

Hệ thống file

- 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
 - Persistence
 - Speed
 - Size
 - Sharing/protection
 - Ease of use

Lưu trữ thông tin cho ứng dụng

Được hỗ trợ bởi

- **Đĩa từ (HD, hard disk drive)**
 - **Persistence**: 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
- **SSD (solid-state drive)**
 - Tốc độ đọc/ghi có thể đến hàng trăm MB/s
 - Công nghệ
 - ▶ Flash-based
 - ▶ DRAM-based: được trang bị thêm pin trong
 - Giao diện: tương tự với giao diện của đĩa từ
 - ▶ Serial ATA (SATA)
 - ▶ Serial Attached SCSI (SAS)
 - ▶ USB

HDD vs SSD

Attribute	SSD (Solid State Drive)	HDD (Hard Disk Drive)
Power Draw / Battery Life	Less power draw, averages 2 – 3 watts, resulting in 30+ minute battery boost	More power draw, averages 6 – 7 watts and therefore uses more battery
Cost	Expensive, roughly \$0.50 per gigabyte (based on buying a 1TB drive)	Only around \$0.15 per gigabyte, very cheap (buying a 4TB model)
Capacity	Typically not larger than 512GB for notebook size drives; 1TB max for desktops	Typically around 500GB and 2TB maximum for notebook size drives; 4TB max for desktops
Operating System Boot Time	Around 22 seconds average bootup time	Around 40 seconds average bootup time
Noise	There are no moving parts and as such no sound	Audible clicks and spinning can be heard

HDD vs SSD

Attribute	SSD (Solid State Drive)	HDD (Hard Disk Drive)
Vibration	No vibration as there are no moving parts	The spinning of the platters can sometimes result in vibration
Heat Produced	Lower power draw and no moving parts so little heat is produced	HDD doesn't produce much heat, but it will have a measurable amount more heat than an SSD due to moving parts and higher power draw
Failure Rate	Mean time between failure rate of 2.0 million hours	Mean time between failure rate of 1.5 million hours

HDD vs SSD

Attribute	SSD (Solid State Drive)	HDD (Hard Disk Drive)
Encryption	Full Disk Encryption (FDE) Supported on some models	Full Disk Encryption (FDE) Supported on some models
File Opening Speed	Up to 30% faster than HDD	Slower than SSD
Magnetism Affected?	An SSD is safe from any effects of magnetism	Magnets can erase data

http://www.storagereview.com/ssd_vs_hdd

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]

Khái niệm file trong UNIX

- Ngoài **regular** file như vừa trình bày, UNIX còn xem là ‘file’:
 - **directory**
 - **special file**
 - ▶ Hầu hết mọi thiết bị đều có một special file liên kết với nó. Ví dụ: special file cho chuột có tên là **/dev/mouse**
 - ▶ Character special file, block special file,... cho phép ứng dụng tương tác với device driver được đặc tả bởi file đó
 - ▶ Chuyện gì xảy ra nếu thực thi “cat /dev/mouse” rồi di chuyển chuột?

Các kiểu file

- Cách phân biệt kiểu file (file type)
 - extension
 - ▶ “.txt”, “.exe”, “.ppt”, ...
 - ▶ Windows dựa trên extension để chạy ứng dụng tương ứng lên file
 - ▶ Trong môi trường UNIX/Linux, extension thường chỉ được xem là một phần của tên file
 - magic number
 - ▶ File GIF bắt đầu bằng chuỗi ASCII “GIF87a” hay “GIF89a”
 - ▶ File PDF bắt đầu bằng chuỗi ASCII “%PDF”

Các kiểu file thông dụng

- Như đã nói, extension được Windows sử dụng để kích hoạt ứng dụng tương ứng

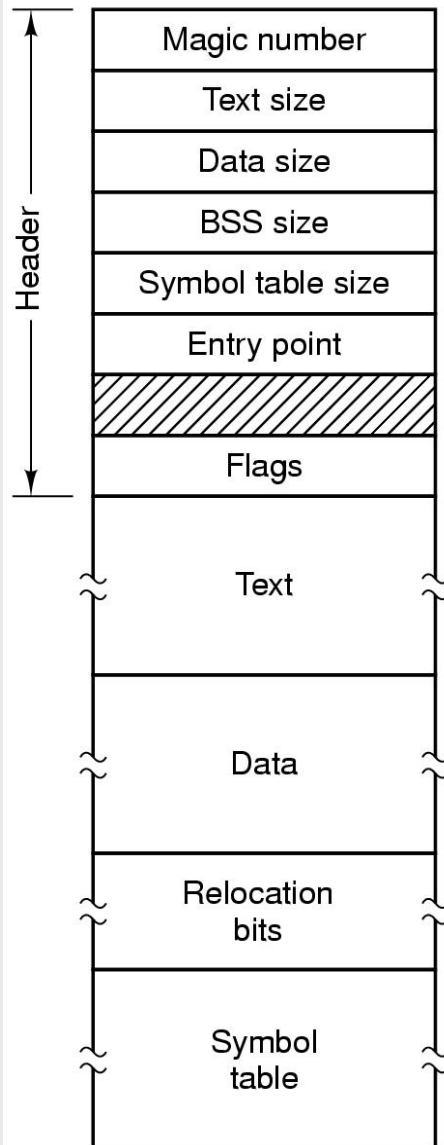
file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual 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 (1/2)

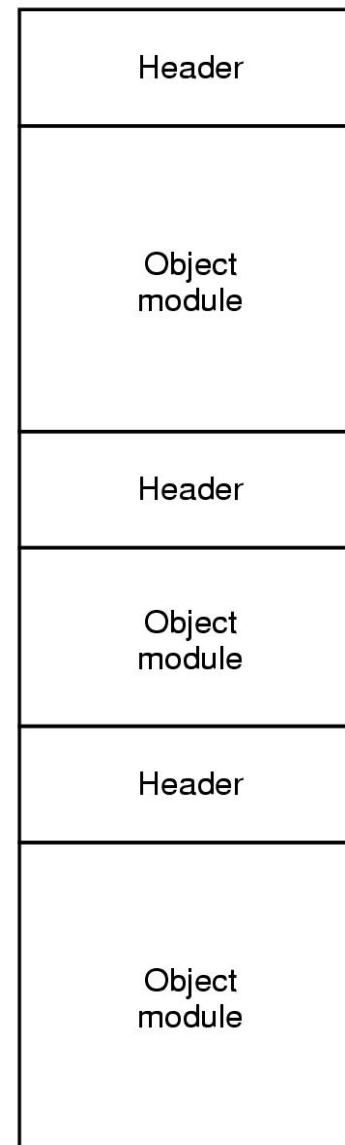
- Ứng dụng có thể lưu và diễn dịch dữ liệu trong file theo định dạng riêng
 - MS Word
 - Adobe Acrobat
 - ...

Định dạng file (2/2)

■ Vd UNIX:



Executable file

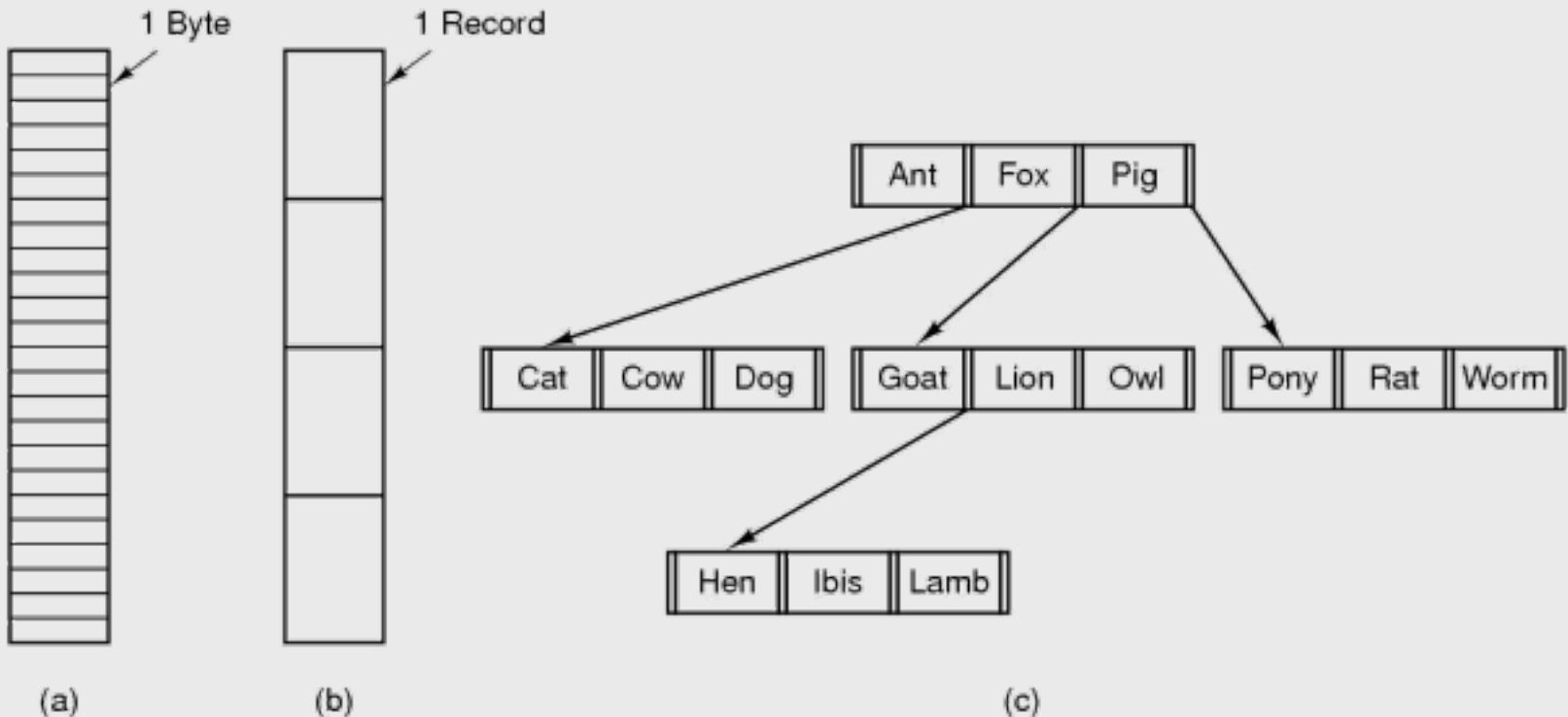


Archive



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



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` (“unmap”)
 - Gọi `map` để “chiếu” file vào không gian địa chỉ ảo của quá trình
 - Dùng các tác vụ truy xuất bộ nhớ (LOAD, STORE) 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 (1)

■ Create

- Tạo một file mới

■ Write

- Ghi dữ liệu vào file tại vị trí cho bởi con trỏ ghi

■ Read

- Đọc dữ liệu từ file tại vị trí cho bởi con trỏ đọc

■ Reposition (Seek)

- Thiết lập trị của con trỏ đọc/ghi
 - ▶ Hơi dư thừa vì trị của con trỏ đọc/ghi cũng là một tham số của Write và Read

Các tác vụ trên file (2)

■ 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 do hệ thống file (file system) cung cấp để người dùng dễ dàng tổ chức các file của mình.
 - Cung cấp ánh xạ từ **tên** của file đến địa chỉ khối mô tả file
 - ▶ UNIX: địa chỉ khối mô tả file là '**i-node number**' của file

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

Các tác vụ trên thư mục (tt)

■ Link

- Thêm một tên khác cho một file
 - ▶ UNIX: hard link, soft link
 - ▶ DOS / Windows: shortcut

■ Unlink

- Xóa một entry của thư mục

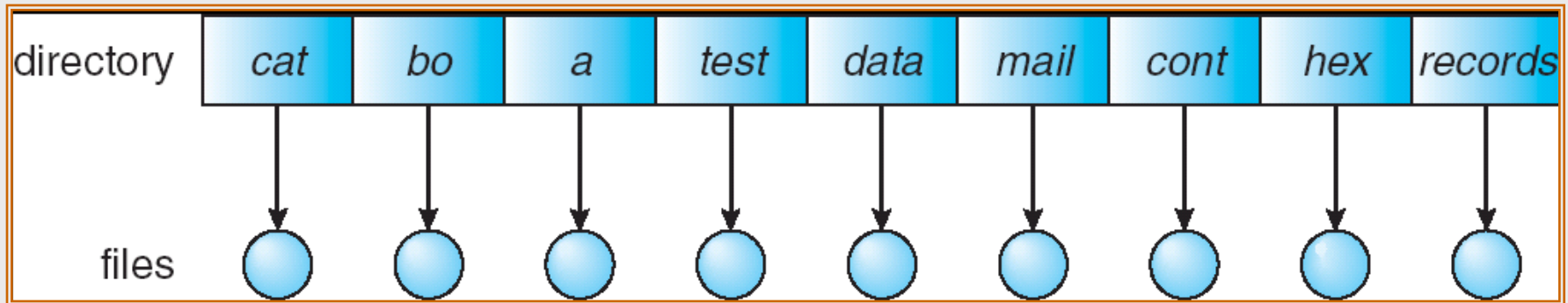
Các loại cấu trúc của thư mục

- 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 (1)

■ Single-level directory

- Chỉ có một directory cho mọi file trong hệ thống
 - ▶ CDC 6600, IBM MVS, PalmOS
- Gặp vấn đề đụng độ tên (name-collision problem)



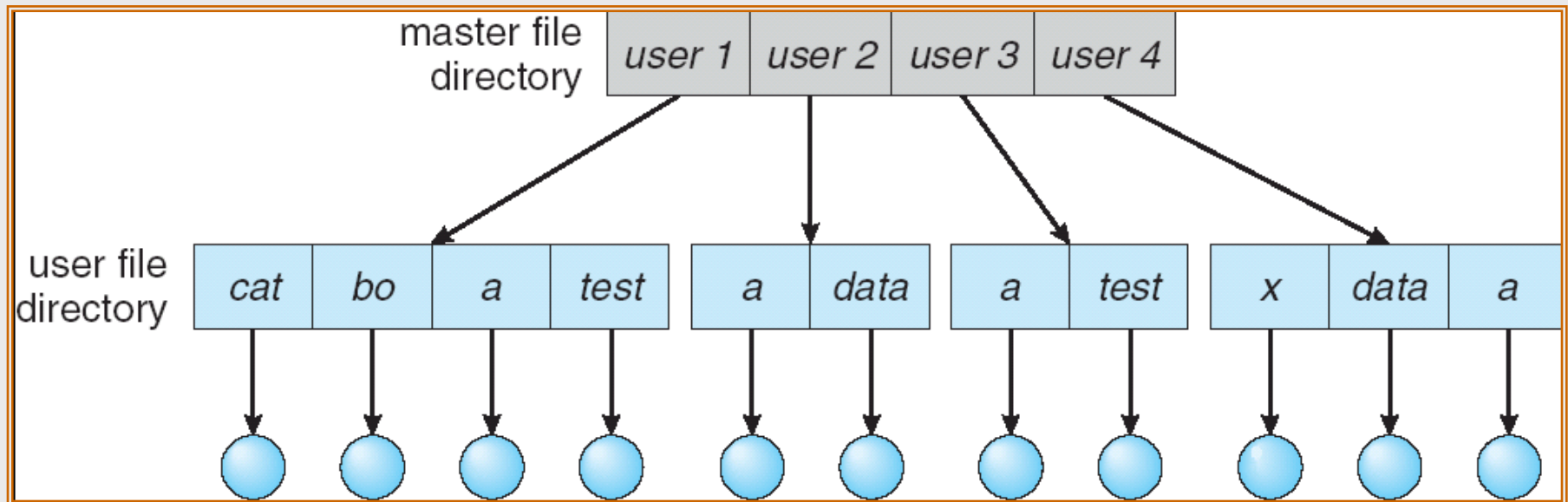
Các loại cấu trúc của thư mục (2)

- Cải tiến cấu trúc single-level directory
- Một số 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 MS Word,...

Các loại cấu trúc của thư mục (3)

■ Two-level directory

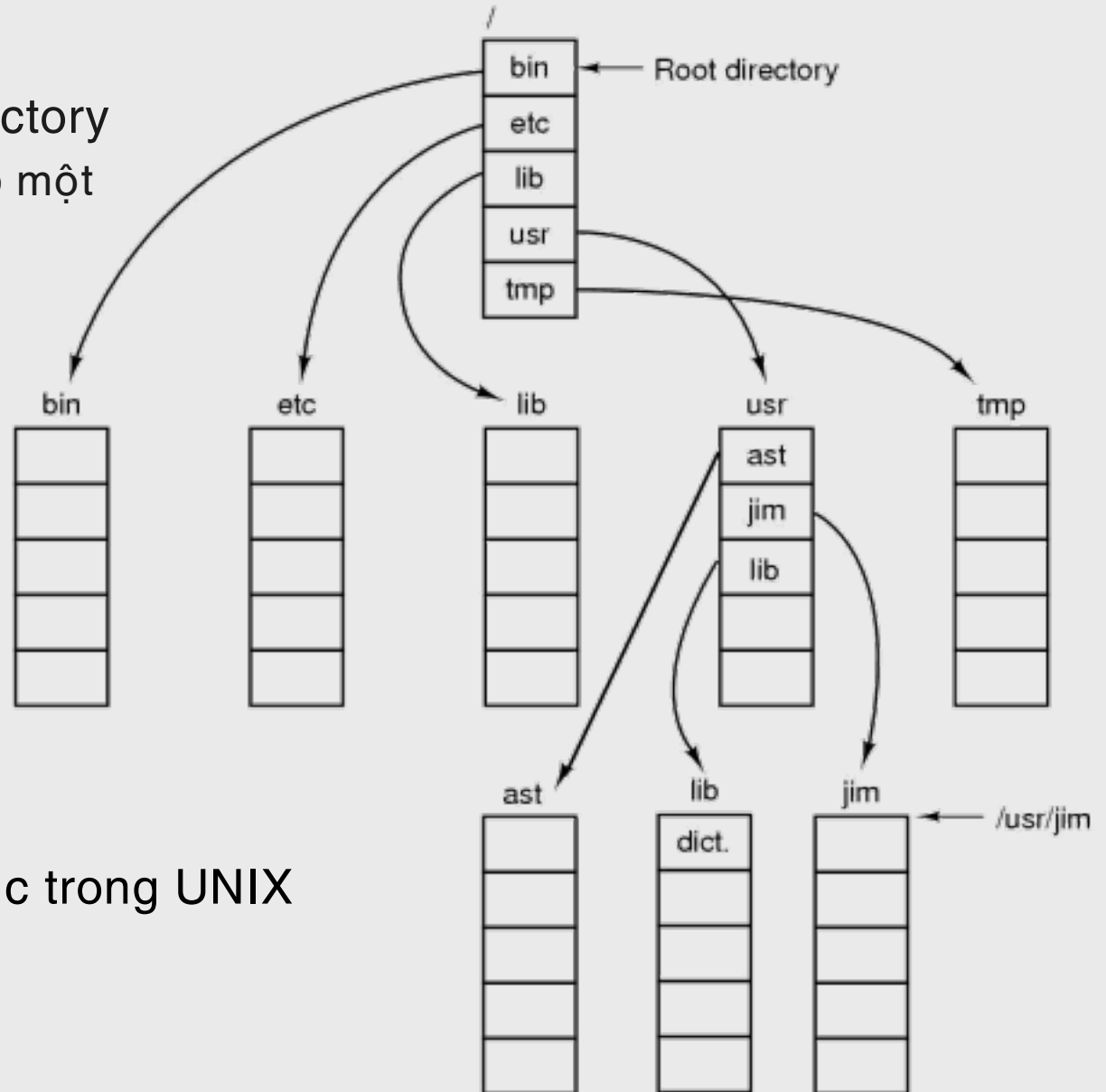
- 'master file' directory
- Các 'user file' directory: một thư mục riêng cho mỗi user
- Giải quyết được vấn đề đụng độ tên



Các loại cấu trúc của thư mục (4)

- Tree-structured directory

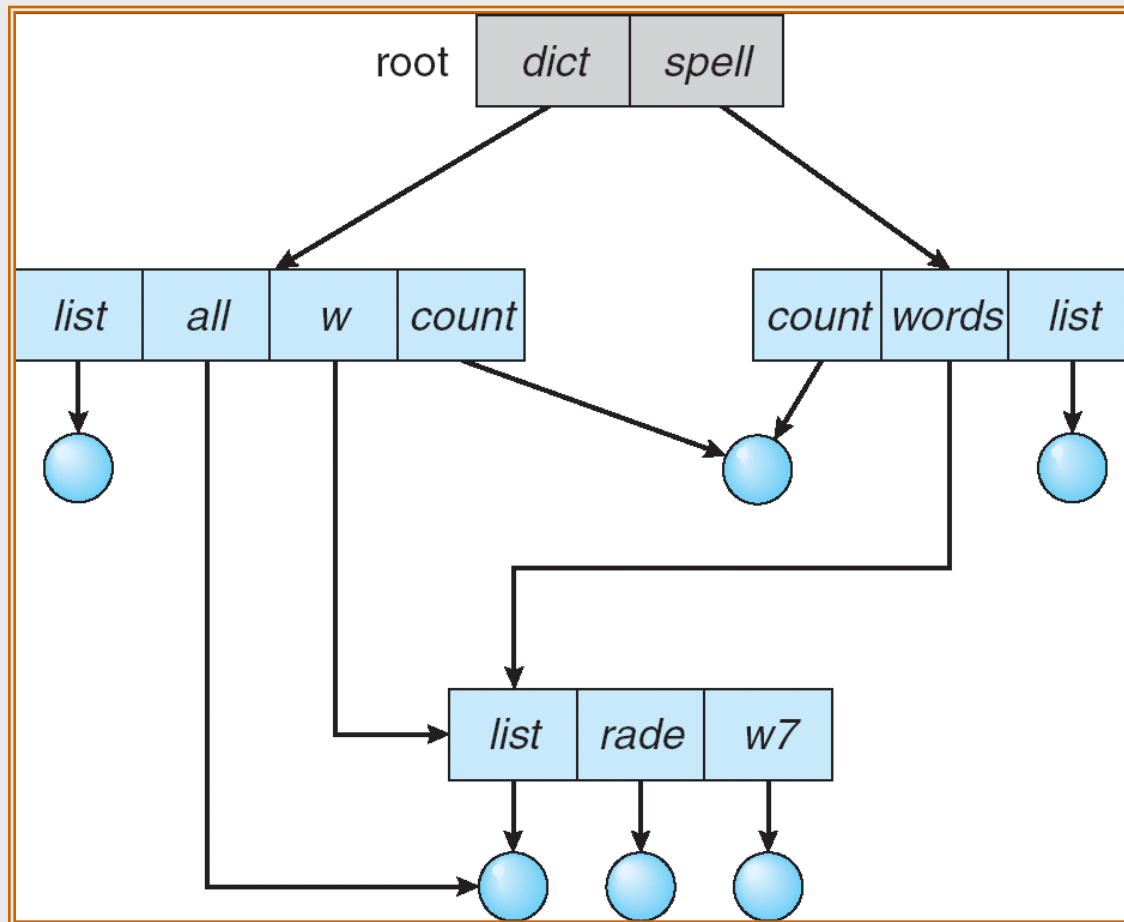
- Mỗi user được cấp một directory
 - ▶ Vd /usr/jim



Một cây thư mục trong UNIX

Các loại cấu trúc của thư mục (5)

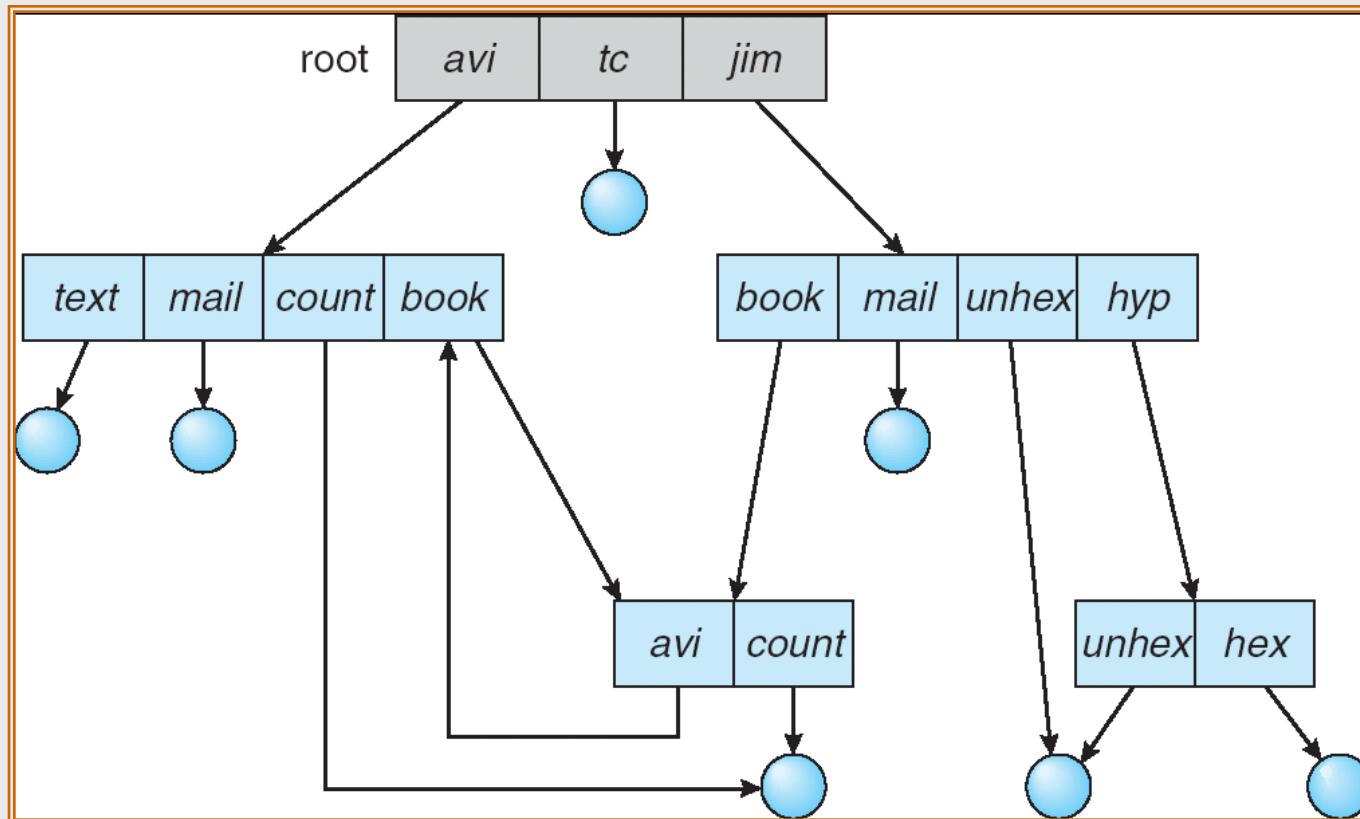
- Acyclic-graph directory



Các loại cấu trúc của thư mục (6)

■ 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**

Kết nối file system vào hệ thống

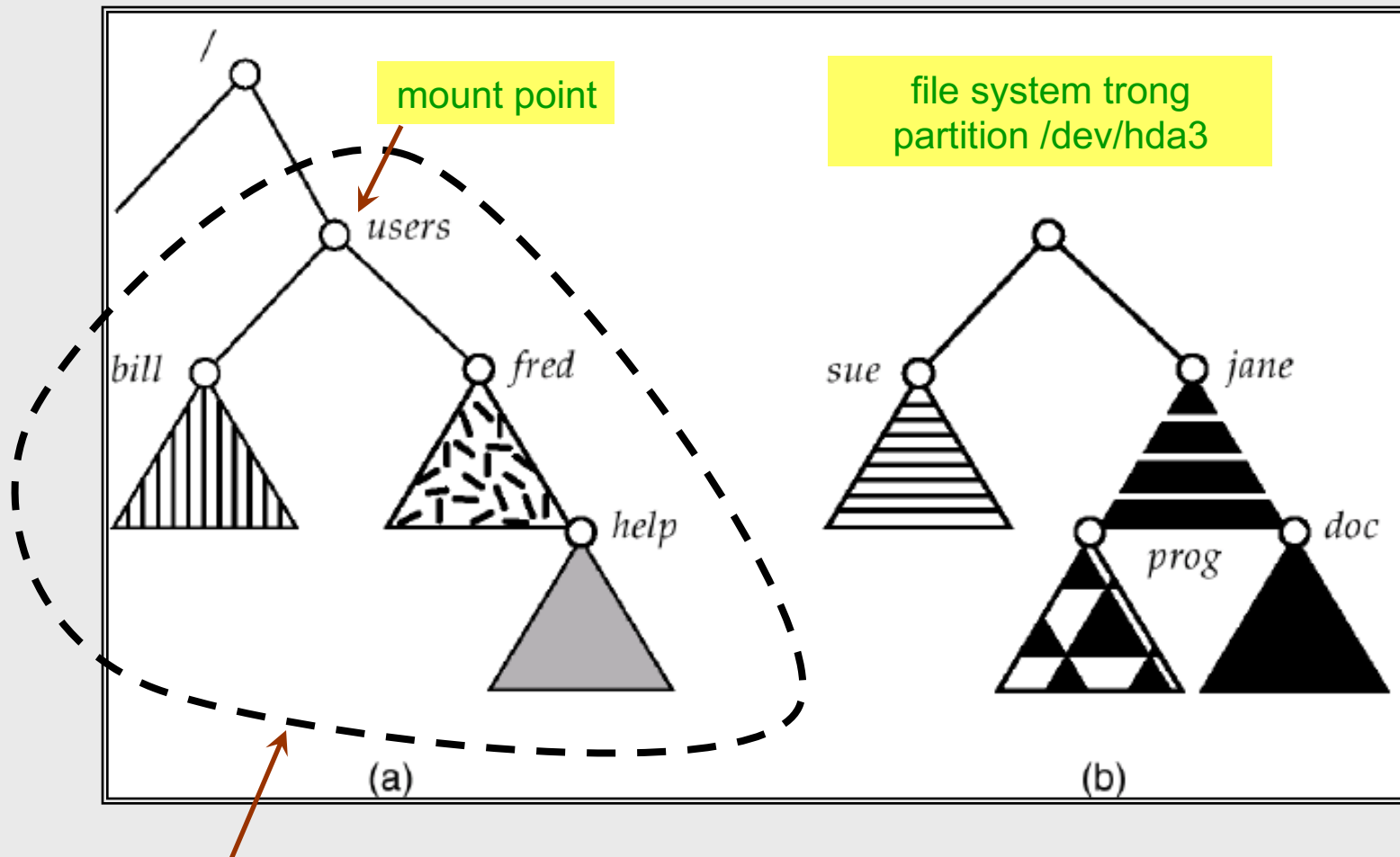
- Phải **gắn** (mount) thiết bị (partition,...) chứa file system để ứng dụng có thể truy cập file của nó
 - Thư mục để gắn vào đó một file system được gọi là **mount point**
- Để thực hiện lệnh gắn

```
# mount /device/thedisk /mountpoint
```

hệ điều hành phải thực thi các bước

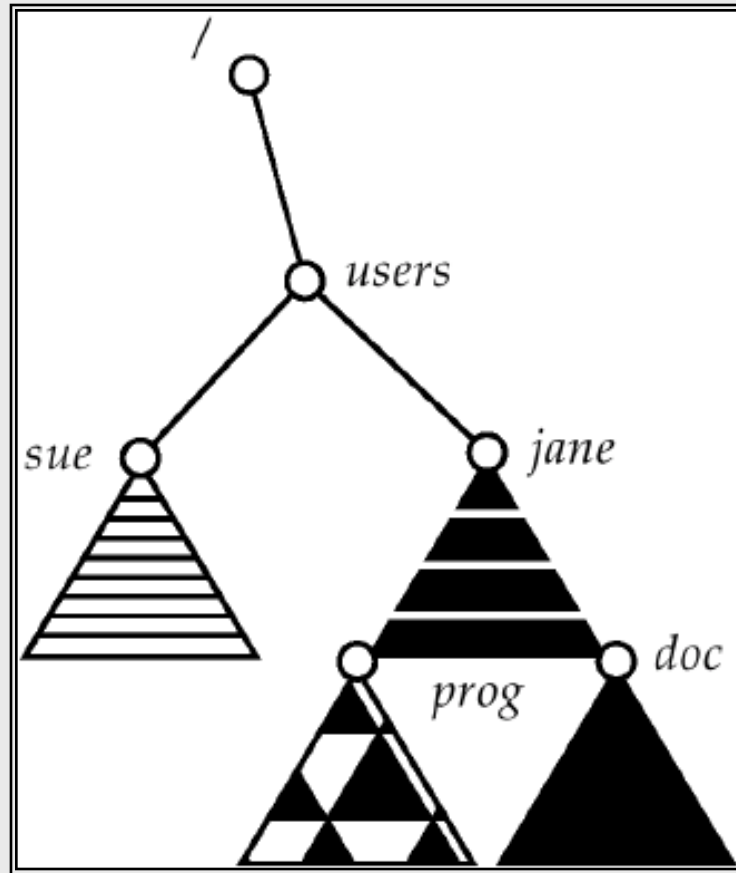
 - OS kiểm tra xem file system có 'hợp lệ' hay không
 - Gắn file system vào vị trí mountpoint
- Nếu mount point có chứa file?
 - Các file chứa trong thư mục mount point sẽ bị che cho đến khi unmount file system.

File system mounting (1)



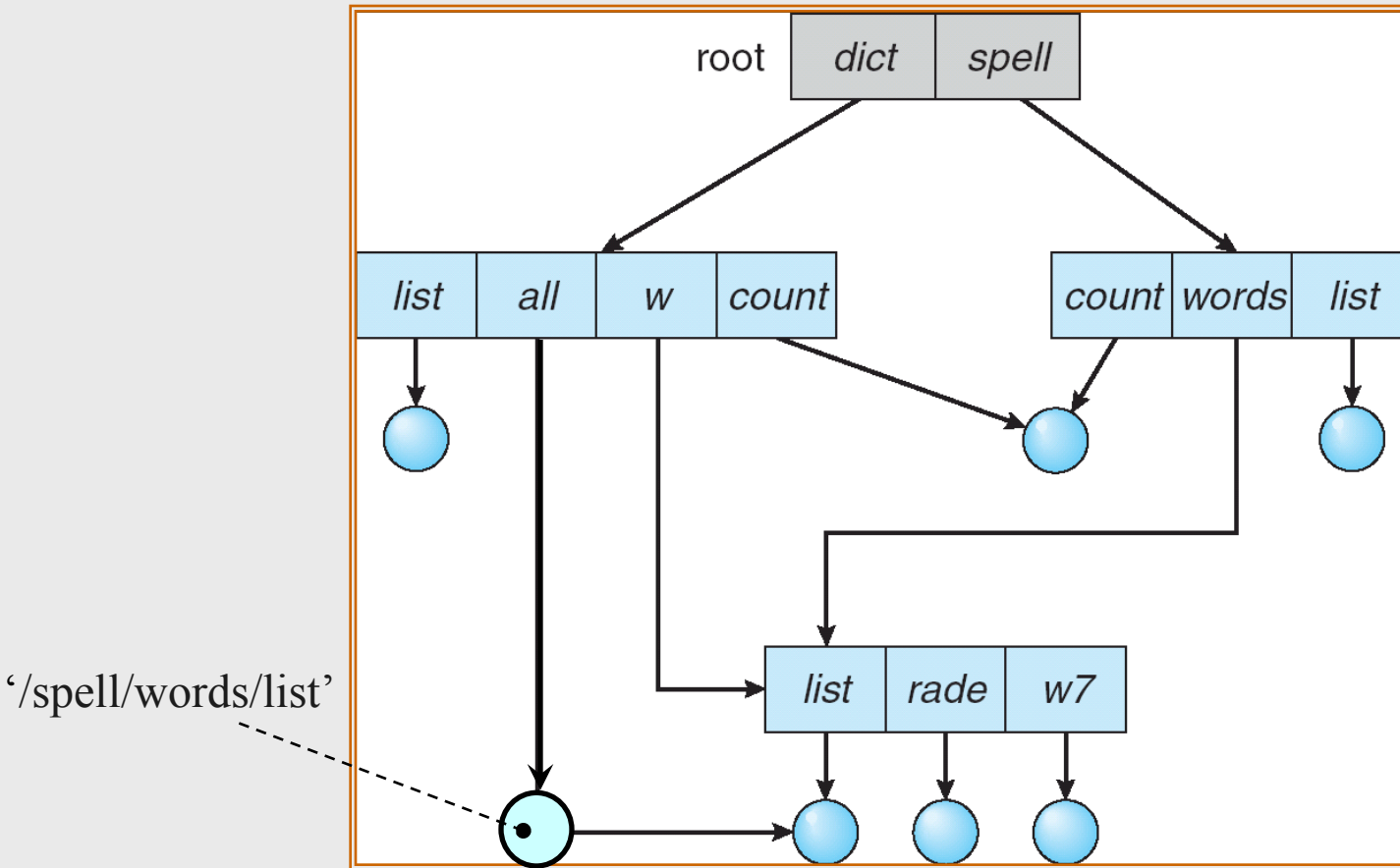
Các file bị che khi file system trong partition `/dev/hda3` được mount với lệnh `mount /dev/hda3 /users`

File system mounting (2)



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

Hard link và soft link

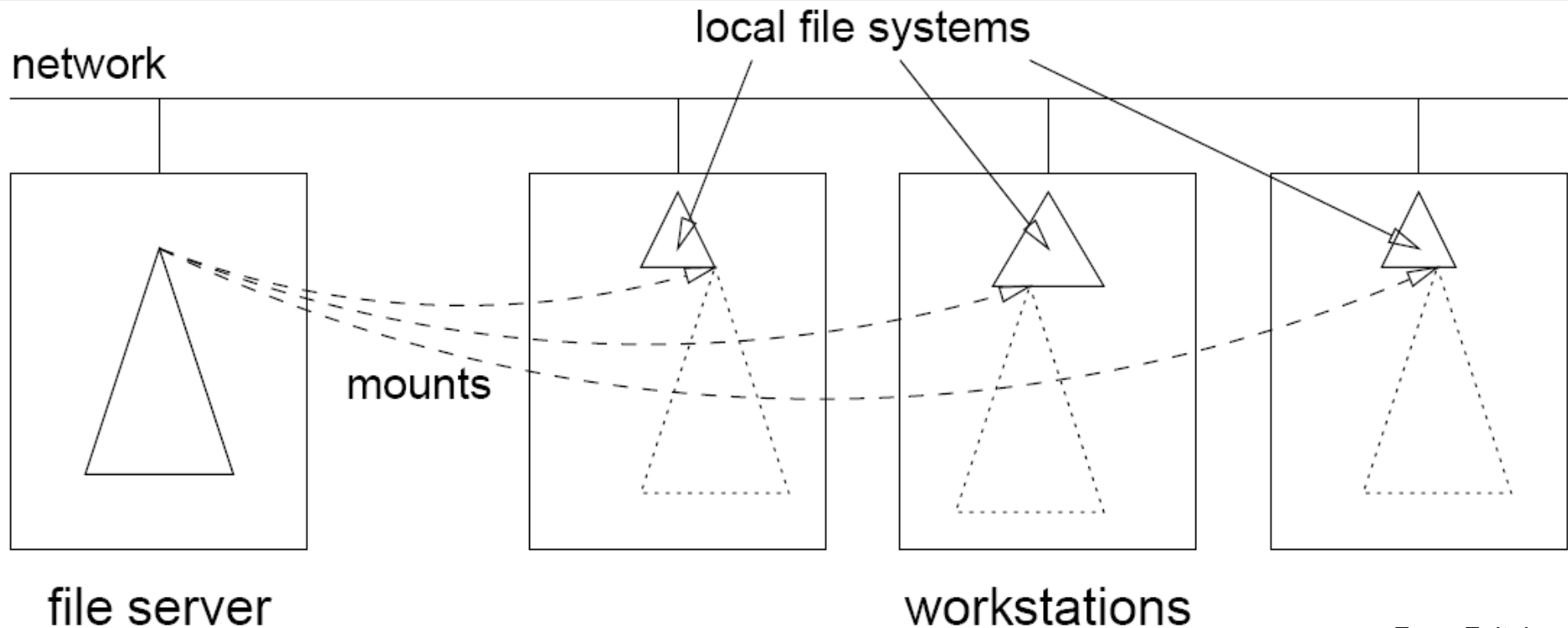


Trong UNIX

- Hard link: `ln /dict/count /spell/count`
- Soft link: `ln -s /spell/words/list /dict/all`

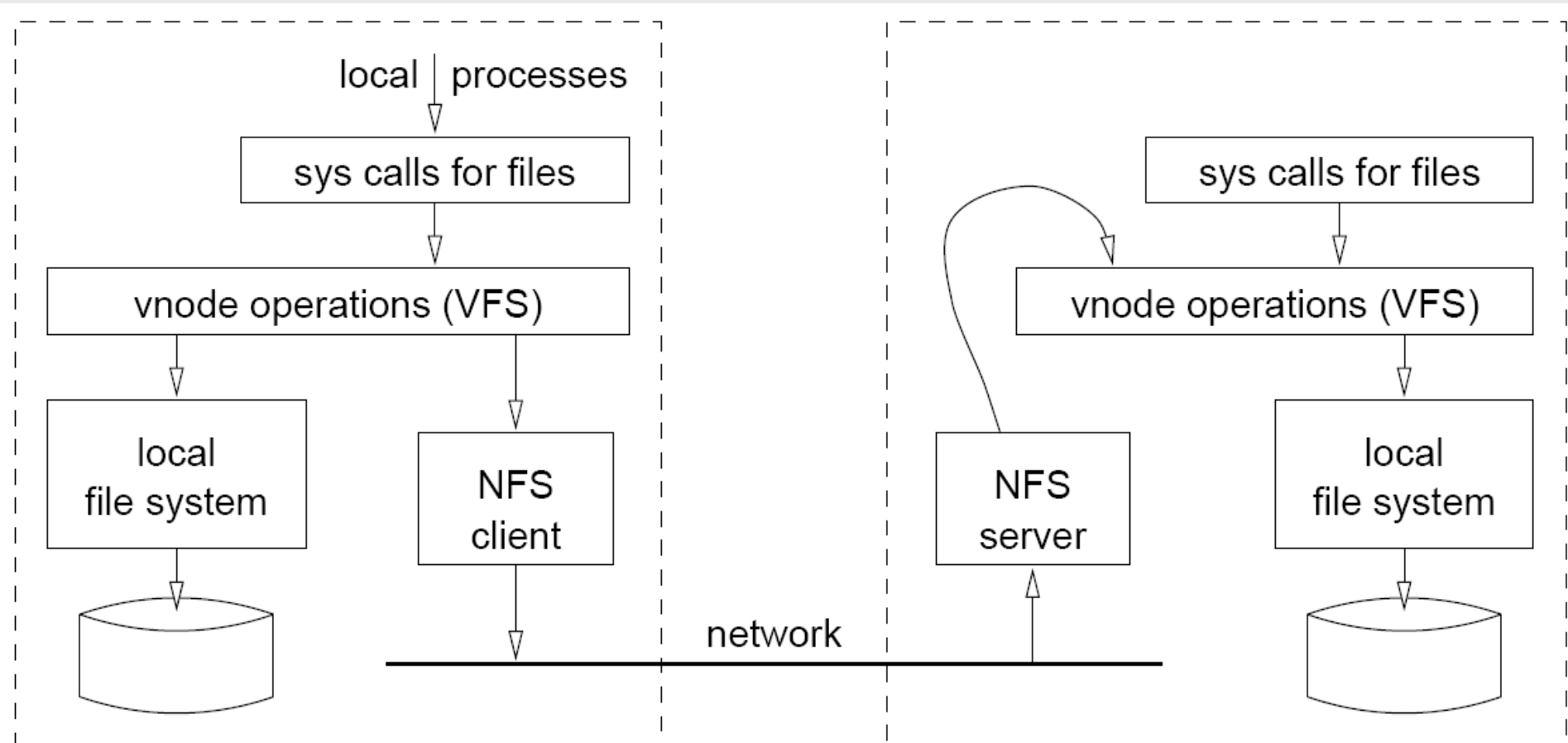
NFS

- **Network File System** (NFS) của Sun là một giải pháp cung cấp dịch vụ file trong mạng máy tính
 - Mô hình client-server, hiện thực với RPC
 - NFS client phải **gắn** (mount) file system ở xa trước khi sử dụng



VFS phân biệt truy xuất cục bộ hay ở xa

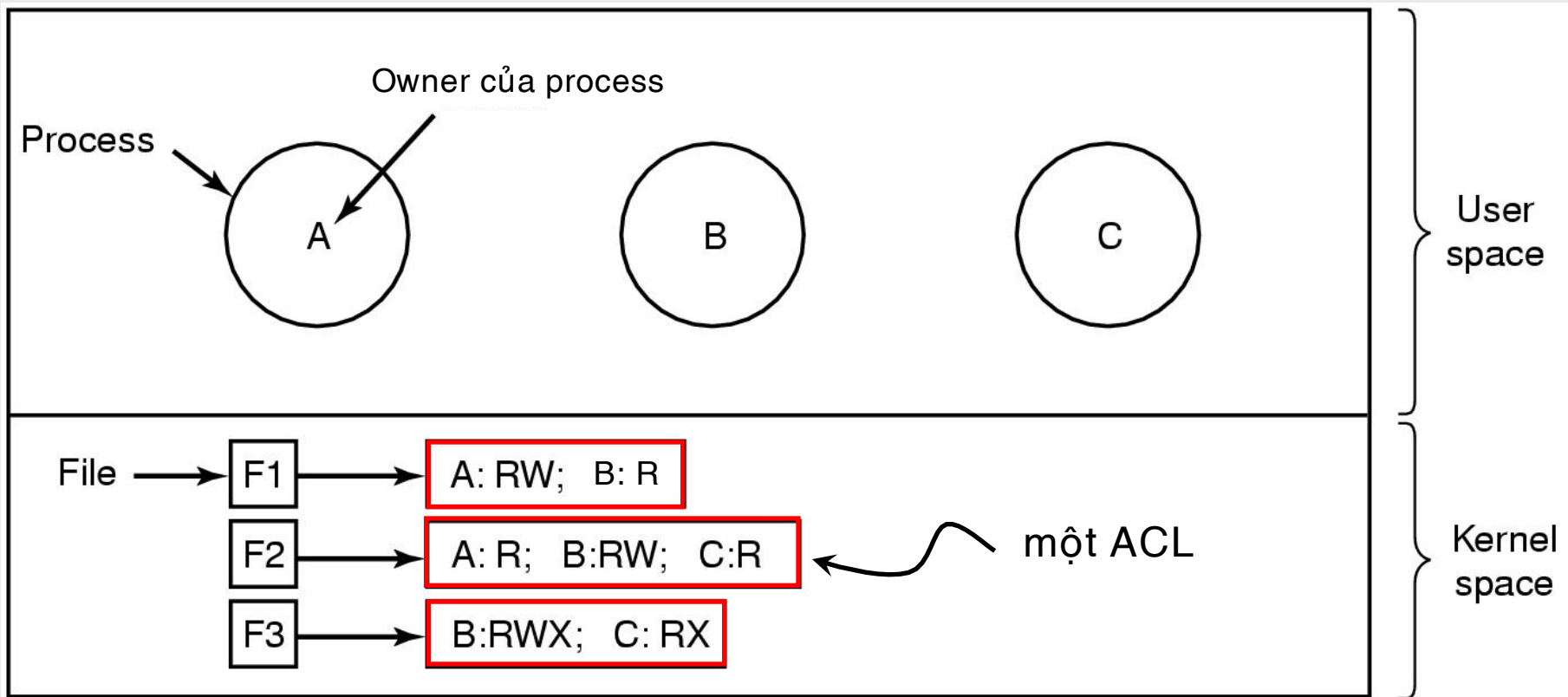
- Nếu file ở xa, VFS kích hoạt NFS client để nó xử lý tiếp



Bảo vệ (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 ⟨user, rights⟩
 - ▶ 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