

# File-System Interface

File คือ uniform logical view of stored information mapped to physical devices

## File Attributes

Name

Identifier (id)

Type

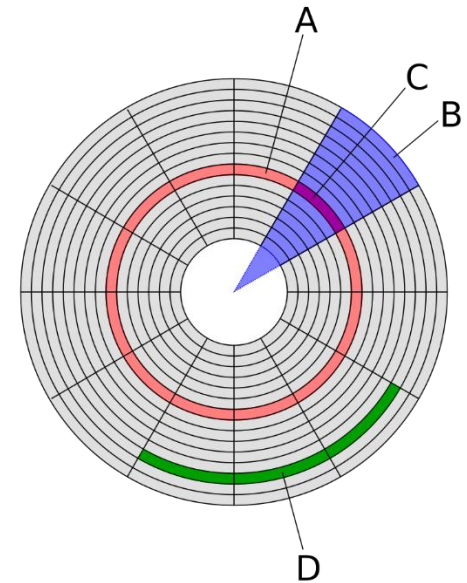
Location

Size

Protection

Timestamps (created, last modified, last use)

User identification (owner)



physical device อาจจะไม่ uniform

# File Operations

Creating a file

Writing a file

Reading a file

Repositioning within a file (seek – not involve any actual I/O)

Delete a file

Truncating a file (erase the contents of a file, but keep its attributes)

Appending, renaming, and copying

**Open-file table** contains all information about open files.

└─ 1 table ต่อ 1 process

**Per-process table** tracks all files that a process has open.

└─ 1 table ต่อ 1 OS

**System-wide table** contains process independent information.

**open()** returns a pointer to the file in open-file (per-process) table.

**close()** removes the entry in per-process table and  
decrease file-open count in system-wide table.

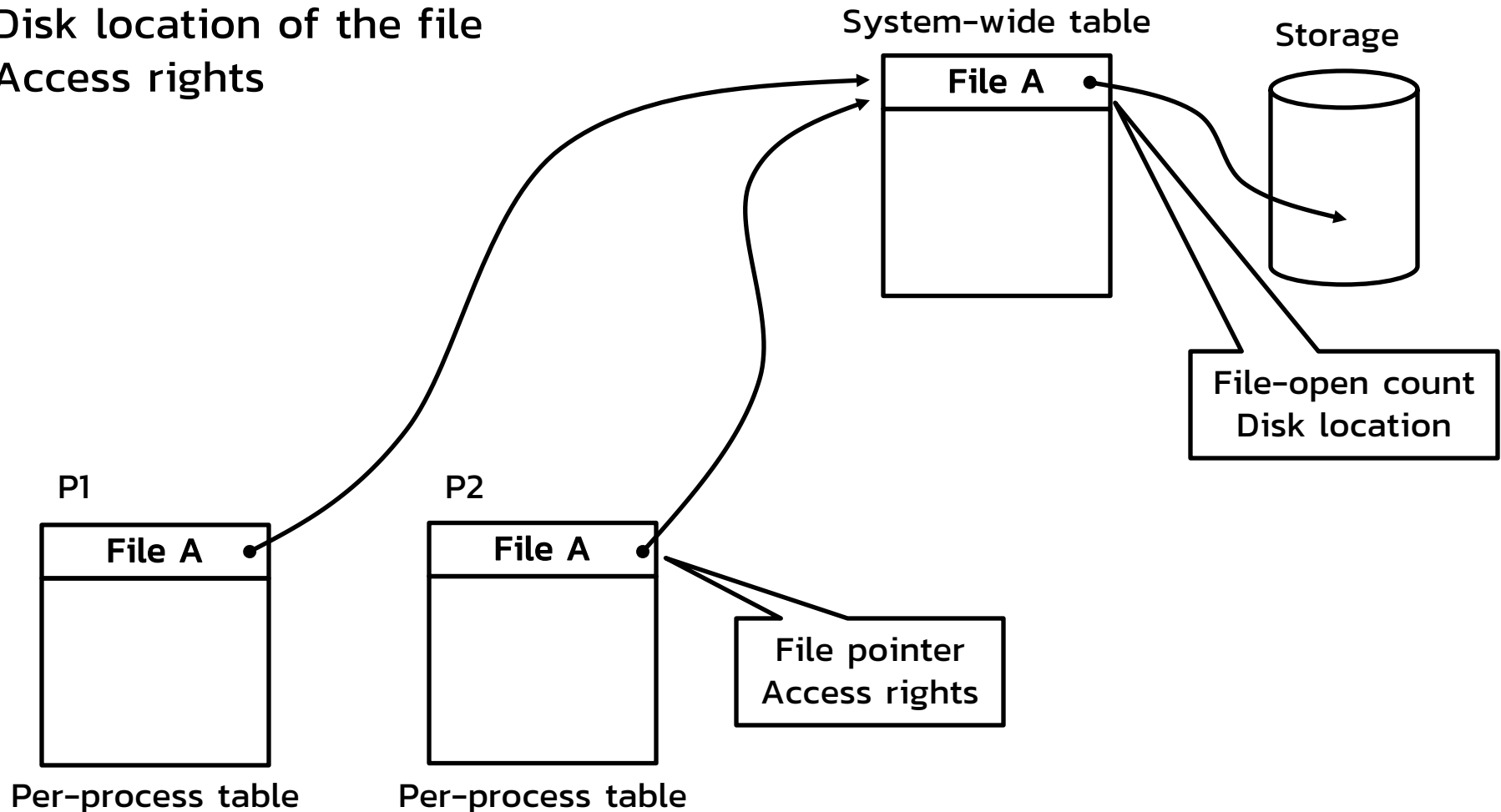
# Information associated with an open file

File pointer

File-open count

Disk location of the file

Access rights



## FILE LOCKING IN JAVA

ของจริงจะเพี้ยน ๆ จากนี้นิดหน่อย  
เข้าใจว่าผู้แต่ง **simplify** มา

In the Java API, acquiring a lock requires first obtaining the `FileChannel` for the file to be locked. The `lock()` method of the `FileChannel` is used to acquire the lock. The API of the `lock()` method is

`FileLock lock(long begin, long end, boolean shared)`

มีทั้งแบบ **synchronous (blocking)** และ **asynchronous (non-blocking)**

where `begin` and `end` are the beginning and ending positions of the region being locked. Setting `shared` to `true` is for shared locks; setting `shared` to `false` acquires the lock exclusively. The lock is released by invoking the `release()` of the `FileLock` returned by the `lock()` operation.

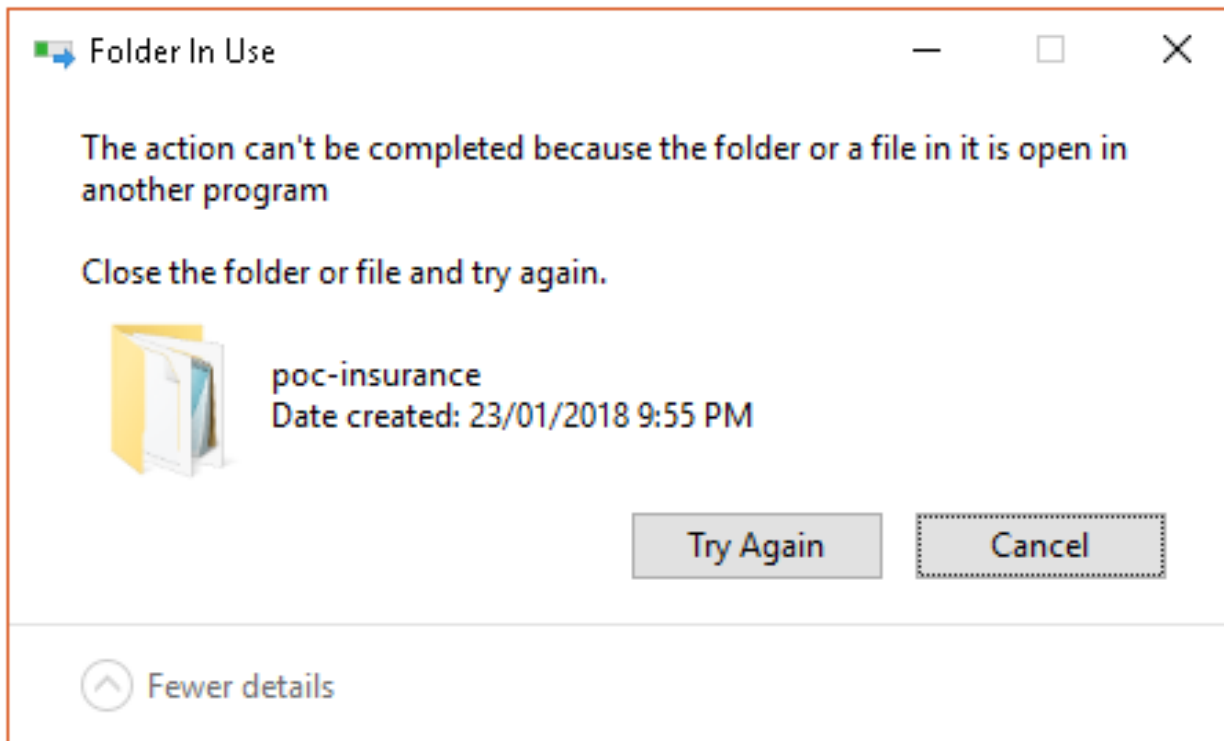
The program in Figure 13.2 illustrates file locking in Java. This program acquires two locks on the file `file.txt`. The lock for the first half of the file is an exclusive lock; the lock for the second half is a shared lock.

- Shared (r)** several processes can acquire the lock concurrently.  
(ถ้ายัง shared lock อยู่ ห้าม exclusive lock)  
ใช้สำหรับ reader
- Exclusive (w)** only one process at a time can acquire such a lock.  
(ถ้ายัง exclusive lock อยู่ ห้าม shared/exclusive lock)  
ใช้สำหรับ writer

# Mandatory vs. Advisory Locking

Mandatory	OS ensures locking integrity.
Advisory	it's up to software developers (acquired & released).

Windows	mandatory locking
Linux	advisory locking



# File Types

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine- language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, perl, asm	source code in various languages
batch	bat, sh	commands to the command interpreter
markup	xml, html, tex	textual data, documents
word processor	xml, rtf, docx	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	gif, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	rar, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

**Figure 13.3** Common file types.

# Magic number

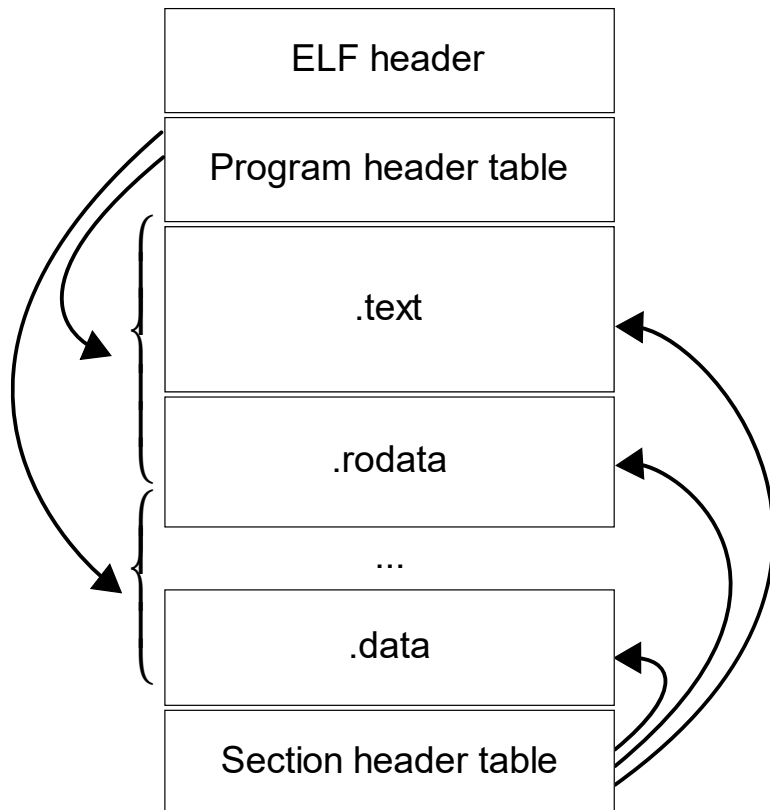
ไฟล์บน Linux ไม่จำเป็นต้องมีนามสกุล (extension)  
ดูจาก magic number

```
od -h --endian=big file.ext
```

EXE	7F 45 4C 46
PNG	89 50 4E 47 0D 0A 1A 0A

# File Structures

OS ส่วนใหญ่จะมี file structures ให้น้อยที่สุด ลองดูตัวอย่าง file structure ของ executable files บน Linux เมื่อสั่ง execute ไฟล์นี้ OS จะต้องตรวจสอบ file structure ก่อนว่าถูกต้องหรือไม่ ถ้าเป็นไฟล์อื่น ๆ ก็ไม่ต้องตรวจสอบ (ง่าย)



แต่ละส่วนของไฟล์ exe จะถูกโหลดลง memory ที่ตำแหน่งต่างกัน



# Internal File Structure

- Linux ไฟล์ คือ a stream of bytes
- Logical record size = 1 byte
- ปัจจุบันอุปกรณ์ที่ใช้เก็บข้อมูลนิยมแบ่งพื้นที่เป็น block (sector)
- Physical block size = 512, 1K, 2K, 4K ขึ้นกับฮาร์ดแวร์ที่ใช้
- Packing technique เป็นวิธีการเก็บ logical records ลงบน physical blocks ใช้ซอฟต์แวร์ทำได้ง่าย ๆ



# File Access Methods (ไม่เกี่ยวกับ physical devices)

Sequential Access (ถึงแม้ว่า device อาจจะเป็น random access)  
to access record  $n$ , must access record  $n - 1$

Direct Access  
fixed-length logical records

Other Access Methods

indexing

ISAM (Index sequential-access method, IBM)

```
AAA\nBBBBB\nCCCC\nDDDDDD
```

Variable length

```
AAA□□□□□□  
BBBBB□□□□□  
CCCC□□□□□□  
DDDDDD□□□□
```

Fixed length



Sequential access

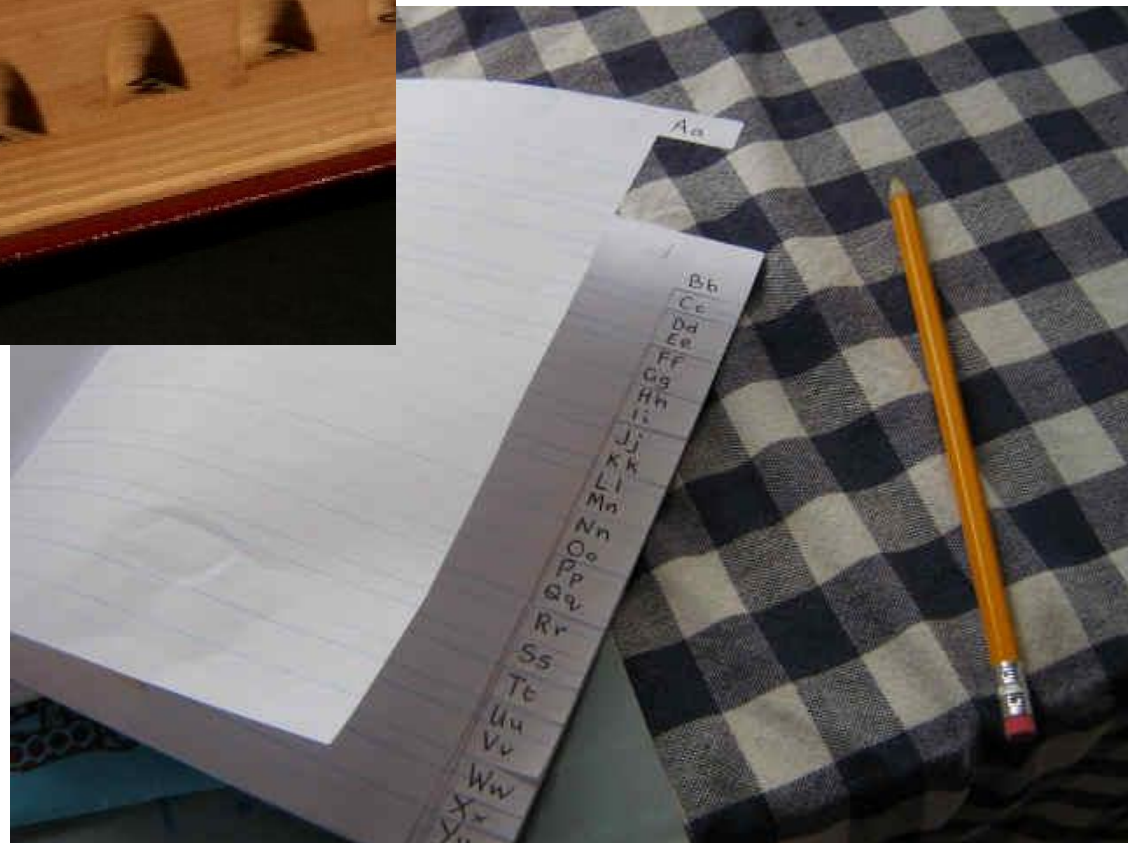


หน่วยความจำ เป็น random access



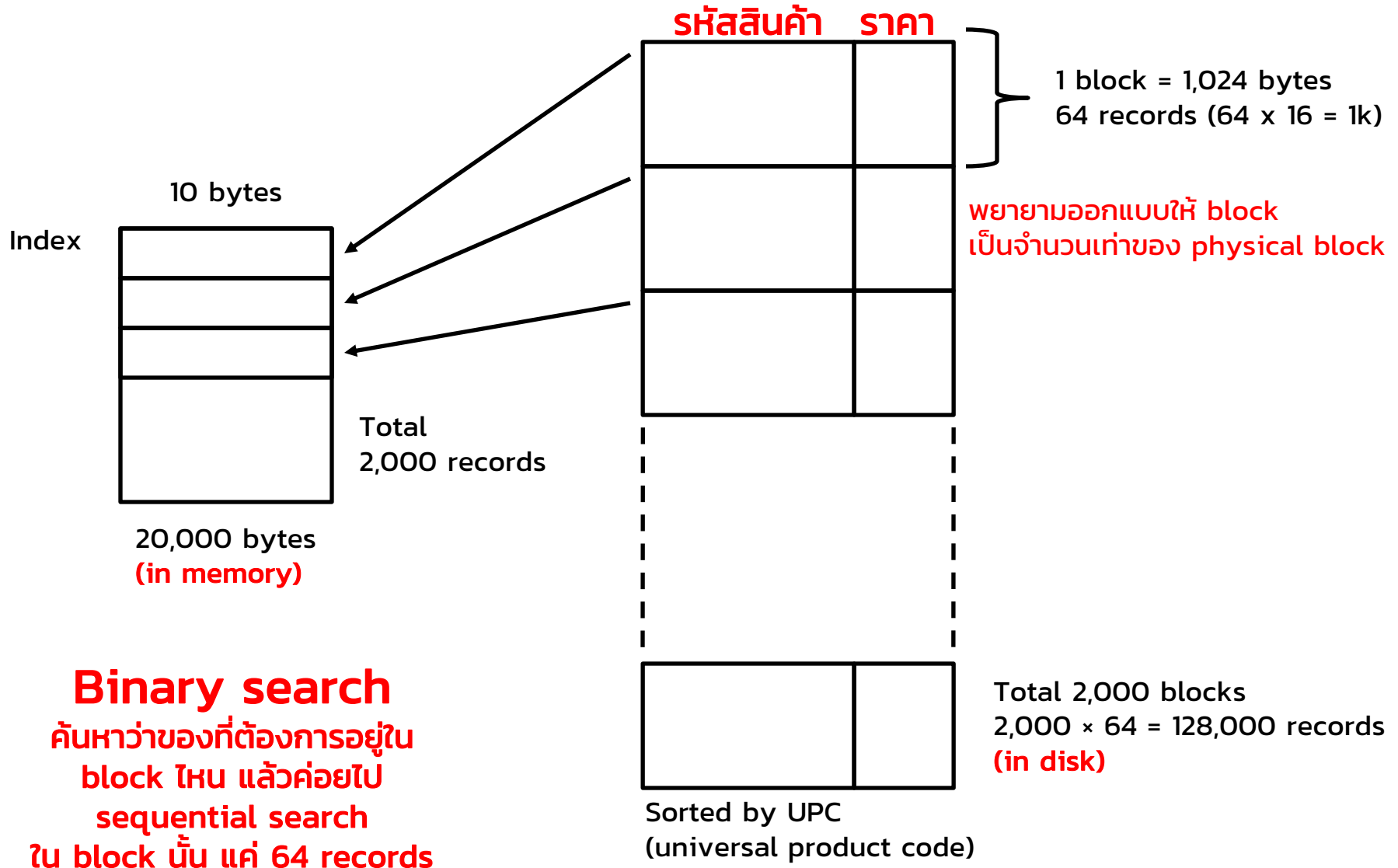
HD อนุโลมว่าเป็น random access

# Indexing

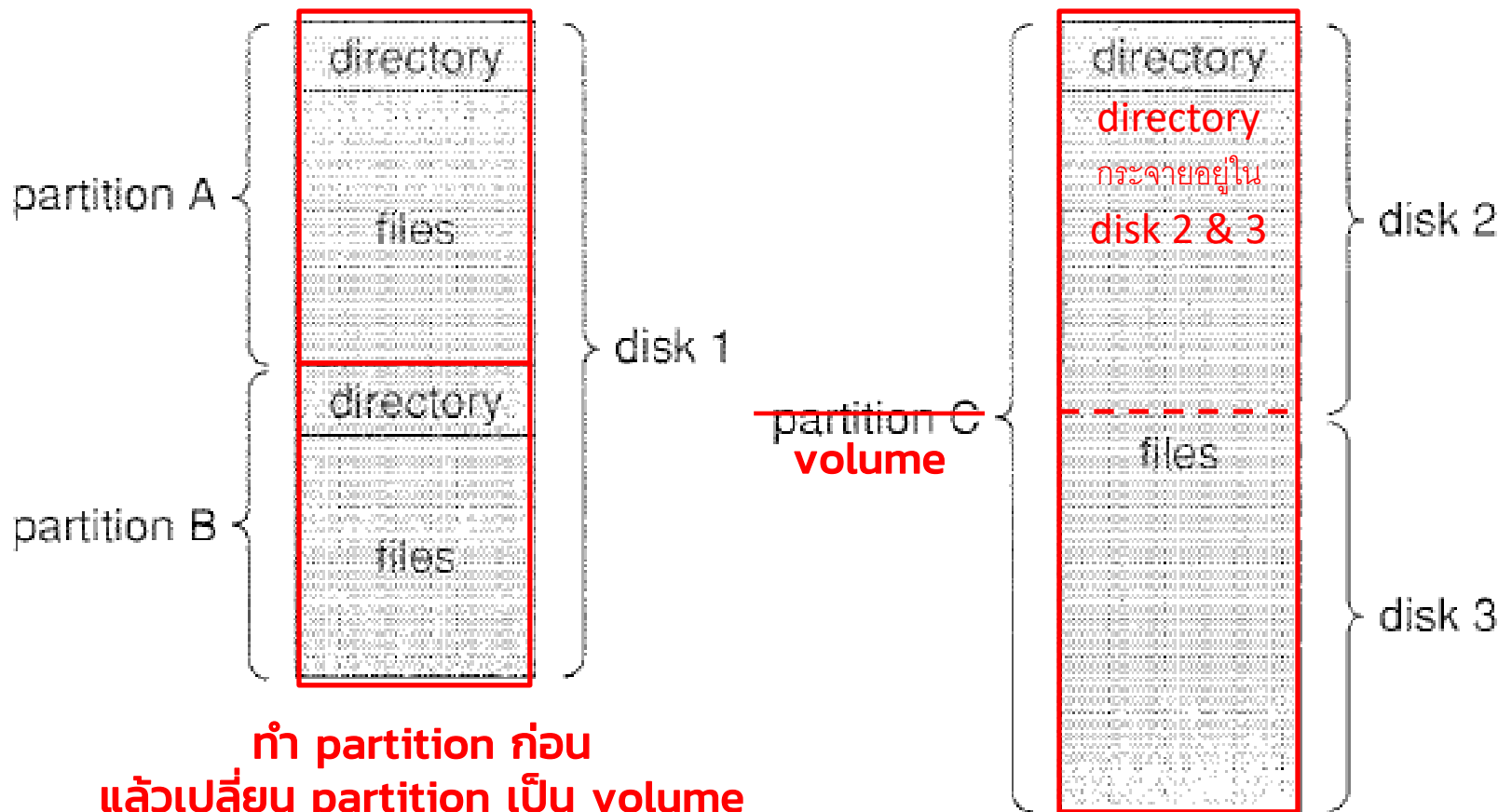


# Indexing

1 record = 10 bytes (UPC) + 6 bytes (price)



# Directory and Disk Structure



ทำ partition ก่อน  
แล้วเปลี่ยน partition เป็น volume  
เช่น drive C และ D ของ Windows  
(directory เกิดขึ้นเมื่อสร้าง file system)

เอาดิสก์ (partition) หลาย ๆ ตัว  
มารวมกันเป็น volume ได้  
เช่น LVM ของ Linux





## Partition

a logical division of a disk

Partition เป็นการแบ่งพื้นที่ดิสก์ไว้เฉย ๆ

ในกรณีทั่วไป  $\#disk : \#partition = 1 : 1$

## Volume

a logical assembly of one or more partitions

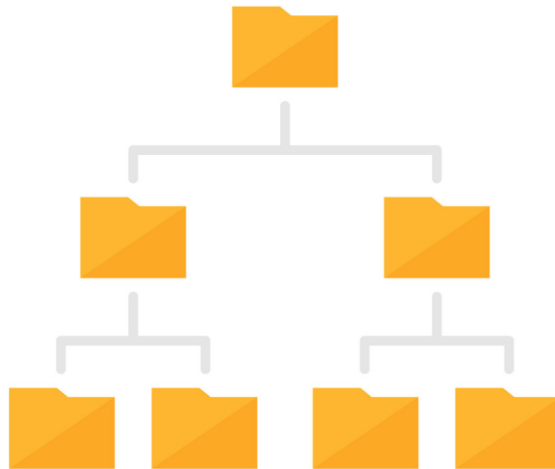
OS จะใช้ file system ได้ ต้อง mount ก่อน เพื่อให้ได้ volume (1 volume ติดตั้งได้แค่ 1 file system เท่านั้น)

ในกรณีทั่วไป  $\#volume : \#partition = 1 : 1$

## Directory

a file system cataloging structure which contains references to other computer files, and possibly other directories.

มี directory ช้อน directory ได้ (directory  $\approx$  folder)



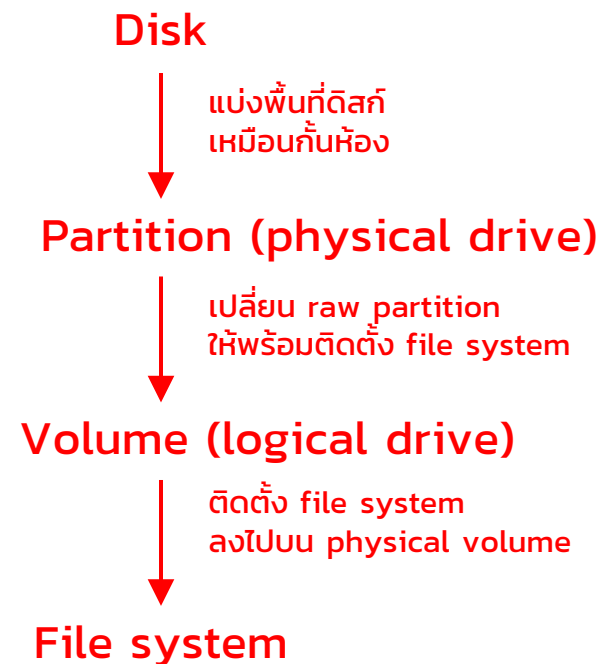
## Example [\[edit\]](#)

This example concerns a [Windows XP](#) system with two physical hard disks. The first hard disk has two partitions, the second has only one. The first partition of the first hard disk contains the operating system. [Mount points](#) have been left at defaults.

Physical disk	Partition	Filesystem	Drive letter
Hard Disk 1	Partition 1	NTFS	C:
	Partition 2	FAT32	D:
Hard Disk 2	Partition 1	FAT32	E:

In this example,

- "C:", "D:", and "E:" are volumes.
- Hard Disk 1 and Hard Disk 2 are physical disks.
- Any of these can be called a "drive".

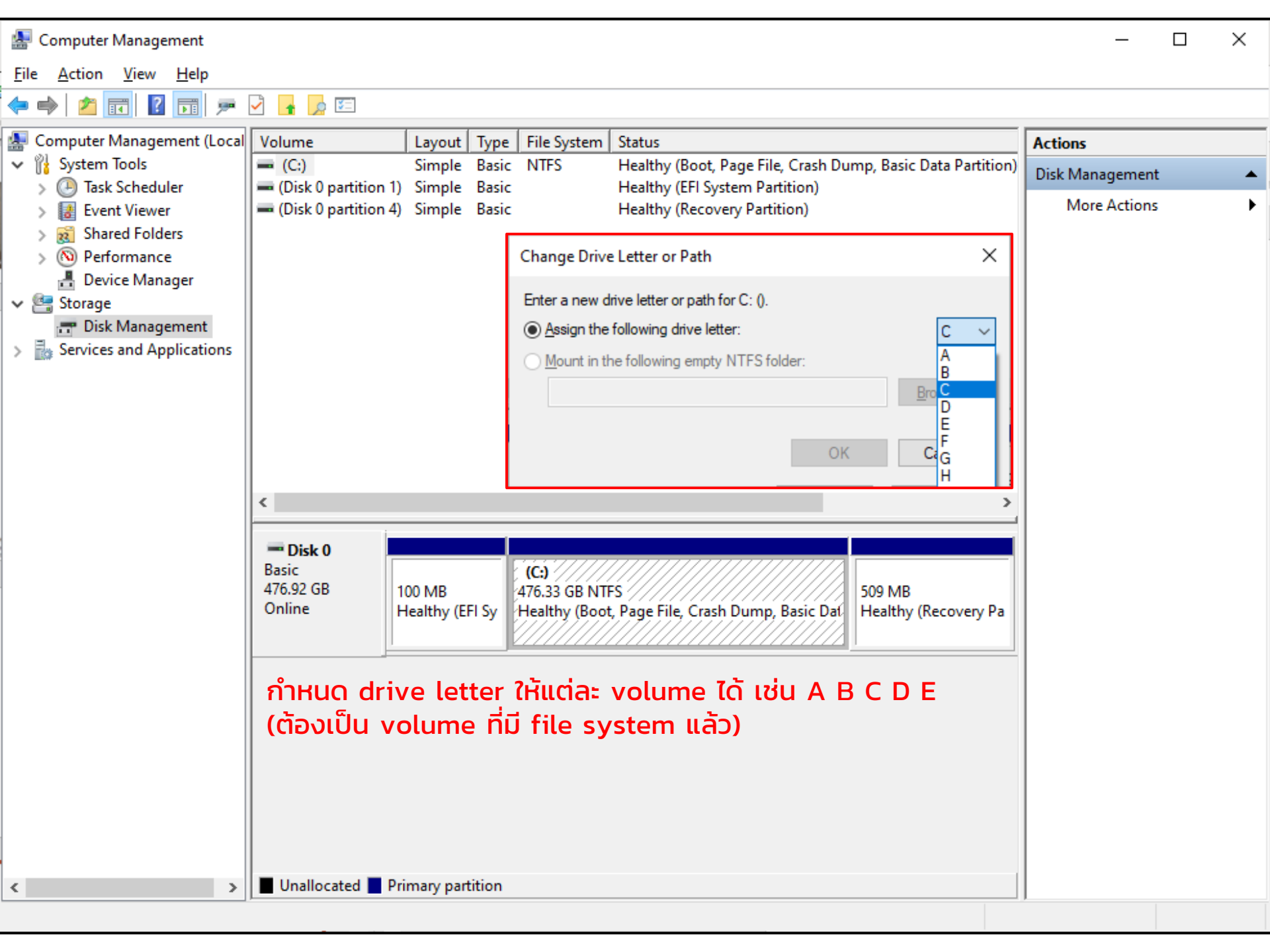


ปกติจะทำแค่ 2 steps คือ

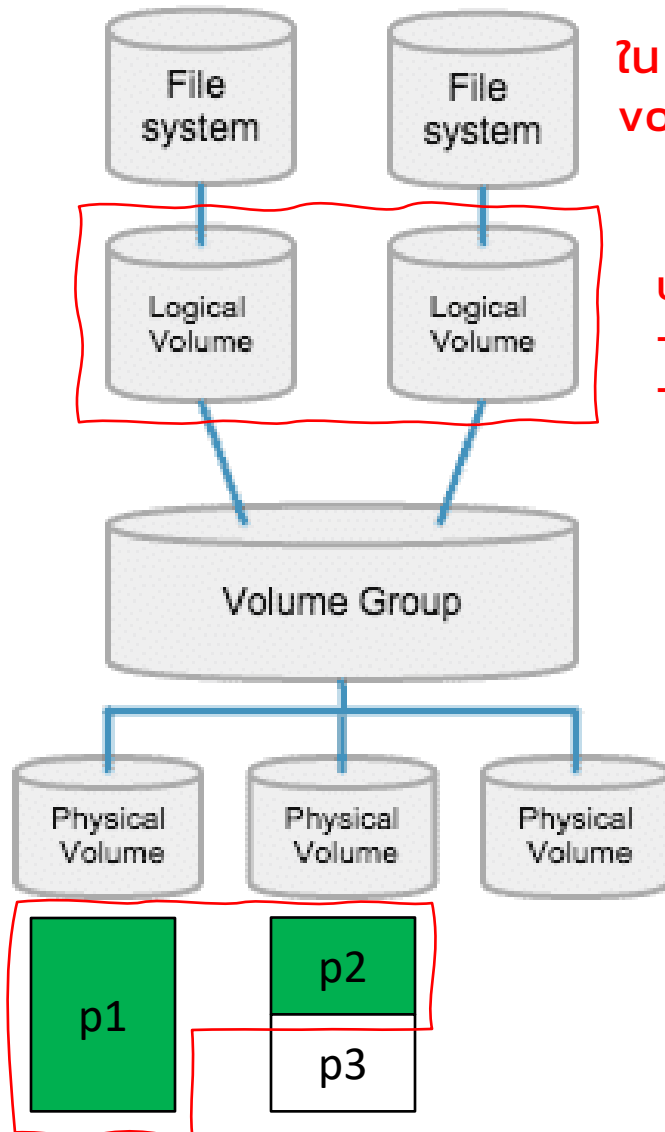
1. create a partition (fdisk)
2. make a file system (mkfs)

การสร้าง volume จะรวมไปกับขั้น 1 หรือ 2 (?)





# Logical Volume Management (Linux)



ใน Linux เวอร์ชันก่อน file system จะอยู่บน physical volume ไม่ต้องมี logical volume ปัจจุบันทำได้ 2 แบบ

ประโยชน์

- สร้าง logical volume ที่ใหญ่กว่า disk ได้ เช่น 100 TB
- logical volume ทำ resize ได้ง่ายกว่า

Linux เมื่อสร้าง partition แล้วต้องสร้าง physical volume ด้วยคำสั่ง pvcreate เพราะเราต้องการแค่ physical volume เปล่า ๆ ที่ยังไม่มี file system

# Directory Overview

Symbol table (file names → directory entry)

Basic operations on a directory

- Search for a file
- Create a file
- List a directory
- Rename a file
- Traverse the file system ၁၃၀ `dir /s` (Windows) ၁၃၁ `ls -R` (Linux)

# Single-Level Directory

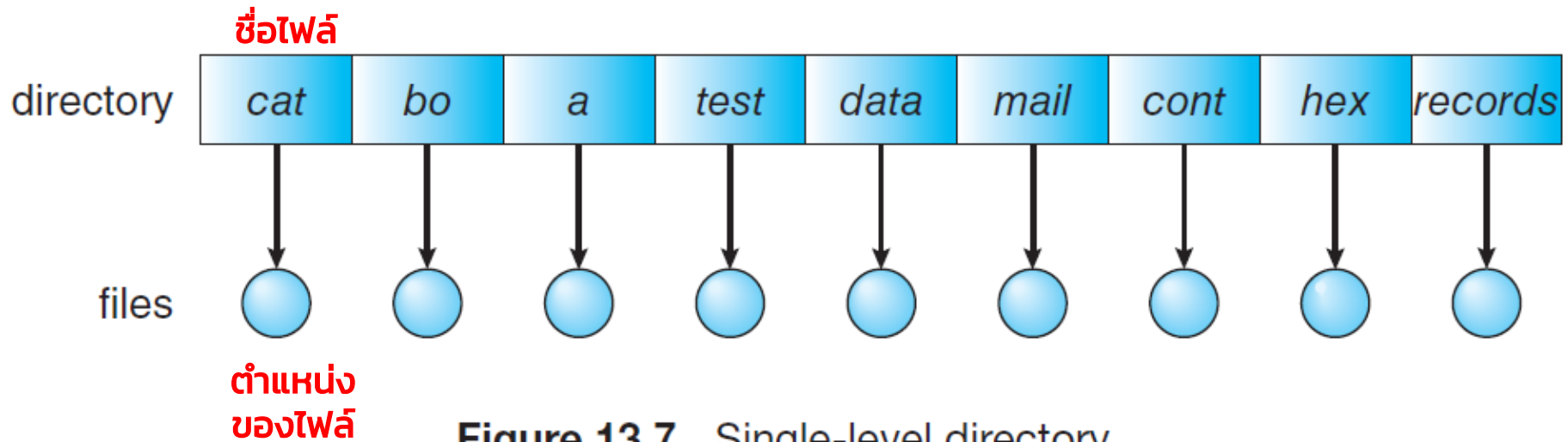
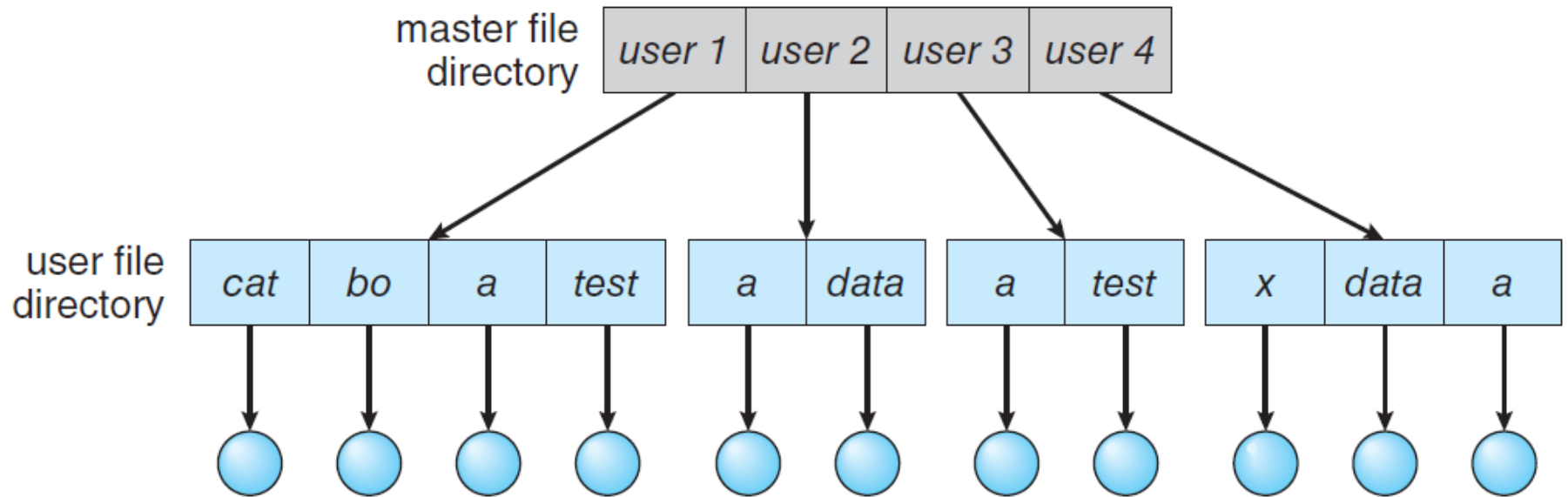


Figure 13.7 Single-level directory.

## Limitations

- when the number of files increases
- when the number of user increases
- naming collision (unique-name rule is violated)

# Two-Level Directory



**Figure 13.8** Two-level directory structure.

The first level (master file directory) separates users.

# Tree-Structured Directory

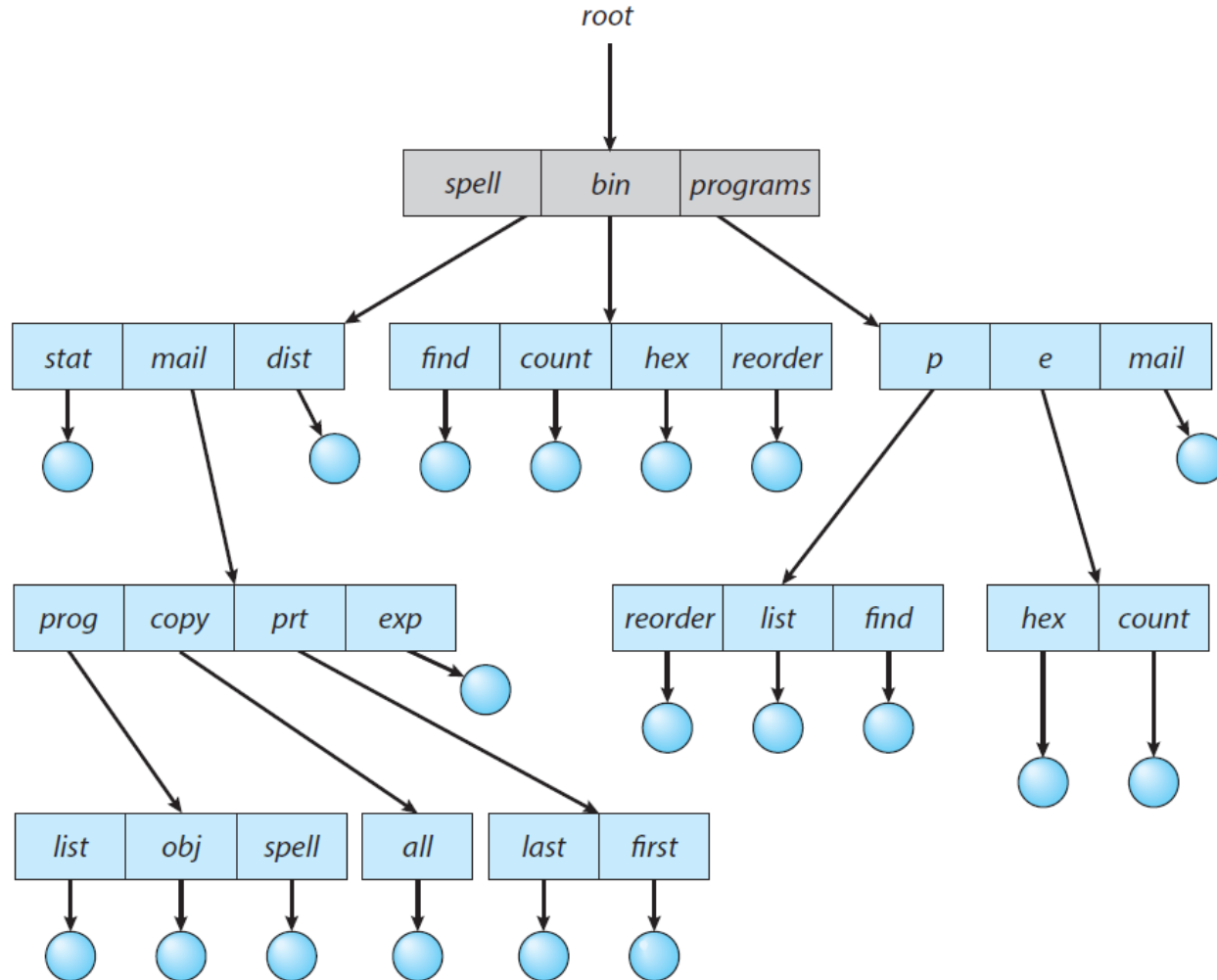
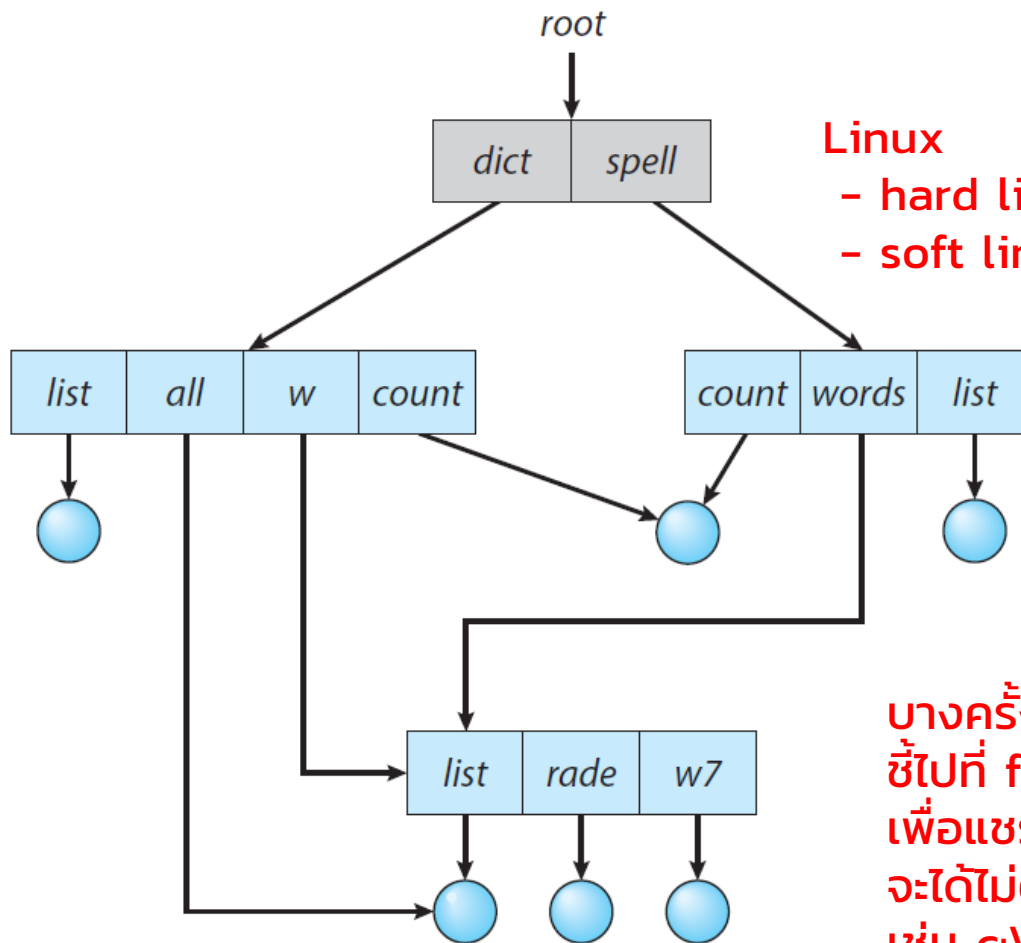


Figure 13.9 Tree-structured directory structure.

Users can manage their own subdirectories.

# Acyclic-Graph Directory

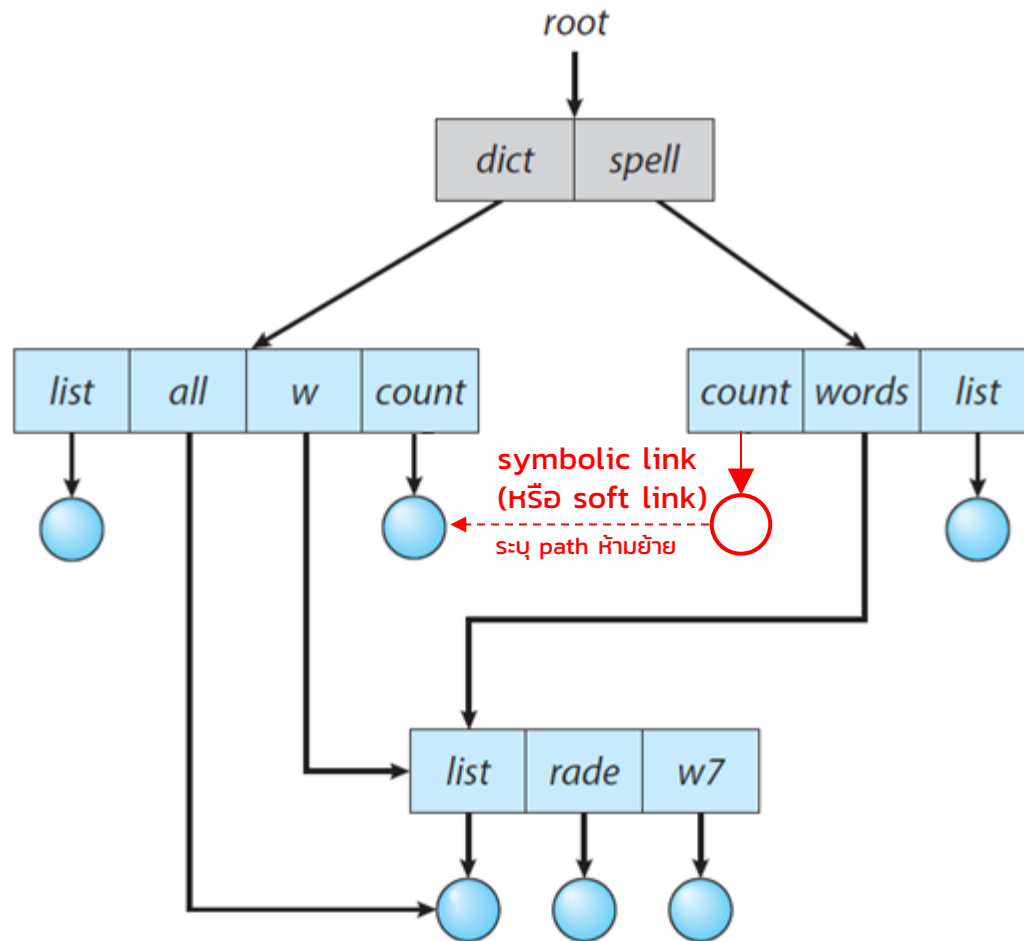


## Linux

- hard link ใช้คำสั่ง `ln` (รูปนี้)
- soft link ใช้คำสั่ง `ln -s` (รูปถัดไป)

บางครั้งเราก็ต้องการให้ 2 folders/files  
ชี้ไปที่ folder/file เดียวกัน เช่น  
เพื่อแชร์ระหว่าง 2 แอปพลิเคชัน  
จะได้ไม่ต้องมี 2 ก๊อปปี้ หรือเพื่อความสะดวก  
เช่น `c:\doc` ชี้ไปที่ `My Documents`

Figure 13.10 Acyclic-graph directory structure.





In a nutshell, a soft link

- can cross the file system,
- allows you to link between directories,
- has different inode number and file permissions than original file,
- permissions will not be updated,
- has only the path of the original file, not the contents.

A hard Link

- can't cross the file system boundaries (i.e. A hardlink can only work on the same filesystem),
- can't link directories,
- has the same inode number and permissions of original file,
- permissions will be updated if we change the permissions of source file,
- has the actual contents of original file, so that you still can view the contents, even if the original file moved or removed.

# General Graph Directory

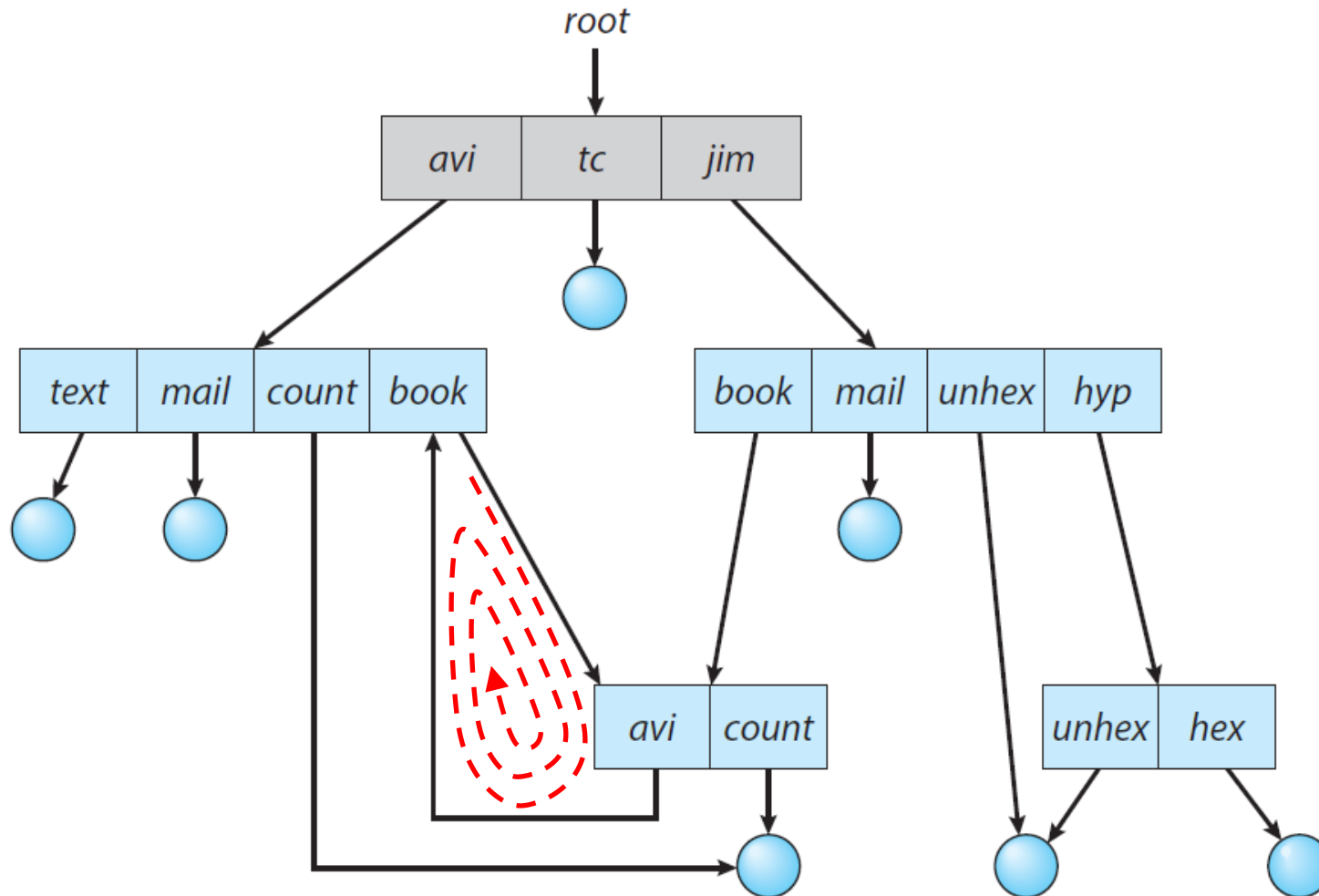


Figure 13.11 General graph directory.

มี loop ได้ อาจจะเกิดจากไวรัสหรืออุบัติเหตุ เช่นทำ soft linkพลาด

# File-System Mounting

A file system must be **mounted** before it can be available to processes.

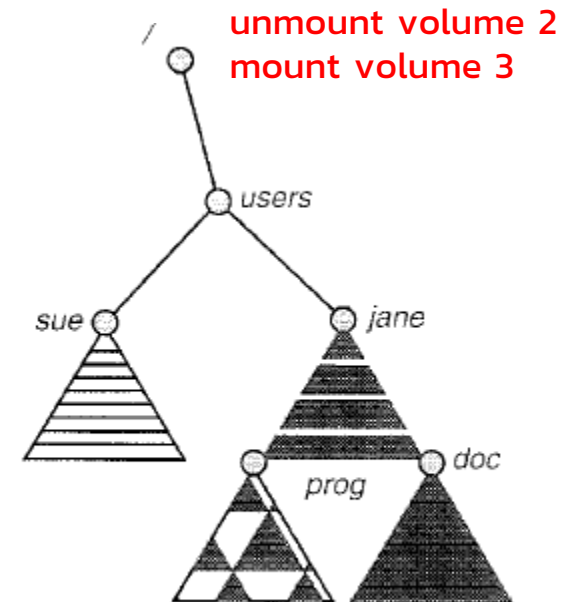
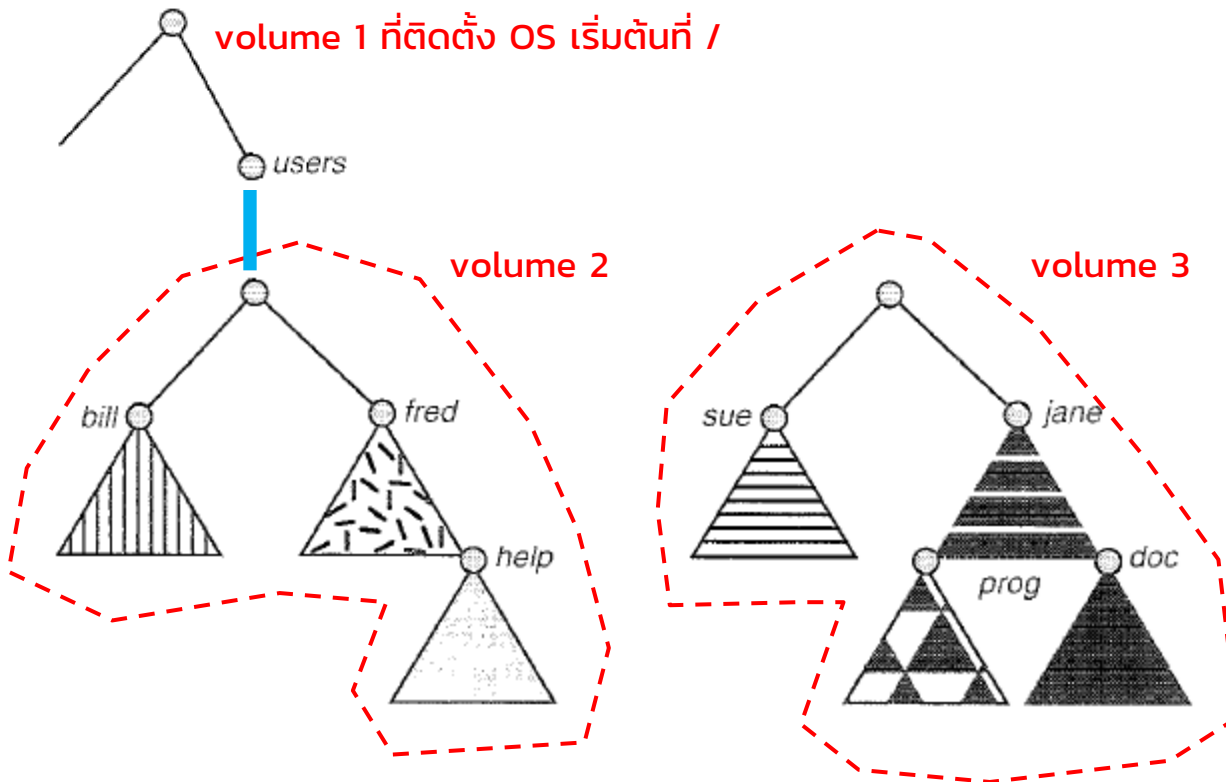


Figure 10.13 Mount point.

## ตัวอย่างการใช้ flash drive บน Linux

พอเสียบ flash drive แล้ว OS จะ mount ให้อัตโนมัติ  
เช่น อยู่ที่ **/media/flashdrive\_name**  
Windows ให้อยู่ที่ **drive letter** ใหม่

คำสั่ง **mount** เป็นของ Linux  
และ Windows 7 ขึ้นไป ทำให้  
Windows ไม่ถูกจำกัดจำนวน  
ดิสก์แล้ว เดิมมีดิสก์ได้แค่ A-Z  
เพราะต้องเป็น drive letter  
เท่านั้น ปัจจุบัน mount ที่  
โฟลเดอร์ก็ได้

## រៀប disk ថ្មី (volume ថ្មី) ប្រើ Windows

New Simple Volume Wizard

**Assign Drive Letter or Path**  
For easier access, you can assign a drive letter or drive path to your partition.

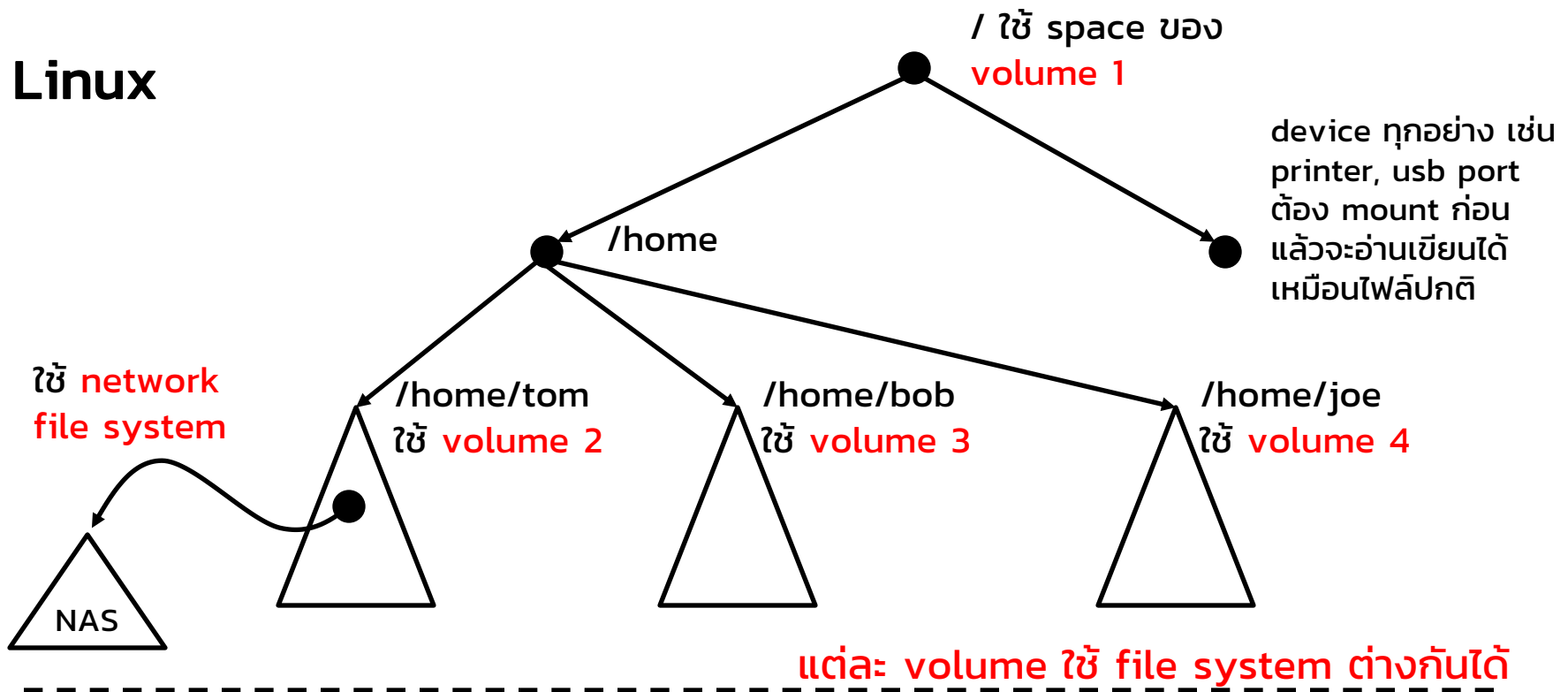
☒ Assign the following drive letter: D

☐ Mount in the following empty NTFS folder:  
Browse...

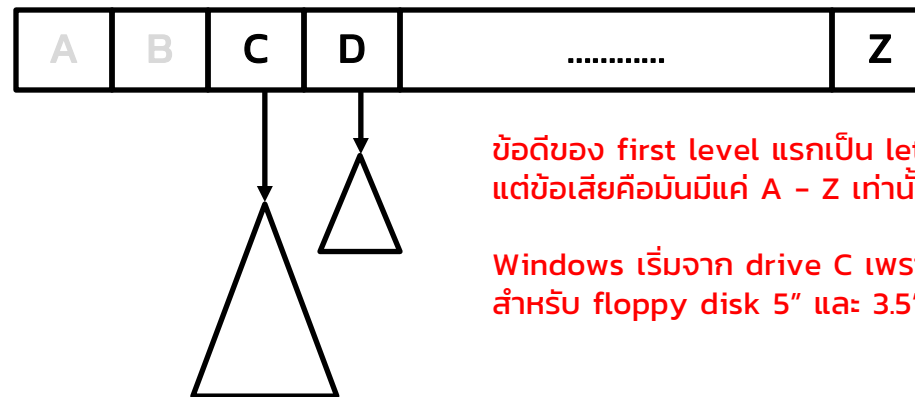
☐ Do not assign a drive letter or drive path

< Back Next > Cancel

# Linux



# Windows



ข้อดีของ first level แรกเป็น letter คือ user-friendly แต่ข้อเสียคือมันมีแค่ A - Z เท่านั้น เพิ่มไม่ได้ (26 drives)

Windows เริ่มจาก drive C เพราะยุคแรกมี drive A, B สำหรับ floppy disk 5" และ 3.5" แล้วเลิกใช้ไป

Windows: maintaining an extended two-level directory

MacOS X: all file systems are mounted under /Volumes

# File Sharing

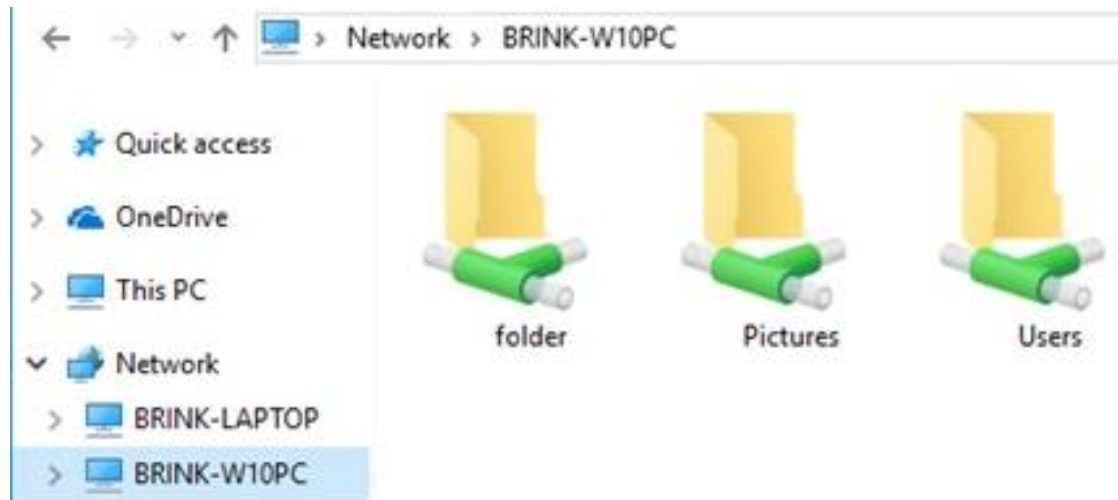
Remote file systems

Windows

sharing files via network

Linux

mount to a file system on another Linux

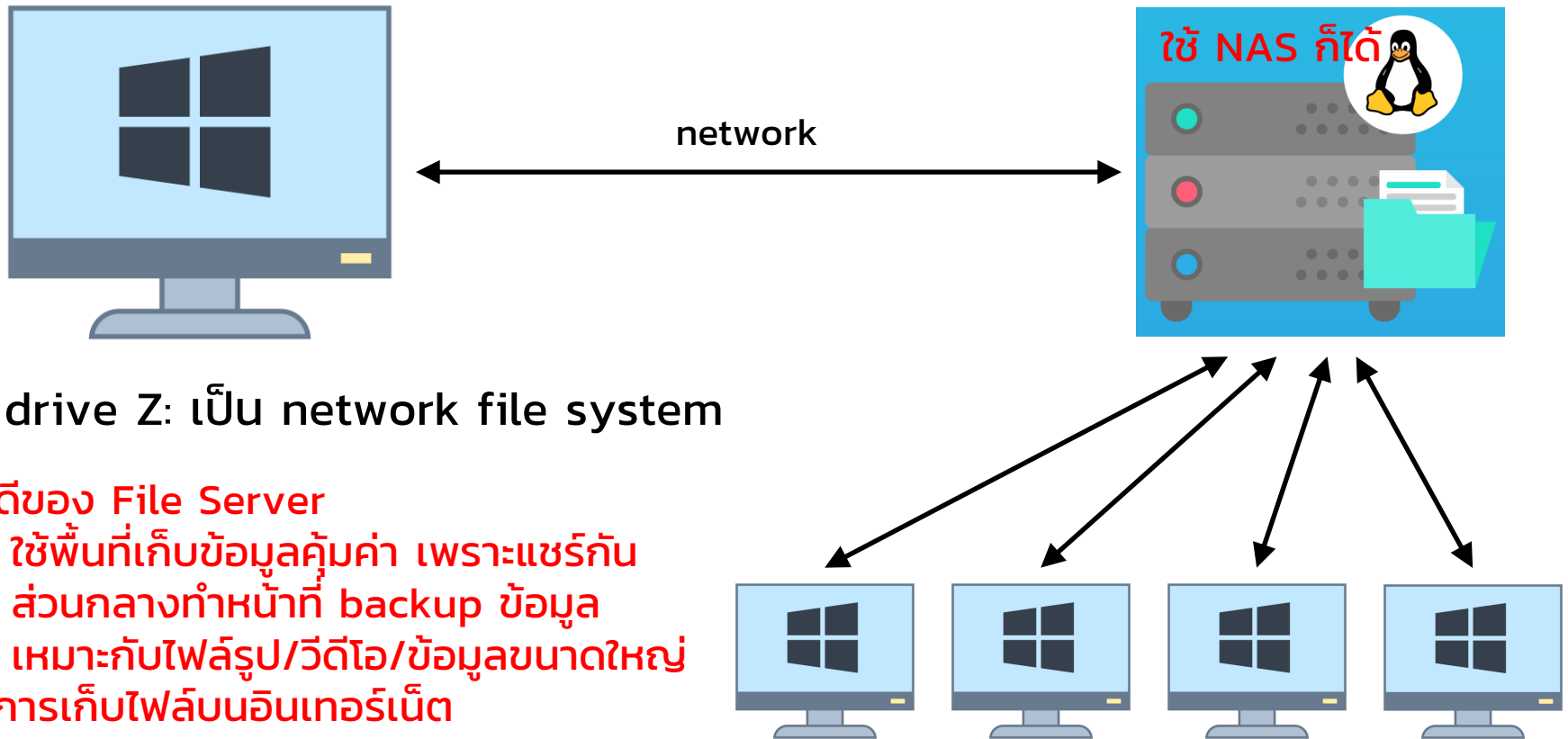


Linux (simplified)

`mount //ip-address/share /mnt/data`

# Network File System

## SMB File Server (SAMBA)

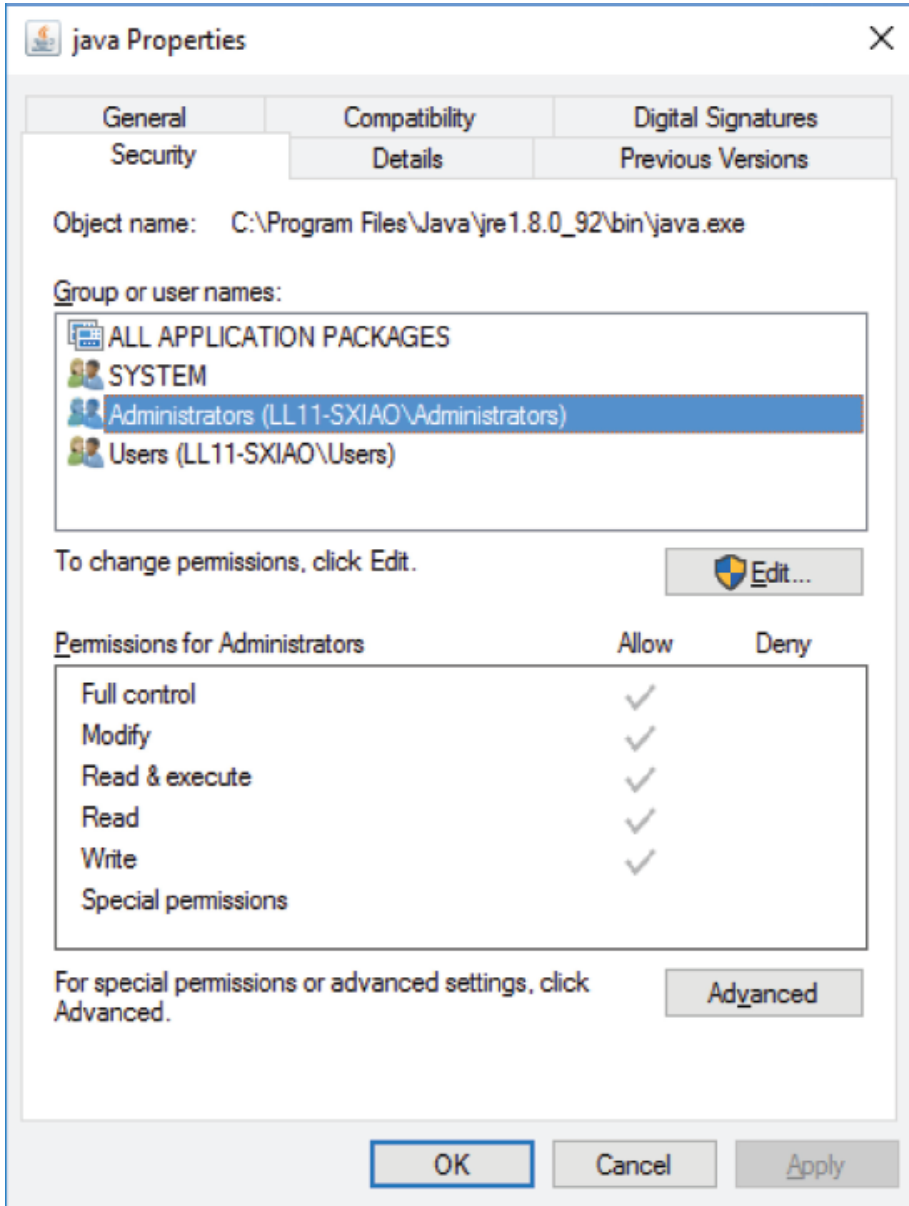


ให้ drive Z: เป็น network file system

### ข้อดีของ File Server

- ใช้พื้นที่เก็บข้อมูลคุ้มค่า เพราะแชร์กัน
  - ส่วนกลางทำหน้าที่ backup ข้อมูล
  - เหมาะกับไฟล์รูป/วิดีโอ/ข้อมูลขนาดใหญ่
- บริการเก็บไฟล์บนอินเทอร์เน็ต
- Google Drive, One Drive
  - เหมาะกับไฟล์เอกสารที่มีขนาดไม่ใหญ่มาก

# Protection



-rw-rw-r--	1	pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2	jwg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1	pbg	staff	9423	Feb 24 2017	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb 24 2017	program
drwx--x--x	4	tag	faculty	512	Jul 31 10:31	lib/
drwx-----	3	pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul 8 09:35	test/

**d** **rw****x** **rw****x** **rw****x**

**Owner** **Group** **Others**

**Directory**

ใน Linux มีคำสั่ง **chown**, **chgrp**, **chmod**

**mode 755** คือ 111 101 101

**mode 644** คือ 110 100 100

**mode 700** คือ 111 000 000

**mode 400** คือ 100 000 000

วัตถุประสงค์ของแต่ละ **mode** คืออะไร?



## Chapter 13 File-System Interface

13.1 File Concept 529

13.2 Access Methods 539

13.3 Directory Structure 541

13.4 Protection 550

13.5 Memory-Mapped Files 555

13.6 Summary 560

Practice Exercises 560

Further Reading 561