX จะใช้วิธีการใช้ไฟล์หรือไดเร็กทอรีย่อยร่วมกัน โดยการสร้างไดเร็กทอรีใหม่ที่เรียกว่า “ลิงก์ (Link)” โดยมีพอยเตอร์ (Pointer) ในการเชื่อมโยงไปยังไฟล์หรือไดเร็กทอรีย่อยอื่นๆ แสดงดังรูปที่ 4.10



รูปที่ 4.10 แสดงโครงสร้างไดเร็กทอรีแบบไม่เป็นวงจร (Acyclic-Graph Directory)

5. ไดเรกทอรีแบบกราฟทั่วไป(General Graph Directory) สำหรับไดเรกทอรีกราฟแบบไม่เป็นวงจรนั้น การเชื่อมไดเรกทอรีจะเป็นการเชื่อมไดเรกทอรีย่อยของไดเรกทอรีเข้าด้วยกัน แต่ไม่มีการเชื่อมไดเรกทอรีย่อยย้อนกลับไปหาไดเรกทอรีหลัก แต่ไดเรกทอรีแบบกราฟทั่วไปจะสามารถเชื่อมย้อนกลับไปมาระหว่างไดเรกทอรีย่อยกับไดเรกทอรีหลัก แสดงดังรูปที่ 4.11



รูปที่ 4.11 แสดงโครงสร้างไดเร็กทอรีแบบกราฟทั่วไป (General Graph Directory)

4.4 การใช้ไฟล์ร่วมกัน (File Sharing)

ระบบปฏิบัติการที่ดีจะต้องมีการจัดการไฟล์ที่ดี ในกรณีที่มีผู้ต้องการใช้งานไฟล์จำนวนมาก ระบบปฏิบัติการจำเป็นต้องคัดลอกไฟล์ให้กับผู้ใช้ทุกคนทำให้ต้องเสียพื้นที่ในหน่วยความจำ และถ้ามีการเปลี่ยนแปลงแก้ไขหรือคัดลอกไฟล์ก็จำเป็นต้องแจ้งให้กับผู้ใช้คนอื่นทราบด้วยทุกครั้ง ทำให้ยุ่งยากและเสียเวลาซึ่งอาจเกิดข้อผิดพลาดและเกิดความเสียหายกับไฟล์ได้ ดังนั้นการแก้ปัญหาดังกล่าวคือการใช้ไฟล์ร่วมกันและสำรองไฟล์ (Backup Files) ไปไว้ที่เดียวกัน เพื่อให้ผู้ใช้แต่ละตนสามารถที่จะเรียกใช้ไฟล์ที่ต้องการร่วมกันได้ ทำให้ประหยัดเวลาและการเข้าถึงไฟล์ทำได้รวดเร็วขึ้น โดยการใช้ไฟล์ร่วมกันแบ่งออกเป็น 3 รูปแบบ ดังนี้

1.การกำหนดผู้ใช้ได้หลายคน (Multiple Users) เป็นการกำหนดให้ไฟล์และไดเร็กทอรีให้กับผู้ใช้ โดยระบบปฏิบัติการจะเป็นตัวกำหนดกลุ่มของผู้ใช้เพื่อตรวจสอบสิทธิ์และการเข้าถึงไฟล์ที่ต้องการโดยแบ่งออกเป็น 2 กลุ่ม คือ

1.1 การระบุเอกลักษณ์เฉพาะผู้ใช้ (User Identifiers/User Ids/Security ID) เป็นการระบุตัวตนของผู้ใช้ โดยตัวเลขที่ใช้ระบุตัวตนของผู้ใช้ต้องไม่ซ้ำกัน และจัดเก็บเอกลักษณ์กลุ่มผู้ใช้ไว้ในรายการรายชื่อ (User Name List) เมื่อมีการเข้าถึงจึงมาเรียกใช้บริการผ่านรายการรายชื่อทุกครั้ง

1.2 การระบุเอกลักษณ์เฉพาะกลุ่ม (Group Identifiers) เป็นการระบุตัวตนของผู้ใช้ภายในกลุ่ม โดยที่ผู้ใช้อาจจะมีตัวตนในกลุ่มเดี่ยวหรือหลายกลุ่มก็ได้ขึ้นอยู่กับระบบปฏิบัติการที่เลือกใช้

2.ระบบการใช้ไฟล์ทางไกล (Remote File System) เป็นการใช้ไฟล์ร่วมกันระหว่างคอมพิวเตอร์ผ่านระบบเครือข่าย (Network) โดยระบบเครือข่ายจะอนุญาตให้ผู้ใช้สามารถติดต่อสื่อสารและใช้ไฟล์ร่วมกันกันได้ โดยวิวัฒนาการของระบบเครือข่ายและเทคโนโลยีเกี่ยวกับไฟล์มีวิธีการใช้ไฟล์ร่วมกันดังนี้

2.1 การใช้ไฟล์ร่วมกันระหว่างเครื่องคอมพิวเตอร์ผ่านทางโปรแกรม โดยดำเนินการผ่านทางโปรโตคอล (Protocol) ในการติดต่อสื่อสารเพื่อรับส่งไฟล์หรือข้อมูล เช่น FTP (File Transfer Protocol)

2.2 การใช้ไฟล์ร่วมกันแบบกระจาย ซึ่งมีการร้องขอและส่งไฟล์หรือข้อมูลโดยการเชื่อมต่อระหว่างเครือข่ายแม่/เครือข่ายลูก (Client/Server) อาศัยโปรแกรมประยุกต์พัฒนาขึ้นมาและทำงานในลักษณะที่เรียกว่า DFS (Distributed File System)

2.3 การใช้ไฟล์ร่วมกันแบบ World Wide Web โดยอาศัยบราวเซอร์ (Browser) เพื่อช่วยในการเข้าถึงไฟล์บนเครื่องแม่ (Server) แล้วจึงถ่ายโอนไฟล์ (Transfer Files) ที่ต้องการไปยังเครื่องลูก (Client) ของผู้ใช้

3.การระบุความหมายให้สอดคล้องกัน (Consistency Semantic) เป็นการระบุเพื่อให้ผู้ใช้แต่ละคนสามารถอ้างถึงไฟล์ที่ต้องการได้ ในกรณีที่มีการใช้ไฟล์ร่วมกันหากมีการแก้ไขข้อมูลในไฟล์โดยผู้ใช้คนใดคนหนึ่งจะต้องแจ้งให้ผู้ใช้คนอื่นแก้ไขข้อมูลตามไปด้วย การทำงานจะสัมพันธ์โดยตรงกับกระบวนการอย่างต่อเนื่องที่เรียกว่า Process Synchronization Algorithm โดยระบบที่ถูกพัฒนาขึ้นมาเพื่อการระบุความหมายให้สอดคล้องกันเมื่อมีการใช้ไฟล์ร่วมกัน คือ

3.1 The UNIX File System เป็นระบบที่อนุญาตให้ผู้ใช้สามารถเปิดไฟล์ขึ้นมาเพื่ออ่านหรือเขียนและใช้พอยเตอร์ร่วมกันในการอ้างถึงตำแหน่งที่เป็นปัจจุบันไปยังไฟล์ที่ต้องการใช้งานได้

3.2 The Andrew File System เป็นระบบที่มีการเข้ารหัสข้อมูลเพื่อช่วยรักษาความปลอดภัยให้กับข้อมูลในไฟล์และรองรับการกระจายของข้อมูลเมื่อมีการใช้ไฟล์ร่วมกัน

4.5 การป้องกัน (Protection)

ในระบบคอมพิวเตอร์ทุกระบบ การป้องกันและควบคุมการเข้าถึงไฟล์โดยผู้ไม่ได้รับสิทธิ์ (Unauthentication) หรือผู้ไม่หวังดีรวมถึงผู้บุกรุก (Hacker) ถือว่าเป็นสิ่งสำคัญอย่างมากและสามารถทำได้หลายวิธีซึ่งจะต้องคำนึงถึงปัจจัยหลายประเภท เช่น ประเภทของการเข้าถึง การควบคุมการเข้าถึง การป้องกันโดยกำหนดรหัสผ่าน ซึ่งแต่ละวิธีมีรายละเอียดการป้องกันไฟล์ ดังนี้

1.ประเภทของการเข้าถึงไฟล์ (Type of Access) กลไกการป้องกันหรือการจัดเตรียมการควบคุมการเข้าถึงไฟล์ โดยจำกัดชนิดของการเข้าถึงไฟล์สามารถทำได้และขึ้นอยู่กับปัจจัยหลายประการรวมถึงชนิดของการร้องขอเพื่อเข้าถึงไฟล์มีดังนี้

1.1 การอ่าน (Read) เป็นการอ่านข้อมูลจากไฟล์

1.2 การเขียน (Write)เป็นการ เขียนหรือแก้ไขข้อมูลในไฟล์

1.3 การประมวลผล (Execute) เป็นการสั่งให้ประมวลผลข้อมูลในไฟล์

1.4 การเพิ่มข้อมูล (Append) เป็นการเพิ่มข้อมูลลงในไฟล์โดยการเพิ่มข้อมูลต่อท้ายไฟล์เดิม

1.5 การลบ (Delete) เป็นการลบไฟล์และคืนพื้นที่กับสู่ระบบเพื่อนำไปใช้ใหม่

1.6 การแสดงรายชื่อ (List) เป็นการแสดงชื่อและคุณลักษณะของไฟล์

2. การควบคุมการเข้าถึง (Access Control) เป็นการกำหนดสิทธิ์ในการเข้าถึงไฟล์ ซึ่งโดยทั่วไปการป้องกันการเข้าถึงไฟล์จะใช้วิธีการระบุตัวตน (Identify) ของผู้ใช้และไฟล์หรือไดเร็กทอรีที่ต้องการเข้าถึงมักจะเก็บไว้ในรายการการเข้าถึง (Access-Control List: ACL) หากเป็นระบบที่ใช้งานคนเดียว เช่น ในเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) มักไม่ค่อยมีการกำหนดสิทธิ์ในการใช้ข้อมูลในไฟล์มากนักเพราะถือว่ามีการใช้งานเพียงคนเดียว แต่ในกรณีที่เป็นระบบเครือข่ายคอมพิวเตอร์ เช่น Server ของเวปเพจ จะมีการเข้ามาใช้และแก้ไขข้อมูลในไฟล์อยู่ตลอดเวลาและมีผู้ใช้บริการหลายคน ดังนั้นจึงจำเป็นจะต้องมีการกำหนดสิทธิ์ในการเข้าถึงไฟล์ ในระบบปฏิบัติการ UNIX จะมีการกำหนดกำหนดสิทธิ์ในการเข้าถึงข้อมูลไว้ 3 ชนิด คือ R(Read), W(Write) และ X(Execute) และในการเข้ามาใช้งานระบบแบบกลุ่มหรือหลายคน ระบบต้องมีการจัดระดับของผู้เข้ามาใช้บริการ เพื่อจำกัดสิทธิ์ในการเข้าถึงข้อมูล โดยในระบบ UNIX จะแบ่งผู้ใช้บริการเป็น 3 กลุ่ม คือ

2.1 เจ้าของ (Owner) เป็นกลุ่มเจ้าของไดเรกทอรีนั้นๆ โดยมากจะกำหนดสิทธิ์ในการเข้าถึงข้อมูลในไฟล์ได้ทุกไฟล์

2.2 กลุ่ม (Group) เป็นกลุ่มผู้ใช้บริการที่มีไดเรกทอรีของตัวเองใน Server เช่นเดียวกัน แต่ไม่ใช่เจ้าของไดเรกทอรี บางครั้งระบบจะให้สิทธิ์ในการเข้าถึงไฟล์ในไดเรกทอรีต่างๆด้วย เช่น สามารถเข้าไปดูข้อมูลในไดเรกทอรีของคนอื่นได้ (ยกเว้นเจ้าของจะไม่อนุญาต) แต่จะไม่มีสิทธิ์ในการแก้ไขข้อมูลในของคนอื่น

2.3 ผู้ใช้ทั่วไป (Universe หรือ Other) เป็นผู้ใช้บริการที่เป็นคนนอกที่ไม่มีไดเรกทอรีเป็นของตนเอง ผู้ใช้บริการเหล่านี้มักจะถูกกำหนดสิทธิ์ไว้ต่ำสุด คืออาจจะอ่านข้อมูลได้อย่างเดียว หรืออาจอ่านและประมวลผลได้ แต่มักไม่อนุญาตให้แก้ไขข้อมูลได้

ตัวอย่าง 1 ชูใจสร้างเว็บไซต์รับสมัครงานขึ้นมาและรับสมัครพนักงานใหม่เข้ามา 3 คน เพื่อช่วยพัฒนาเว็บไซต์ได้แก่ มานะ ปิติ และมานี ชูใจสร้างไฟล์ชื่อ "Web" สำหรับจัดเก็บรายละเอียด และป้องกันการเข้าถึงไฟล์ โดยกำหนดสิทธิ์การใช้งานให้กับผู้ใช้แต่ละคนดังนี้

1. ชูใจ เจ้าของ (Owner) สามารถจะดำเนินการกับไฟล์ได้ทุกอย่าง
2. มานะ ปิติ และมานี กลุ่ม (Group) สามารถอ่านและเขียนไฟล์ได้ แต่ไม่สามารถลบไฟล์นี้ได้
3. ผู้ใช้ทั่วไป (Universe หรือ Other) สามารถเข้ามาอ่านไฟล์นี้ได้เท่านั้น (ชูใจต้องการให้ผู้สนใจเข้ามาเยี่ยมชมเว็บไซต์ของตัวเอง)

นอกจากนี้ชูใจยังสามารถป้องกันไฟล์ได้โดยสร้างกลุ่มใหม่ (New Group) ขึ้นมา โดยใช้ชื่อว่า "Site" ซึ่งกำหนดสิทธิ์ให้มานะ ปิติ และมานี สามารถเข้าถึงไฟล์ได้และไฟล์ "Site" จะเกี่ยวข้องกับไฟล์ "Web" และกำหนดสิทธิ์ในการเข้าถึงไฟล์ตามเงื่อนไขที่กำหนดไว้เดิม

ตัวอย่าง 2 ระบบปฏิบัติการ UNIX กำหนดบิตควบคุมในการเข้าถึงไฟล์จำนวน 3 บิต คือ r, w และ x โดย r ควบคุมการอ่าน (Read) w ควบคุมการเขียน (Write) x ควบคุมการประมวลผล (Execution) จากตัวอย่างที่ 1 สามารถป้องกันไฟล์โดยกำหนดสิทธิ์ในการเข้าถึงไฟล์ได้ดังนั้น

1. ชูใจ เจ้าของ (Owner) สามารถควบคุมการเข้าถึงไฟล์ได้ทั้ง 3 บิต
2. มานะ ปิติ และมานี กลุ่ม (Group) สามารถควบคุมการเข้าถึงโดยใช้บิต r และ w
3. ผู้ใช้ทั่วไป (Universe หรือ Other) สามารถควบคุมการเข้าถึงโดยใช้บิต r เท่านั้น

3. การป้องกันโดยกำหนดรหัสผ่าน (Password Protection) โดยกำหนดกลุ่มของรหัสผ่านให้แต่ละไฟล์ การเข้าถึงไฟล์ในระบบคอมพิวเตอร์มักจะถูกควบคุมโดยรหัสผ่าน วิธีอาจมีข้อเสียคือ

1. ถ้าจำนวนไฟล์มีมากผู้ใช้อาจจำรหัสผ่านไม่ได้ทั้งหมด
2. ถ้ากำหนดให้ใช้เพียงรหัสผ่านเดียวในการเข้าถึงทุกๆ ไฟล์ได้ หากถูกค้นพบอาจจะทำให้ผู้ใช้สามารถนำรหัสผ่านไปใช้กับไฟล์ได้ทั้งหมด
3. ปกติการใช้รหัสผ่านเพียงรหัสเดียวในการกำหนดกลุ่มของไฟล์ทั้งหมด ทำให้การป้องกันการเข้าถึงไฟล์ขั้นพื้นฐานทำได้ยาก จึงจำเป็นจะต้องกำหนดระดับรายละเอียดโดยใช้รหัสผ่านหลายตัว

สรุป

ไฟล์ คือ กลุ่มข้อมูลหรือสารสนเทศที่มีความสัมพันธ์กันและถูกจัดเก็บไว้บนหน่วยความจำรอง แบ่งออกเป็น 2 ประเภท คือ ไฟล์โปรแกรมและไฟล์ข้อมูล การจัดเก็บไฟล์ข้อมูล เมื่อมีการเรียกใช้จะต้องมีการเข้าถึงและอ่านไฟล์นั้นเข้ามาเก็บยังหน่วยความจำหลัก ซึ่งข้อมูลที่อยู่ในไฟล์สามารถเข้าถึงได้หลายวิธี ดังนั้นระบบปฏิบัติการจะทำการเลือกวิธีที่เหมาะสมเพียงหนึ่งวิธีเท่านั้นในการเข้าถึงไฟล์และเลือกโปรเซสที่เหมาะสมเพื่อใช้ในการแก้ปัญหาการเข้าถึงไฟล์ แบ่งออกเป็น การเข้าถึงแบบเรียงลำดับและการเข้าถึงแบบทางตรง

โครงสร้างไดเร็กทอรี เป็นกลไกที่ใช้สร้างและจัดเก็บแฟ้ม (Folder) ที่บรรจุไฟล์ต่างๆ ไว้ สามารถจำแนกออกเป็นลักษณะต่างๆ เช่น โครงสร้างแบบไดเร็กทอรีเดียว โตรงสร้างไดเร็กทอรีแบบ 2 ระดับ โครงสร้างไดเร็กทอรีแบบต้นไม้ โครงสร้างไดเร็กทอรีแบบกราฟไม่เป็นวงจรและโครงสร้างไดเรกทอรีแบบกราฟทั่วไป

การใช้ไฟล์รวมกัน เป็นวิธีการสำรองไฟล์ (Back up) ไปวางไว้เพียงที่เดียว และให้ผู้ใช้แต่ละคนสามารถที่จะระบุชื่อไฟล์ที่ต้องการเรียกใช้ได้ โดยการใช้ไฟล์รวมกันแบ่งออกเป็น 3 รูปแบบ คือ การกำหนดผู้ใช้ได้หลายคน (Multiple User) ระบบการใช้ไฟล์ทางไกล (Remote File System) และการระบุความหมายให้สอดคล้องกัน (Consistency Semantic)

ส่วนการป้องกันและควบคุมการเข้าถึงไฟล์โดยผู้ไม่ได้รับสิทธิ์ (Unauthentication) หรือผู้ไม่หวังดีรวมถึงผู้บุกรุก (Hacker) ถือว่าเป็นสิ่งสำคัญอย่างยิ่งของระบบปฏิบัติการสามารถทำได้หลายวิธีซึ่งจะต้องคำนึงถึงปัจจัยหลายประเภท เช่น ประเภทของการเข้าถึง การควบคุมการเข้าถึง การป้องกันโดยกำหนดรหัสผ่าน ซึ่งแต่ละวิธีมีรายละเอียดการป้องกันไฟล์ เป็นต้น