



---

# Hệ thống tập tin (Files System)



# Nội dung

---

(Phần A)

- Giao diện hệ thống file
  - Các yêu cầu của ứng dụng
  - File
  - Thư mục
  - File system mounting
  - Chia sẻ & bảo vệ



# Lưu trữ thông tin cho ứng dụng

---

- Các yêu cầu của ứng dụng
  - Tính bền vững (Persistence)
  - Tốc độ (Speed)
  - Dung lượng (Size)
  - Chia sẻ/Bảo vệ (Sharing/protection)
  - Dễ dàng sử dụng (Ease of use)



# Lưu trữ thông tin cho ứng dụng

Được hỗ trợ bởi

- Phần cứng

- **Persistence**: các thiết bị lưu trữ bền vững (non-volatile memory)
- **Speed**: cung cấp khả năng truy cập ngẫu nhiên, nâng cao tốc độ đĩa (5400 → 7200 → 10K → 15K rpm)
- **Size**: dung lượng lớn, 20M → 700M → 40G → 200G → 1T byte



# Lưu trữ thông tin của ứng dụng

- Hệ điều hành
  - **Persistence**: lưu trữ dư thừa, back-up phục vụ cho recovery
    - Ví dụ: RAID (Redundant Array of Independent Disks)
  - **Ease of use**:
    - Gán tên cho một chuỗi các khối dữ liệu → **file**
    - Hỗ trợ quản lý các file → **thư mục**
    - User không cần quan tâm đến các chi tiết hardware (giao tiếp với disk controller theo IDE, SCSI,...)
  - **Sharing/Protection**: cho phép thiết lập các quyền truy cập file/thư mục



# Định nghĩa file

---

- File là một chuỗi các byte
  - được đặt tên,
  - persistent,
  - các tác vụ lên một file gồm ít nhất là
    - read
    - write



# Định nghĩa file

- “The **file**, an abstraction in operating systems, is a sequence of bytes with a name. Users are allowed to do only two things with a file: read and write. **Read** means to copy the file’s bytes into the user’s workspace. **Write** means to replace the contents of the file with a new sequence of bytes from the user’s workspace.” [Denning]



# Các loại file

---

- Các loại file
  - regular file
    - ASCII
    - binary
  - directory
  - special files (trong UNIX): character special file, block special file,...





# Các kiểu file

---

- Cách phân biệt kiểu file
  - extension
  - magic number

# Các kiểu file thông dụng

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, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information



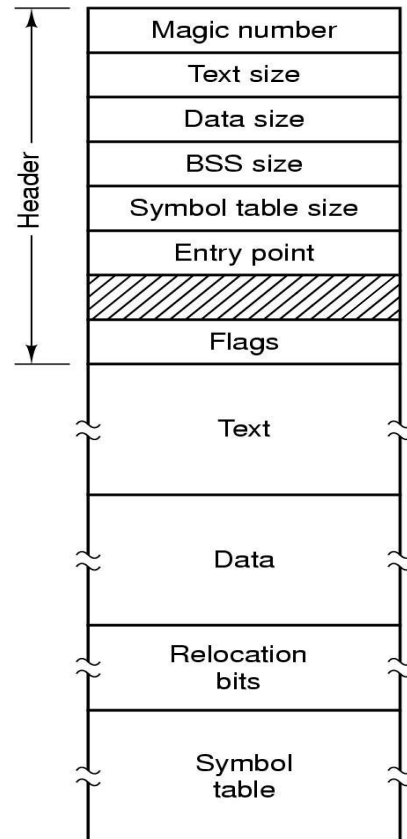
# Định dạng file

---

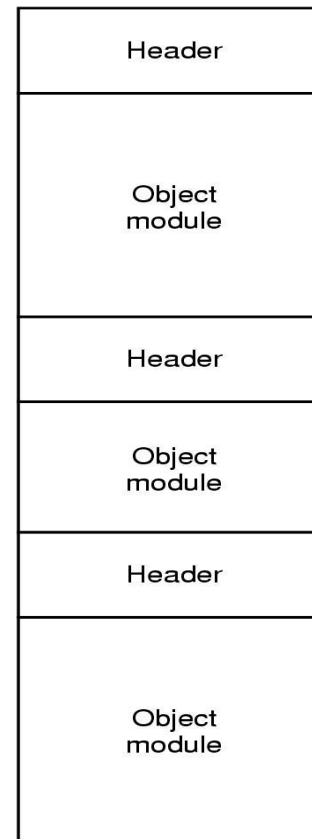
- Ứng dụng có thể lưu và diễn dịch dữ liệu trong file theo định dạng riêng
  - MS Word
  - PDF
  - ELF
  - ...

# Định dạng file (tt.)

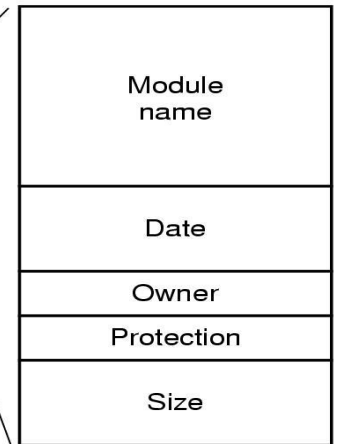
■ Ví dụ trong  
UNIX



Executable file

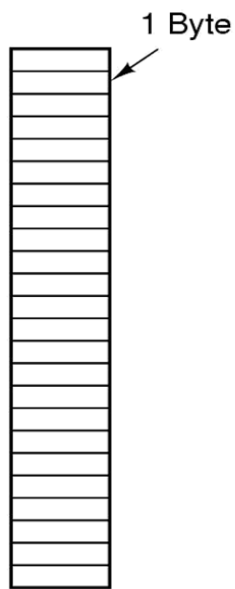


Archive

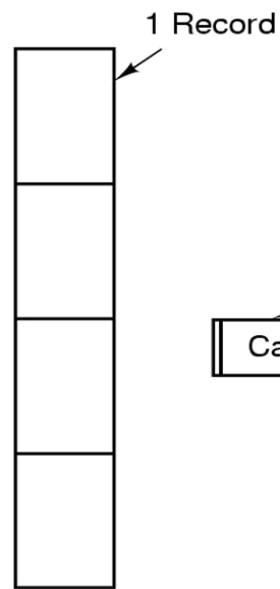


# Cấu trúc file

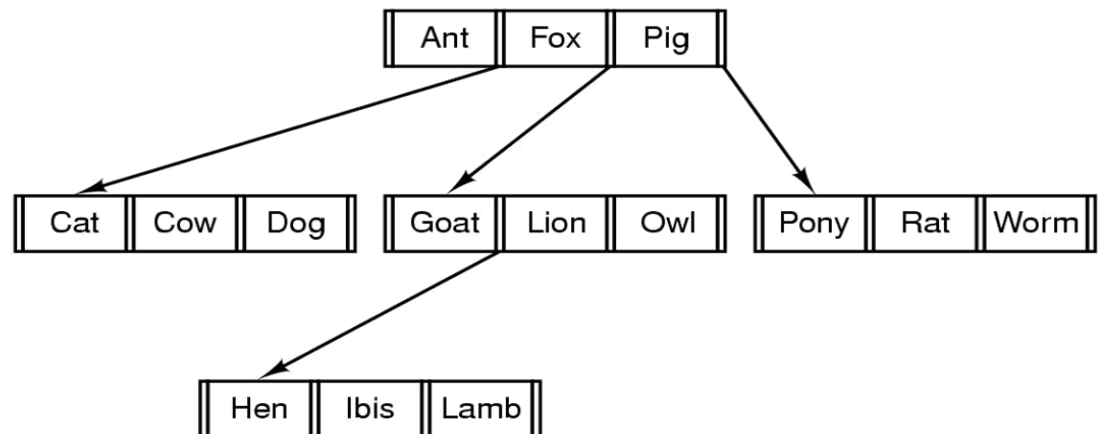
- Cấu trúc file: mô hình file để người dùng lập trình
  - Không có cấu trúc: một chuỗi byte (Unix, DOS, Windows), Hình a
  - Cấu trúc record
    - Fixed length, Hình b
    - Variable length: hỗ trợ tìm nhanh chóng một record với key cho trước (IBM mainframe), Hình c



(a)



(b)



(c)



# Các cách truy cập file

- Dùng các tác vụ thông thường (read, write) trên file,
  - truy cập **tuần tự** (sequential access)
    - truy xuất dữ liệu theo thứ tự, từng byte hoặc record
  - truy cập **bất kỳ** (random access)
    - truy xuất dữ liệu, byte hoặc record, tại vị trí bất kỳ
  - truy cập **dùng khóa**
    - truy xuất một record dựa vào key của nó
- **Memory-mapped** file
  - Giả sử có các system call map và umap
  - Gọi map để “chiếu” file vào không gian địa chỉ ảo của quá trình
  - Dùng các tác vụ truy xuất bộ nhớ để truy xuất file
  - Gọi umap khi xong



# Các thuộc tính của file

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to



# Các tác vụ trên file

- **Create:** Tạo một file mới
- **Write:** Thực hiện tác vụ ghi dữ liệu vào file tại vị trí con trỏ ghi
- **Read:** Thực hiện tác vụ đọc dữ liệu từ file tại vị trí con trỏ đọc
- **Reposition:** Thiết lập con trỏ đọc/ghi đến vị trí do quá trình chỉ định
- **Delete:** Xóa file
- **Truncate:** Giữ lại tất cả các thuộc tính của file, ngoại trừ kích thước file được thiết lập về 0
- **Open:** Quá trình phải mở file trước khi sử dụng
- **Close:** Quá trình phải đóng file sau khi sử dụng





# Thư mục

---

- **Thư mục** (directory, folder) là một tiện ích của hệ điều hành để người dùng dễ dàng tổ chức các file của mình.



# Các tác vụ trên thư mục

---

- **Create:** Tạo một directory mới
- **Delete:** Xóa một directory trống
- **Opendir:** Quá trình phải mở directory trước khi đọc nó
- **Closedir:** Quá trình phải đóng directory sau khi đọc nó
- **Readdir:** Đọc entry tới của directory
- **Link**
- **Unlink**

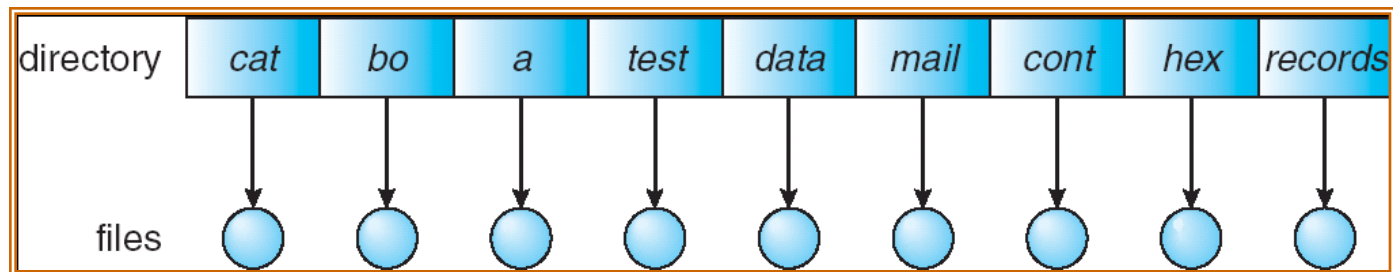


# Các loại cấu trúc của thư mục

- Từ đơn giản đến phức tạp, tùy hệ thống
  - Single-level directory
  - Multi-level directory
  - Tree-structured directory
  - Acyclic-graph directory
  - General graph directory

# Các loại cấu trúc của thư mục

- Single-level directory
  - Chỉ có một directory cho mọi file trong hệ thống (CDC 6600, IBM MVS, PalmOS)



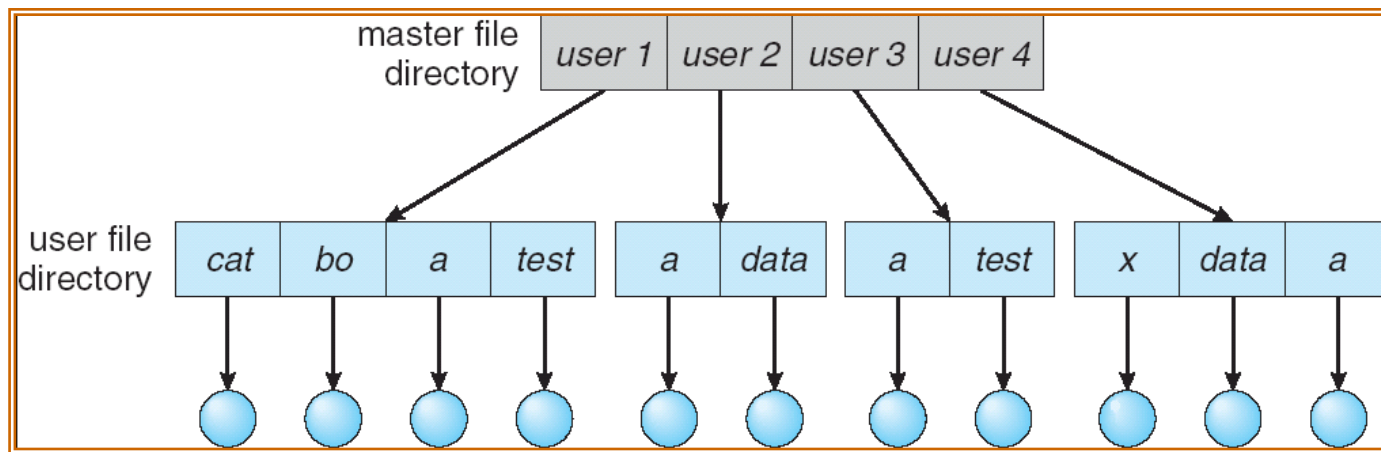


# Cấu trúc của thư mục (tt.)

- Cải tiến cấu trúc single-level directory
- Mục tiêu:
  - Thuận tiện cho user khi đặt tên file, các user khác nhau có thể đặt tên file trùng nhau
  - File có thể có nhiều hơn một tên
  - Hỗ trợ phân nhóm (grouping): tổ chức các files cùng đặc điểm vào chung một nhóm (ví dụ nhóm file mã nguồn ngôn ngữ C, nhóm file Word,...)

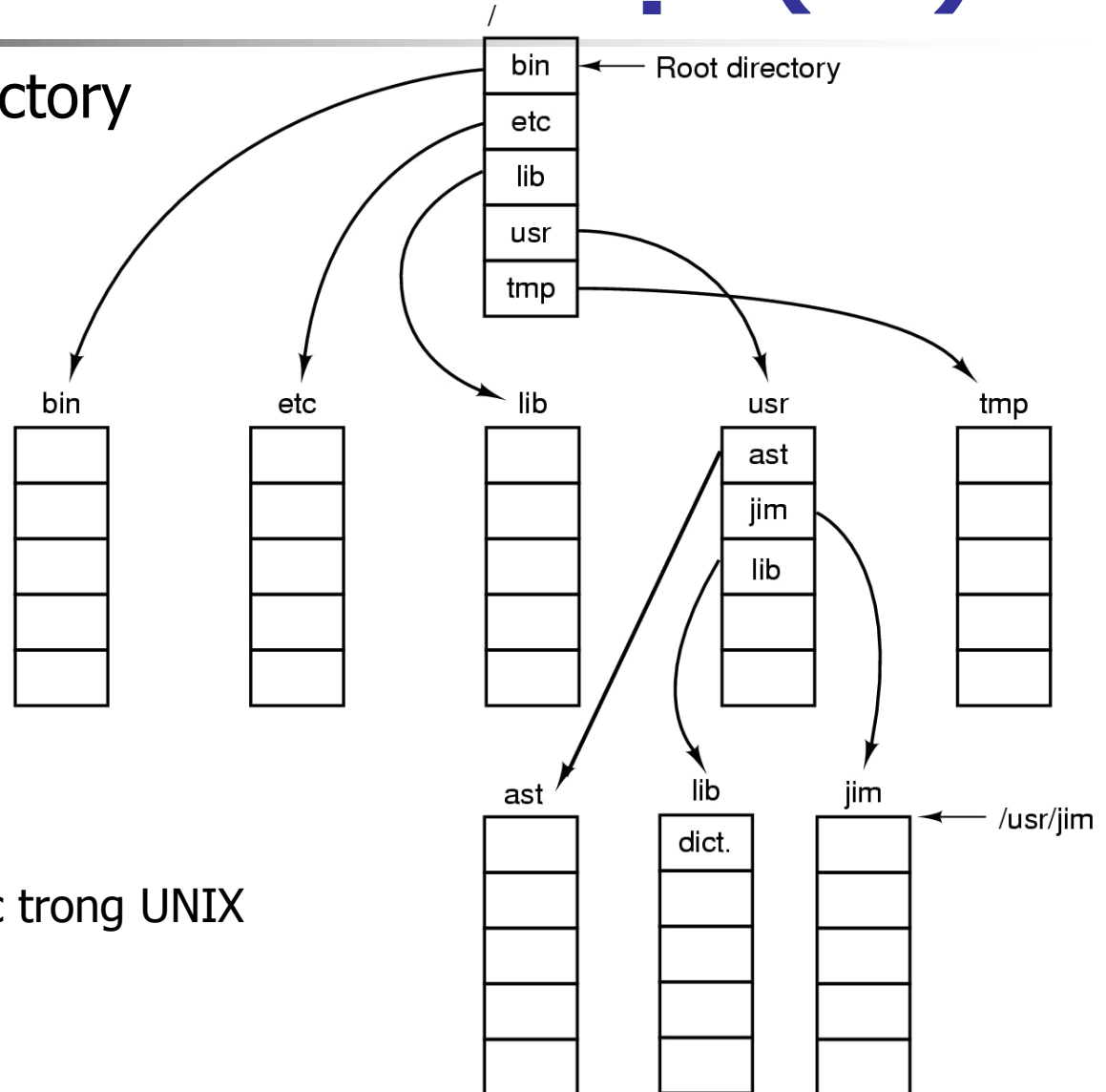
# Cấu trúc của thư mục (tt.)

- Two-level directory



# Cấu trúc của thư mục (tt.)

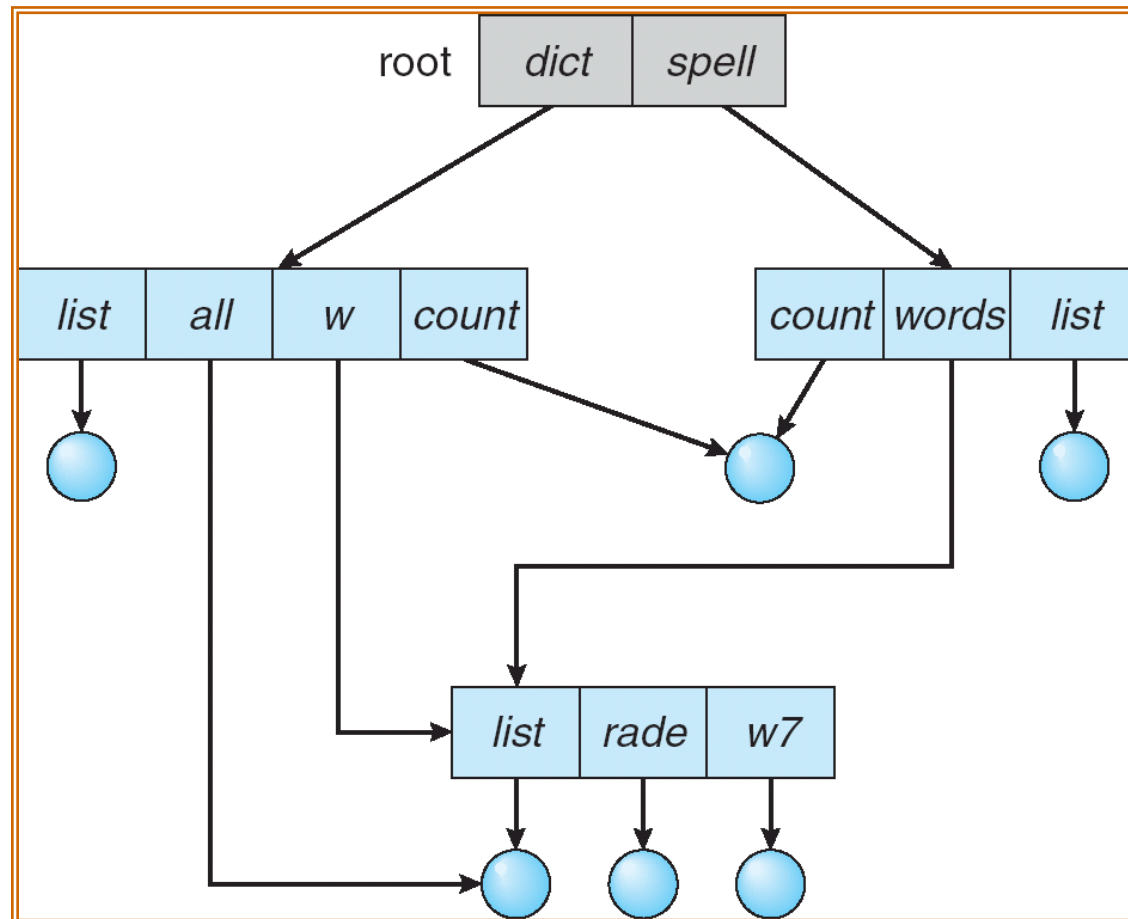
## ■ Tree-structured directory



Một cây thư mục trong UNIX

# Cấu trúc của thư mục (tt.)

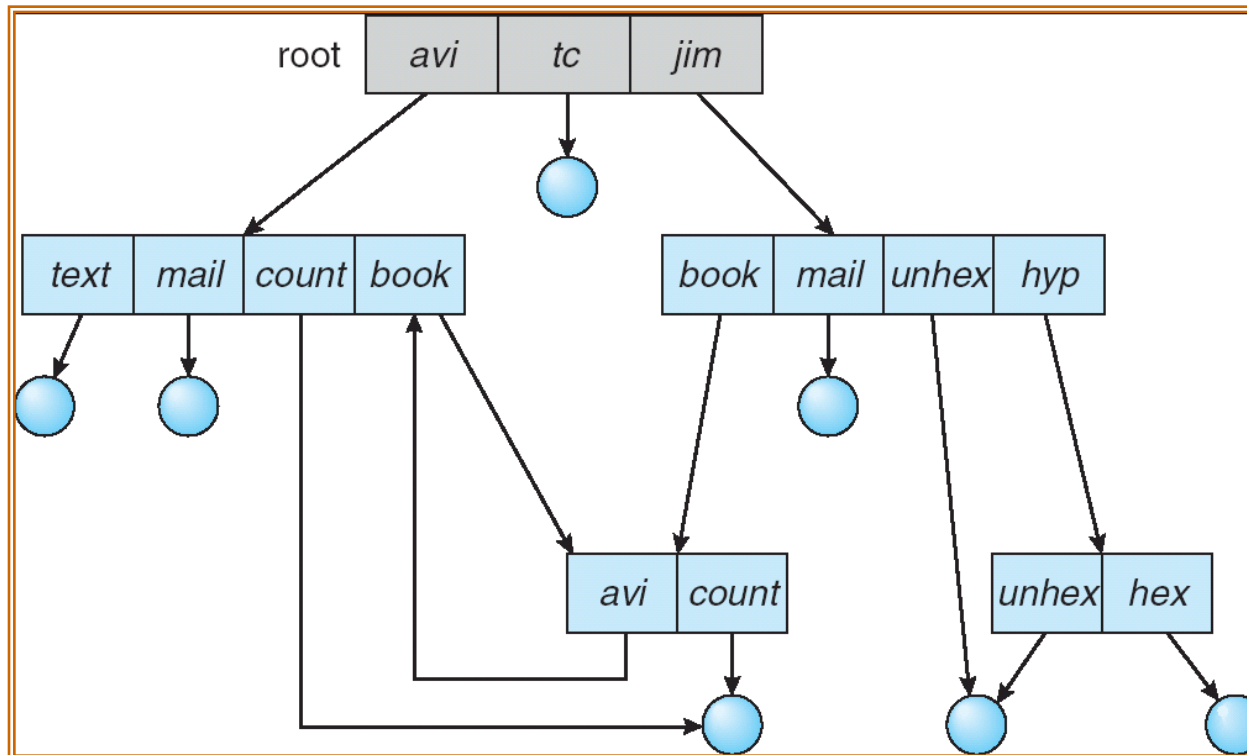
- Acyclic-graph directory





# Cấu trúc của thư mục (tt.)

- General-graph directory
  - Vấn đề: cycle có thể xuất hiện
    - Dùng giải thuật kiểm tra chu trình mỗi khi tạo file?





# Định vị file trong cây/rừng thư mục

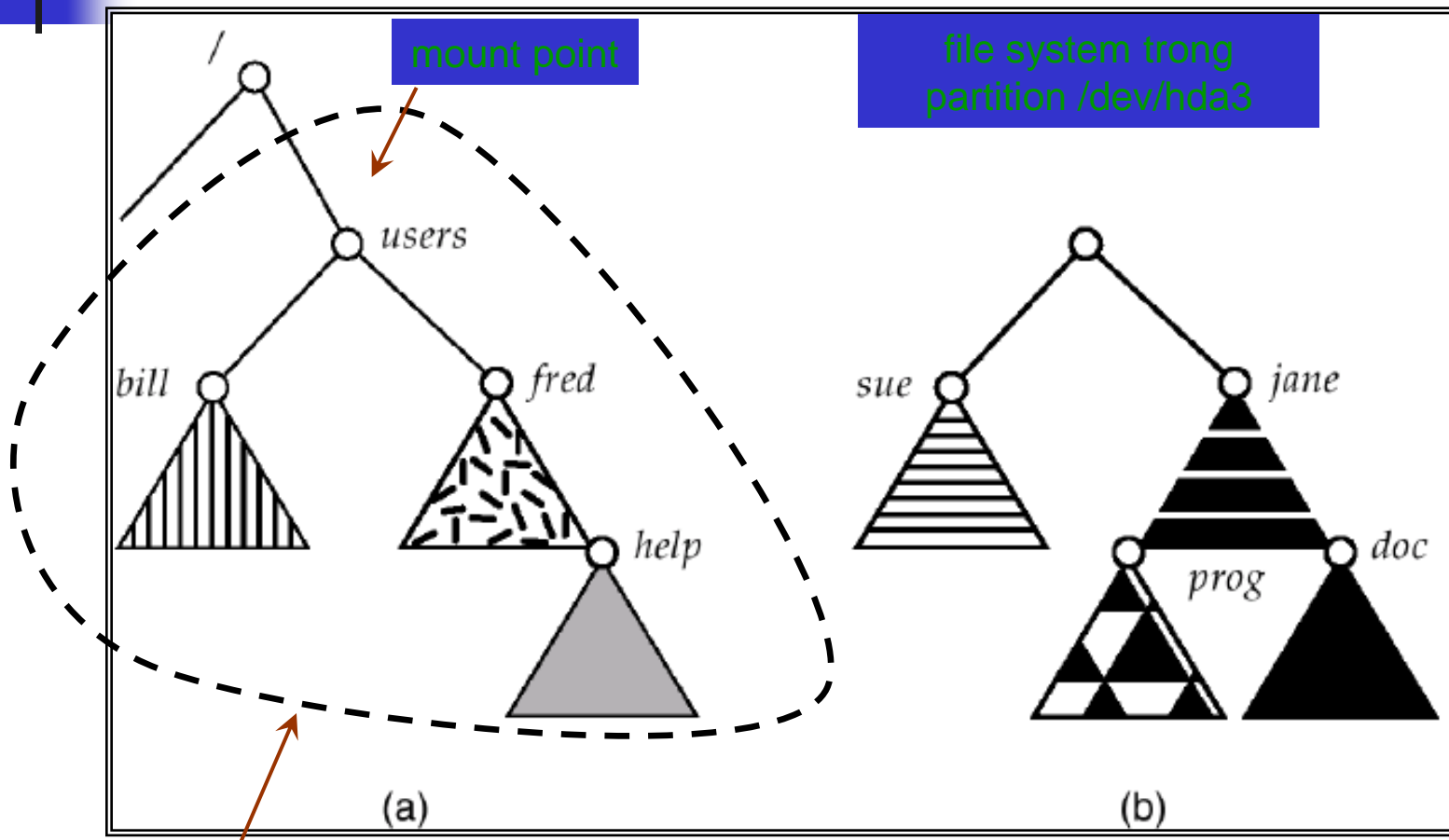
- Dùng **đường dẫn** (path name) đến một file để đặc tả vị trí của file trong cây hay rừng directory
  - Ví dụ
    - UNIX, LINUX `/usr/ast/mailbox`
    - DOS, Windows `\usr\ast\mailbox`
- Đường dẫn **tuyệt đối**
- **Thư mục hiện thời** (current directory, working directory)
- Đường dẫn **tương đối**



# Sử dụng hệ thống files

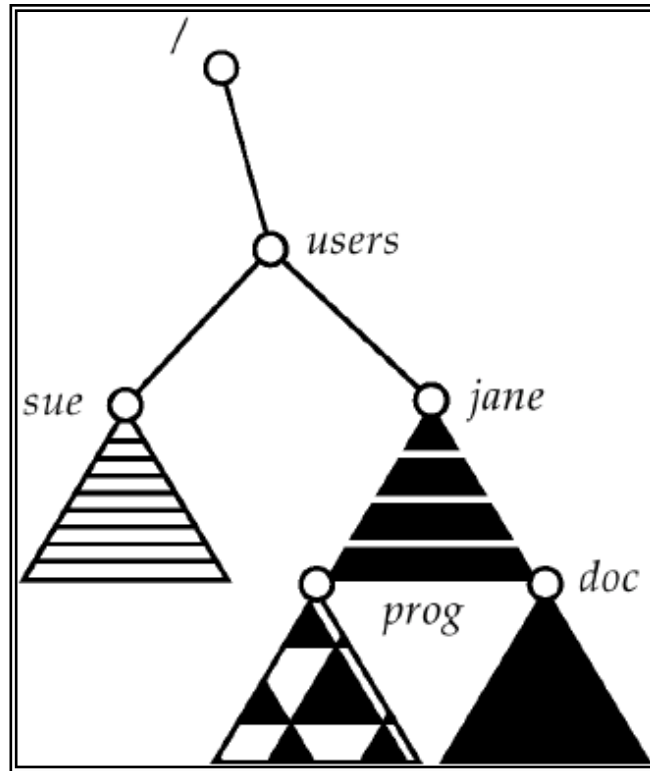
- Phải **gắn** (mount) thiết bị (partition,...) chứa file system để ứng dụng có thể truy cập file của nó
  - Thư mục để gắn vào đó một file system được gọi là **mount point**
- Để thực hiện lệnh gắn
  - `"# mount /device/thedisk /mountpoint"`
  - hệ điều hành phải thực thi các bước:
    - OS kiểm tra xem file system có "hợp lệ" hay không
    - Gắn file system vào vị trí mountpoint
- Nếu mount point có chứa file?
  - Các file chứa trong thư mục mount point sẽ bị che cho đến khi unmount file system.

# File system mounting



Các file bị che khi file system trong partition /dev/hda3 được mount với lệnh  
`mount /dev/hda3 /users`

# File system mounting (tt.)



Sau khi mount, có thể truy cập các file trong partition /dev/hda3



# Chia sẻ file

- Nhu cầu chia sẻ file trong hệ thống multiuser
- Việc chia sẻ phải thực hiện dưới sự bảo vệ và kiểm soát nghiêm ngặt
- Chia sẻ file
  - Máy đơn: **hard link** và **soft link**
  - Mạng máy tính: ví dụ dùng giao thức **NFS** (Network File System)



- In **-s** /spell/words/list /dict/all



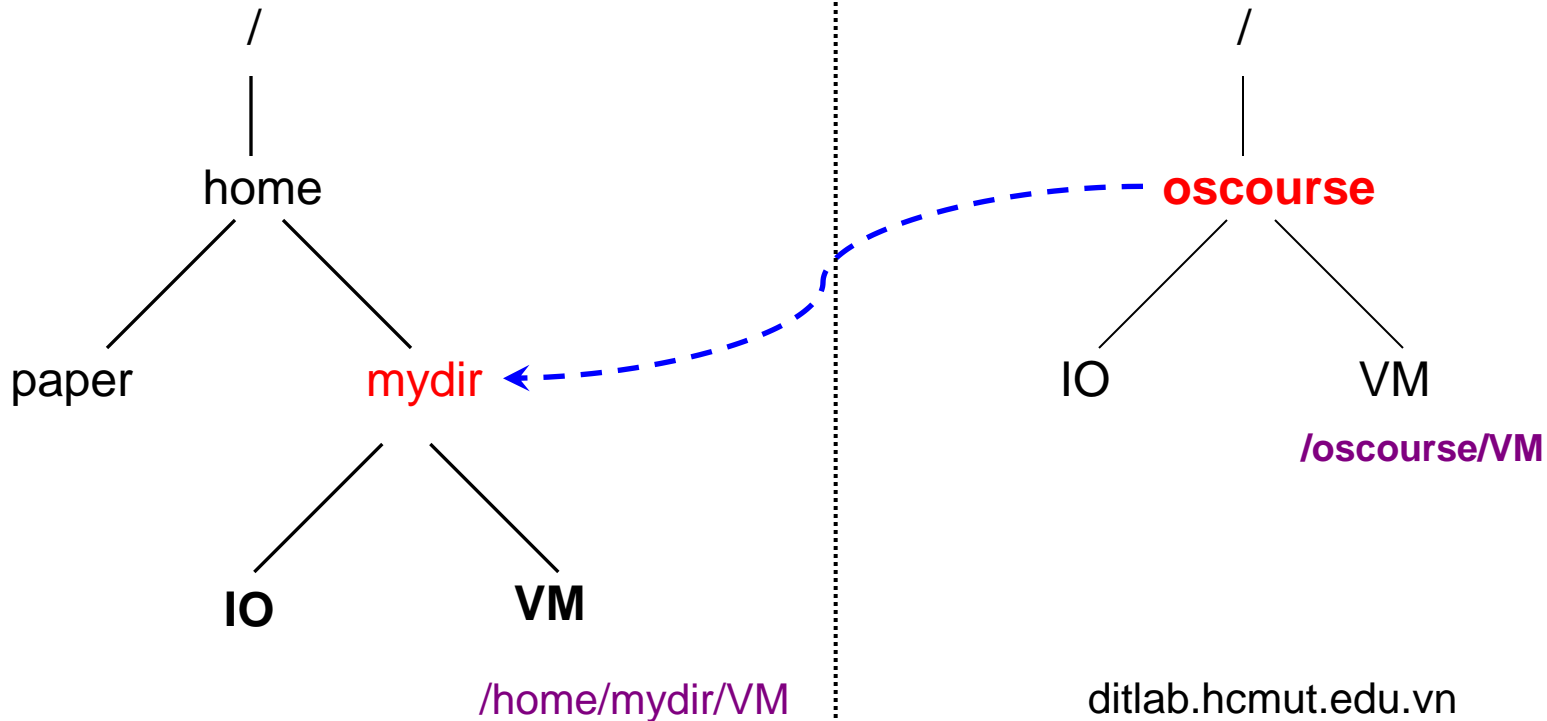
# NFS (*Network File System*)

---

- Trong mạng máy tính
  - **Network File System** (NFS) là phương pháp chia sẻ file rất phổ biến dùng RPC (mô hình client-server)
    - File system từ xa phải được gắn (mount) trước khi sử dụng. Ví dụ:  
"mount ditlab.hcmut.edu.vn:/oscourse /home/mydir"
    - Khi file system đã được mount, mọi truy xuất đến file hay thư mục trên đó như bình thường, user không phân biệt truy cập file/directory là local hay remote.



# mount remote file system



Local Client

Remote Server

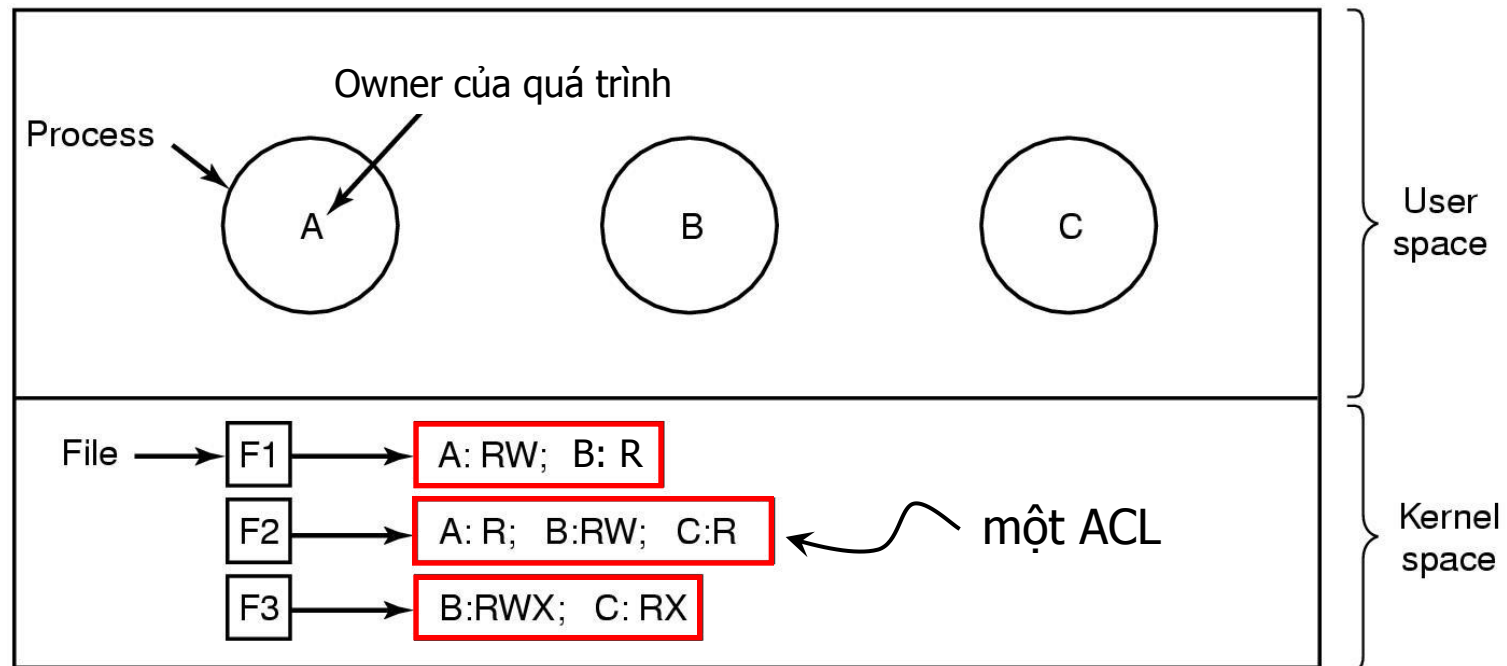
ditlab.hcmut.edu.vn



# Bảo vệ file (protection)

- Hệ điều hành phải hỗ trợ chủ nhân của file trong việc kiểm soát truy cập file
  - Các tác vụ có thể thực hiện trên file?
  - Những ai được quyền thực hiện thao tác trên file?
- Các **quyền** truy cập file
  - Read
  - Write
  - Execute, Append, Delete,...
- Cách tiếp cận thông thường
  - Mỗi file có một **Access Control List** (ACL), gồm các cặp  $\langle \text{user}, \text{rights} \rangle$ 
    - Windows NT/2K/XP, Linux
    - user có thể là một nhóm

# Access Control List



File F1:

- Mọi quá trình của user A có quyền đọc/ghi
- Mọi quá trình của user B có quyền đọc



# Các Access Control Bit

- Các thao tác: read, write, execute
- Ba nhóm user: **owner**, **group**, **other** (public)
  - User name và user identifier (user ID)
  - Group name và group identifier (group ID)
- Ví dụ trong UNIX: mỗi nhóm user có một field, mỗi field có 3 bit
  - Field:    read | write | execute
  - Trị:        0/1        0/1        0/1

			r	w	x
owner access	7	⇒	1	1	1
group access	6	⇒	1	1	0
public access	1	⇒	0	0	1