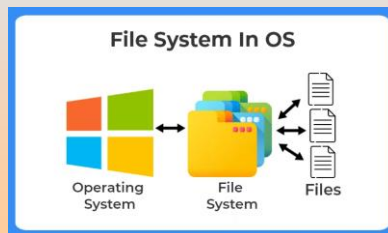


Chương 5

Giao diện Hệ thống File (File System Interface)



Mục tiêu

- Giải thích chức năng của hệ thống file
- Miêu tả giao diện của hệ thống file
- Thảo luận việc cân bằng các yếu tố thiết kế hệ thống file, bao gồm các phương pháp truy nhập, chia sẻ file, khóa file, các cấu trúc thư mục
- Giải thích sự bảo vệ hệ thống file

Nội dung:

- Khái niệm File
- Các phương pháp truy nhập - Access Methods
- Cấu trúc thư mục - Directory Structure
- Chia sẻ file - File Sharing
- Protection

5.1. Khái niệm File

- File là một tập hợp của các thông tin liên quan, được ghi trên bộ nhớ thứ cấp (là bộ nhớ ổn định) và được đặt tên.
- Từ góc nhìn của người sử dụng, file là đơn vị bộ nhớ logic nhỏ nhất. Các file được ánh xạ bởi HĐH vào các thiết bị nhớ vật lý.
- Kiểu tệp:
 - Data
 - số - numeric
 - ký tự - character
 - nhị phân - binary
 - Program
- Nói chung, file là một chuỗi các bit, byte, dòng hoặc bản ghi

Cấu trúc File

- Không cấu trúc - chuỗi các words, bytes
- Cấu trúc bản ghi đơn giản
 - các dòng (lines)
 - độ dài cố định
 - độ dài thay đổi
- Các cấu trúc phức tạp
 - văn bản có định dạng - Formatted document
 - file nạp có thể tái định vị - Relocatable load file
- Ai quyết định cấu trúc file?
 - HĐH
 - Chương trình

Thuộc tính File

- **Name** – chỉ là thông tin ở dạng người đọc được.
- **Type** – cần thiết cho các HĐH hỗ trợ nhiều kiểu file.
- **Location** – con trỏ tới vị trí file trên thiết bị.
- **Size** – kích thước hiện tại của file.
- **Protection** – kiểm soát ai có thể đọc, ghi, thực hiện file.
- **Time, date, user identification** – dữ liệu dùng cho protection, security, và theo dõi sử dụng.
- Thông tin về file được lưu trong cấu trúc thư mục, cũng được lưu trên đĩa.

Thao tác với File

- Tạo file
- Ghi file
- Đọc file
- Định vị trong file – file seek
- Xóa file
- Cắt bớt file (truncate)
- $\text{Open}(F_i)$ – tìm chỉ mục F_i trong cấu trúc thư mục trên đĩa rồi chuyển nội dung của chỉ mục vào bộ nhớ.
- $\text{Close}(F_i)$ – chuyển nội dung của chỉ mục F_i trong bộ nhớ ra cấu trúc thư mục trên đĩa.

File Types – Name, Extension

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

5.2. Các phương pháp truy nhập

■ Truy nhập tuần tự - Sequential Access

- Truy nhập tuần tự qua các bản ghi từ đầu tệp đến cuối tệp

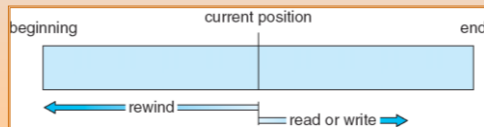
read next

write next

reset

no read after last write

- Một số HĐH cho phép nhảy tới hoặc lui n bản ghi.
- Các trình soạn thảo và trình biên dịch thường truy nhập tệp theo phương pháp này.



Các phương pháp truy nhập (tiếp)

■ Truy nhập trực tiếp - Direct Access

- Tệp được tạo bởi các bản ghi có kích thước cố định
- Có thể truy nhập các bản ghi tại vị trí bất kỳ trong tệp mà không cần theo thứ tự.
- Các CSDL thường được tổ chức theo phương pháp này
- Sử dụng các phương thức:

read n

write n

position to n

read next

write next

rewrite n

n = số hiệu bản ghi cần truy nhập, có thể bắt đầu từ 0 hoặc 1 tùy thuộc HĐH

Minh họa cách truy nhập tuần tự trong một file truy nhập trực tiếp

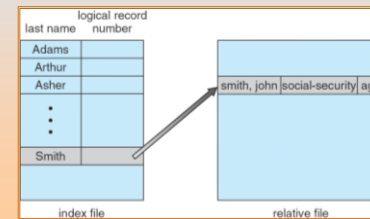
cp - current position: biến xác định vị trí hiện tại

sequential access	implementation for direct access
<i>reset</i>	<i>cp = 0;</i>
<i>read next</i>	<i>read cp;</i> <i>cp = cp + 1;</i>
<i>write next</i>	<i>write cp;</i> <i>cp = cp + 1;</i>

Các phương pháp truy nhập (tiếp)

■ Truy nhập index-relative

- Tệp index chứa các con trỏ tới các bản ghi trong tệp relative
- Để truy nhập các bản ghi trong tệp relative, trước tiên tìm index, tiếp theo dùng con trỏ để truy nhập trực tiếp tệp relative để tìm bản ghi.
- Hữu dụng khi tìm kiếm trong các tệp lớn vì số lần thực hiện vào-ra ít
- Có thể có nhiều hơn một mức index: index-index-relative

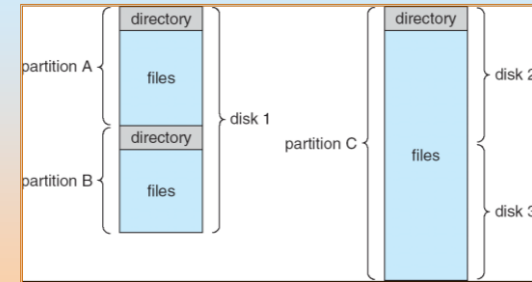


5.3. Cấu trúc thư mục

Để quản lý số lượng lớn các tệp trên đĩa, tổ chức chúng trong 2 bước:

- Đầu tiên, chia đĩa thành một hay nhiều *partition* (minidisk-IBM, volume-PC & Macintosh)
 - partition - cấu trúc mức thấp, để chứa các tệp và thư mục
 - một số HDH cho phép partition lớn hơn đĩa
- Tiếp theo, mỗi partition có một **device directory** (directory) ghi thông tin về tất cả các tệp trên partition đó: tên tệp, vị trí, kích thước, kiểu tệp...

Một tổ chức hệ thống file cơ bản



Các thao tác trên một thư mục

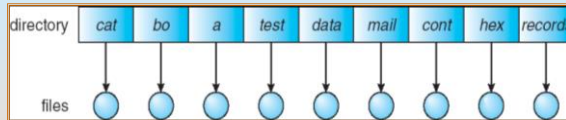
- Tìm kiếm 1 tệp
- Tạo 1 tệp
- Xóa 1 tệp
- Liệt kê danh sách tệp trong thư mục
- Đổi tên 1 tệp
- Truy nhập toàn bộ hệ thống file

Tổ chức logic thư mục để sử dụng

- **Hiệu quả (Efficiency)** – định vị file nhanh chóng.
- **Đặt tên (Naming)** – thuận tiện cho người sử dụng.
 - nhiều tệp có thể có cùng tên.
 - 1 tệp có thể có nhiều tên.
- **Gom nhóm (Grouping)** – nhóm logic các tệp theo thuộc tính, (vd: all Java programs, all games, ...)

5.3.1. Single-Level Directory

- Một thư mục chứa tất cả các tệp.



- Ưu:

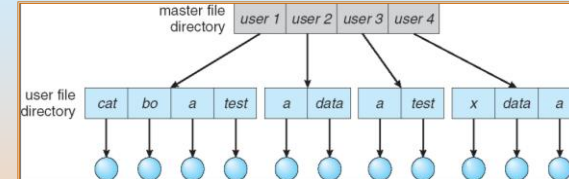
- dễ hiểu, dễ quản lý
- kích thước nhỏ

- Nhược:

- vấn đề đặt tên: mỗi tệp phải có tên duy nhất
- vấn đề gom nhóm: không thể

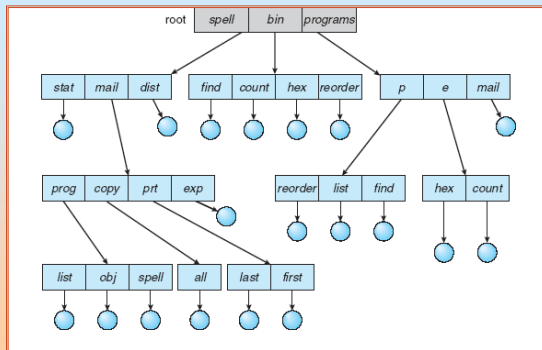
5.3.2. Two-Level Directory

- Mỗi user có một thư mục riêng



- Để truy nhập 1 tệp ở thư mục khác, cần có đường dẫn đầy đủ
- Có thể có các tệp trùng tên cho các user khác nhau
- Tìm kiếm hiệu quả hơn
- Không có khả năng gom nhóm

5.3.3. Tree-Structured Directories

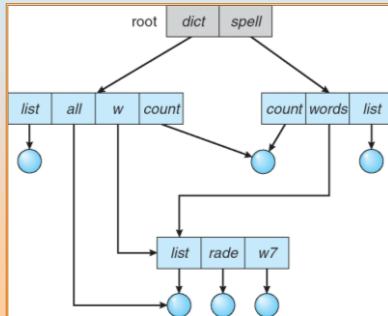


Tree-Structured Directories (tiếp)

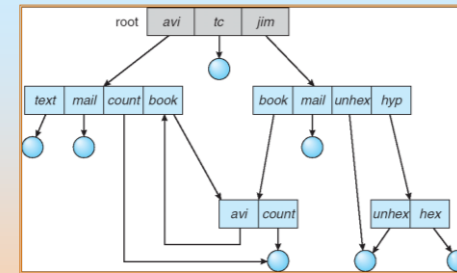
- Có 1 thư mục gốc (root)
- Mỗi tệp có 1 đường dẫn duy nhất:
 - tuyệt đối, vd: `C:\Windows\php.ini`
 - quan hệ (với thư mục hiện tại), vd: `.\System32\test.dll`
- Mỗi thư mục chứa các tệp và/hoặc các thư mục con
- Tìm kiếm hiệu quả
- Thuận tiện trong đặt tên
- Có khả năng gom nhóm

5.3.4. Acyclic-Graph Directories

- Có sự chia sẻ (not copy!) các thư mục con và các tệp, thuận tiện khi nhiều user làm việc trong 1 dự án.



5.3.5. General Graph Directory



- Khi một liên kết được thêm vào cấu trúc, cần đảm bảo không tạo thành chu trình → sử dụng giải thuật tìm kiếm chu trình trong đồ thị, nhưng là việc "nặng nhọc" vì đồ thị trên đĩa, không phải trong bộ nhớ trong.

5.4. File Sharing

- Yêu cầu: phải chia sẻ các file trên các hệ thống đa người dùng (multi-user systems).
- Chia sẻ file có thể được thực hiện thông qua một lược đồ *protection*.
- Trên các hệ thống phân tán (distributed systems), các file có thể được chia sẻ qua mạng (network).
- Network File System (NFS) là một phương thức chia sẻ file phân tán (distributed file-sharing method) phổ biến.
 - sử dụng trong mô hình client-server
 - các user ID phải phù hợp cả với client và server để xác nhận quyền truy nhập file trên server.

5.5. Protection

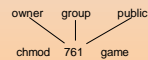
- Người tạo/sở hữu file cần có khả năng giám sát:
 - thao tác nào đã được thực hiện
 - bởi user nào?
- Các loại truy nhập
 - Read
 - Write
 - Execute
 - Append
 - Delete
 - List

Access Lists and Groups

- Chế độ truy nhập: read, write, execute
- Ba lớp người sử dụng:

a) owner access	7	⇒	RWX 1 1 1
b) group access	6	⇒	RWX 1 1 0
c) public access	1	⇒	RWX 0 0 1

- Yêu cầu người quản lý tạo một group G (có tên duy nhất), rồi thêm các user vào group.
- Đối với các file (vd *game*) hoặc subdirectory, xác định sự truy nhập tương tự như trên.



Gán group cho file:

chgrp G game



BÀI TẬP

1. Tập tin là gì ? Thư mục là gì ? Tại sao phải quản lý tập tin và thư mục ?
2. Tập tin có những đặc tính gì ? Những đặc tính nào là quan trọng ? Tại sao ?
3. Nêu các chức năng của tập tin và thư mục.
4. So sánh các phương pháp cài đặt bảng phân phối vùng nhớ.
5. Tập tin chia sẻ là gì ?
6. Vì sao phải lưu ý đến độ an toàn của hệ thống tập tin ?