

Chương 2: Hệ Thống Files

Hệ điều hành

ThS. Đinh Xuân Trường
truongdx@ptit.edu.vn



Posts and Telecommunications
Institute of Technology
Faculty of Information Technology 1



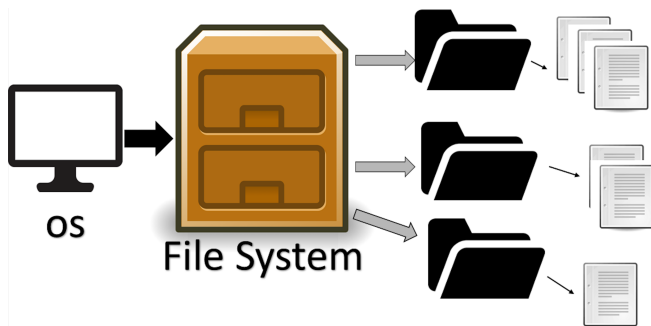
CNTT1
Học viện Công nghệ Bưu chính Viễn thông

August 15, 2022

Tổ chức bên trong của thư mục

1. Các khái niệm
2. Các phương pháp truy cập file
3. Các thao tác với file
4. Thư mục
5. Cấp phát không gian cho file
6. Quản lý không gian trống trên đĩa
7. Độ tin cậy của hệ thống file
8. Bảo mật cho hệ thống file
9. Hệ thống file FAT

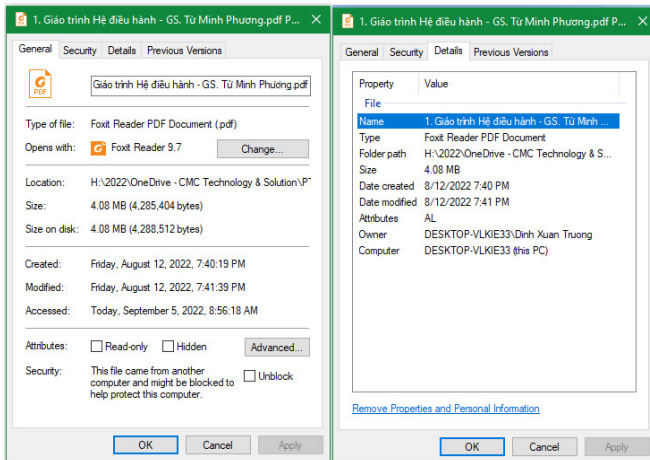
- *File hay tập tin được định nghĩa như tập hợp các thông tin liên quan đến nhau được đặt tên và được lưu trữ trên bộ nhớ ngoài*



Hình 2: Hệ thống files

- *Thuộc tính của file: để quản lý file ngoài nội dung, HĐH còn định nghĩa các thuộc tính, tính chất.*

- ▶ File có các thuộc tính như sau:
 - Tên file
 - Kiểu file
 - Kích thước file
 - Người tạo file, người sở hữu
 - Quyền truy cập file
 - Thời gian tạo file, sửa file, truy cập lần cuối
 - Vị trí file



Hình 3: Thuộc tính của files

- ▶ Đặt tên cho file:
 - Cho phép xác định file
 - Là thông tin người dùng thường sử dụng nhất khi làm việc với file
 - Quy tắc đặt tên cho file của một số HDH:

Hệ điều hành	Độ dài tối đa	Phân biệt chữ hoa, chữ thường	Cho phép sử dụng dấu cách	Các ký tự cấm
MS-DOS	8 cho tên file 3 cho mở rộng	không	không	Bắt đầu bằng chữ cái hoặc số Không được chứa các ký tự / \ [] : ; = , ^ ? @
Windows NT FAT	255 ký tự cho cả tên file và đường dẫn	không	có	Bắt đầu bằng chữ cái hoặc số Không được chứa các ký tự / \ [] : ; = , ^ ? @
Windows NT NTFS	255	không	có	Không được chứa các ký tự / \ < > * :
Linux (EXT3)	256	Có	có (nếu tên file chứa trong ngoặc kép)	Không được chứa các ký tự ! @ # \$ % ^ & * () [] { } ' " " \ ; < > `

Hình 4: Quy tắc đặt tên files

- ▶ Các thông tin trong file có thể rất khác nhau
- ▶ Có những file chứa nhiều thông tin không có cấu trúc: file văn bản. File có cấu trúc như: file CSDL, file excel.
Cấu trúc của file cũng rất khác nhau và phụ thuộc vào thông tin chứa trong file
- ▶ HDH có cần biết và hỗ trợ các kiểu cấu trúc file?
- ▶ Hỗ trợ cấu trúc file ở mức HDH:
 - Ưu điểm:
 - ▶ Các thao tác với file sẽ dễ dàng hơn đối với người lập trình ứng dụng
 - ▶ HDH có thể kiểm soát được các thao tác với file
 - Nhược điểm:
 - ▶ Tăng kích thước hệ thống
 - ▶ Tính mềm dẻo của HDH bị giảm
- ▶ Thực tế các HDH chỉ coi file là tập hợp các byte không cấu trúc

- ▶ Đa số HDH không hỗ trợ và quản lý kiểu cấu trúc file
- ▶ Cấu trúc file do chương trình ứng dụng và người dùng tự quản lý
- ▶ Trong HDH UNIX, DOS, WINDOWS, file được xem như tập hợp các byte.
- ▶ Các chương trình ứng dụng khác nhau sẽ tự tạo ra và quản lý cấu trúc file riêng mình.
Ví dụ: chương trình đồ họa lưu file dưới dạng mã nhị phân đã được giải nén, chương trình hệ thống quản lý dữ liệu sẽ tạo ra file bao gồm các bản ghi.

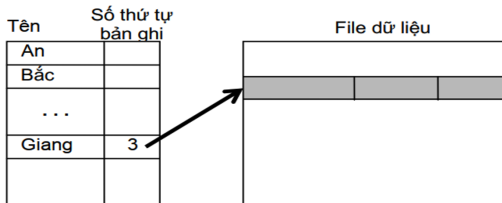
- ▶ Để đọc/ghi file hệ điều hành phải quy định cách thức truy cập tới nội dung file. Mỗi HDH có thể hỗ trợ một hoặc nhiều cách truy cập khác nhau.
- ▶ **Truy cập tuần tự:**
 - Thông tin được đọc, ghi theo từng byte/ bản ghi lần lượt từ đầu file
 - Sử dụng 1 con trỏ để định vị vị trí hiện thời trong file
 - Sau mỗi thao tác đọc hoặc ghi con trỏ sẽ được tăng lên 1 để trỏ tới vị trí tiếp theo
 - Kiểu truy cập này rất phù hợp với các file được lưu trữ trên băng từ. Thông tin được ghi hoặc đọc lần lượt theo chiều quay của băng.

► Truy cập trực tiếp:

- File được xem như các khối/ bản ghi được đánh số
- Các khối có thể truy cập theo thứ tự bất kỳ
- Chẳng hạn ta có thể đọc khối 50 sau đó đọc khối 13 rồi khối 101.
- Việc truy cập trực tiếp dựa trên đặc điểm của đĩa cho phép truy cập các khối bất kỳ
- File được chứa trong các khối khác nhau của đĩa do vậy cũng cho phép truy cập không tuần theo thứ tự

► Truy cập dựa trên chỉ số:

- Cho phép truy cập tới bản ghi trong file, không theo số thứ tự hoặc vị trí của bản ghi trong file mà theo khóa ứng với bản ghi đó.
- File chứa 1 chỉ số riêng: gồm các khóa và con trỏ chỉ tới các bản ghi trong file
- Truy cập: tìm khóa tương ứng trong chỉ mục, sau đó theo con trỏ xác định bản ghi và truy cập trực tiếp tới nó



Hình 5: Truy cập dựa trên chỉ số

- ▶ Hệ điều hành quy định các thao tác mà người dùng và ứng dụng có thể thực hiện theo file
- ▶ Các thao tác được hệ điều hành thực hiện khi chương trình ứng dụng gọi lời gọi hệ thống tương ứng
- ▶ Những thao tác với file thường gặp trong hệ điều hành bao gồm:
 - **Tạo file:** Tạo file trống chưa có dữ liệu; File được dành 1 chỗ trong thư mục kèm theo một số thuộc tính như thời gian tạo file, tên file, người tạo file
 - **Xoá file:**
 - ▶ Giải phóng không gian mà dữ liệu của file chiếm trên đĩa
 - ▶ Giải phóng chỗ của file trong thư mục
 - ▶ Việc giải phóng không gian có thể đơn thuần là đánh dấu không gian như không gian tự do.
 - **Mở file:**
 - ▶ Thực hiện trước khi ghi và đọc file

- ▶ Đọc các thuộc tính của file trên đĩa vào bộ nhớ để tăng tốc độ cho thao tác đọc ghi tiếp theo.
- **Đóng file:**
 - ▶ Xóa các thông tin về file ra khỏi bảng trong bộ nhớ để nhường chỗ cho các file sắp mở.
 - ▶ Rất nhiều hệ điều hành hạn chế số lượng file được mở cùng một lúc nên việc đóng các file đã truy cập xong là rất quan trọng.
- **Ghi vào file:**
 - ▶ Dữ liệu được ghi vào vị trí hiện thời của file
 - ▶ Nếu vị trí hiện thời là cuối file, thông tin được thêm vào và kích thước file tăng lên
 - ▶ Nếu vị trí file không phải cuối file thông tin sẽ bị ghi đè lên
- **Đọc file** Thông tin ở vị trí hiện thời sẽ được đọc
- **Định vị (seek)** Cho phép xác định vị trí hiện thời của file

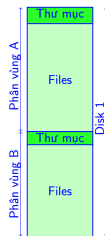
- **Đọc thuộc tính của file:** Một số chương trình trong quá trình làm việc cần đọc các thuộc tính của file như quyền truy cập cho hệ thống bảo mật.
- **Thay đổi thuộc tính của file:** Đặt lại giá trị cho một số thuộc tính của file
- **Cắt bỏ nội dung file :** Thao tác xoá bỏ nội dung file và giải phóng vùng không gian cho file chiếm nhưng vẫn giữ lại các thuộc tính của file trừ độ dài và vị
- **Khoá file:** Khi đồng thời nhiều tiến trình thay đổi một nội dung file, có thể dẫn đến kết quả không mong đợi nên một số HĐH cho phép khoá file. Khi một tiến trình mở file, thì tiến trình đó có thể yêu cầu khoá file và không cho phép các tiến trình khác truy cập hoặc có thể đọc file nhưng không thể ghi vào file đó.

- ▶ Số lượng file lưu trữ trên đĩa rất lớn => phải tổ chức để dễ dàng quản lý, truy cập files
- ▶ Không gian trên đĩa được chia thành các phần (partition/ volume) gọi là đĩa logic

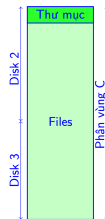
* Đĩa được chia thành nhiều phân vùng:

Partitions, Minidisks, Volumes.

* Mỗi phân vùng được xử lý như vùng lưu trữ phân biệt. Có thể chứa một HĐH riêng

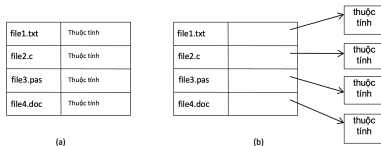


- ▶ Kết hợp một vài đĩa thành một cấu trúc logic lớn
 - * Người dùng chỉ quan tâm tới cấu trúc file và thư mục logic
 - * Không quan tâm tới cách phân phối vật lý không gian đĩa cho files



- ▶ Để quản lý file trên các đĩa logic, thông tin về file được lưu trong thư mục của đĩa
- ▶ **Thư mục** tạo thành từ các *khoản mục* (entry) và mỗi khoản mục tương ứng với một file
- ▶ *Khoản mục* chứa thông tin của file: tên, kích thước, vị trí, kiểu file, ... hoặc con trỏ tới nơi lưu trữ thông tin này.

- ▶ Thư mục như 1 bảng, mỗi dòng là khoản mục ứng với 1 file, việc tìm ra dòng cần thiết thực hiện theo tên file. Thư mục cho phép ánh xạ tên file vào file đó.
- ▶ Các cách lưu thông tin về file trong thư mục:
 - Toàn bộ thuộc tính của file được lưu trong thư mục, file chỉ chứa data => kích thước khoản mục, thư mục lớn
 - Một phần thuộc tính được lưu trữ luôn cùng với dữ liệu của file. Thư mục chỉ lưu thông tin tối thiểu cần thiết cho việc tìm kiếm vị trí file trên đĩa => kích thước giảm

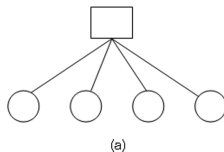


Hình 6: Lưu trữ thuộc tính file a) Cùng thư mục b) Cùng với file

- ▶ Mở file:
 - HDH tìm trong thư mục khoản mục ứng với tên file cần mở
 - Đọc các thuộc tính và vị trí dữ liệu của file vào bảng chứa thông tin về các file đang mở
 - Nếu khoản mục trở tới cấu trúc dữ liệu khác chứa thuộc tính file, cấu trúc này sẽ được đọc vào bảng
- ▶ Tìm kiếm file: cấu trúc thư mục cho phép tìm kiếm theo tên file
- ▶ Tạo file: tạo khoản mục mới và thêm vào thư mục
- ▶ Xoá file: thông tin về file và khoản mục bị xóa khỏi thư mục
- ▶ Duyệt thư mục: liệt kê các file trong thư mục và thông tin chứa trong khoản mục của file
- ▶ Đổi tên: chỉ thực hiện với thư mục, không liên quan đến dữ liệu file

► Thư mục 1 mức:

- Đơn giản nhất
- Có 1 thư mục duy nhất và tất cả các file được giữ trong thư mục này
- Khó chọn tên cho file
- Tìm kiếm file khó

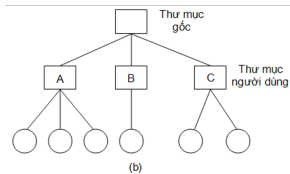


□ = Thư mục ○ = File

Hình 7: Tổ chức thư mục một mức

► Thư mục 2 mức:

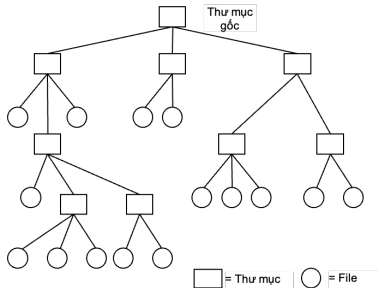
- Phân cho mỗi người dùng 1 thư mục riêng (UFD: User File Directory), chứa các file của mình
- Khi người dùng truy cập file, file sẽ được tìm kiếm trong thư mục ứng với tên người đó
- Các người dùng khác nhau có thể đặt tên file trùng nhau
- Cô lập người dùng, Các file mà nhiều người dùng truy cập tới => chép vào từng thư mục của từng người dùng => lãng phí



Hình 8: Tổ chức thư mục hai mức

► Thư mục cấu trúc cây:

- Thư mục con có thể chứa các thư mục con khác và các files
- Hệ thống thư mục được biểu diễn phân cấp như 1 cây: cành là thư mục, lá là file

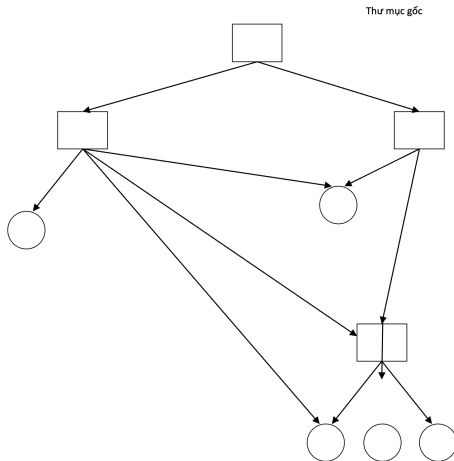


Hình 9: Tổ chức thư mục cây thư mục

► Thư mục cấu trúc cây:

- Phân biệt khoản mục file và khoản mục của thư mục con: thêm bit đặc biệt trong khoản mục
 - 1: khoản mục của thư mục mức dưới
 - 0: khoản mục của file
- Tại mỗi thời điểm, người dùng làm việc với thư mục hiện thời (current directory)
- Tổ chức cây thư mục cho từng đĩa:
 - Trong hệ thống file như FAT của DOS, cây thư mục được xây cho từng đĩa. Hệ thống thư mục được coi là rừng, mỗi cây trên 1 đĩa
 - Linux: toàn hệ thống chỉ gồm 1 cây thư mục

► Thư mục cấu trúc đồ thị không tuần hoàn (acyclic graph):



- Chia sẻ files và thư mục để có thể xuất hiện ở nhiều thư mục riêng khác nhau
- Đây là một cấu trúc đồ thị không chứa chu trình, là mở rộng của cấu trúc cây
- Cây: lá và cành có thể đồng thời thuộc về những cành khác nhau
- Triển khai:
 - ▶ Sử dụng liên kết: con trỏ tới thư mục hoặc file khác
 - ▶ Tạo bản sao của file và thư mục cần chia sẻ và chứa vào các thư mục khác nhau => phải đảm bảo tính đồng bộ và nhất quán => tránh sao lưu chồng chéo, thống kê một file nhiều lần.
- Mềm dẻo nhưng phức tạp

- ▶ Mô tả vị trí của file trong thư mục
- ▶ Đường dẫn tuyệt đối:
 - Đường dẫn từ gốc của cây thư mục, đi qua các thư mục trung gian, dẫn tới file
Ví dụ: C: -> bc -> bin -> bc.exe
- ▶ Đường dẫn tương đối:
 - Tính từ thư mục hiện thời
 - Thêm 2 khoản mục đặc biệt trong thư mục: "." để biểu diễn thư mục hiện tại, và ".." cho thư mục mức trên.

► Danh sách:

- Tổ chức thư mục dưới dạng danh sách các khoản mục
- Tìm kiếm khoản mục được thực hiện bằng cách duyệt lần lượt danh sách
- Thêm file mới vào thư mục:
 - Duyệt cả thư mục để kiểm tra xem khoản mục với tên file như vậy đã có chưa
 - Khoản mục mới được thêm vào cuối danh sách hoặc 1 ô trong bảng
- Mở file, xóa file
- Tìm kiếm trong danh sách chậm
- Lưu trữ thư mục trong bộ nhớ

► Cây nhị phân:

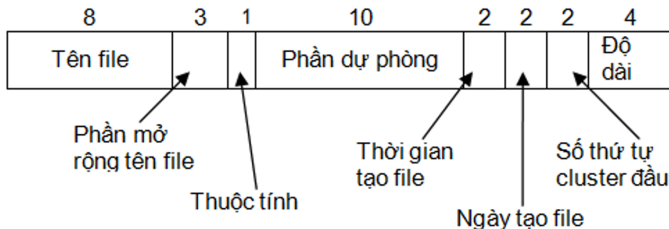
- Tăng tốc độ tìm kiếm nhờ CTDL có hỗ trợ sắp xếp
- Hệ thống file NTFS của WinNT

► Bảng băm (hash table):

- Dùng hàm băm để tính vị trí của khoản mục trong thư mục theo tên file
- Thời gian tìm kiếm nhanh
- Hàm băm phụ thuộc vào kích thước của bảng băm \Rightarrow kích thước bảng cố định

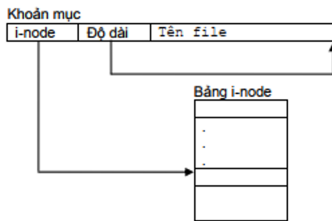
► Tổ chức thư mục của DOS:

- Mỗi đĩa logic có cây thư mục riêng, bắt đầu từ thư mục gốc ROOT
- Thư mục gốc được đặt ở phần đầu của đĩa, ngay sau sector khởi động BOOT và bảng FAT
- Thư mục gốc chứa files và các thư mục con
- Thư mục con có thể chứa files và các thư mục cấp dưới nữa
- Được tổ chức dưới dạng bảng: mỗi khoản mục chiếm 1 dòng trong bảng và có kích thước cố định 32 bytes



► Tổ chức thư mục của Linux:

- Thư mục hệ thống file Ext2 của Linux có cách tổ chức đơn giản
- Khoản mục chứa tên file và địa chỉ l-node
- Thông tin còn lại về các thuộc tính file và vị trí các khối dữ liệu được lưu trên l-node chứ không phải thư mục
- Kích thước khoản mục phụ thuộc vào độ dài tên file
- Phần đầu của khoản mục có trường cho biết kích thước khoản mục



Chương 2

- ▶ Các khái niệm
- ▶ Các phương pháp truy cập file
- ▶ Các thao tác với file
- ▶ Thư mục

Tiếp theo

- ▶ Cấp phát không gian cho file
- ▶ Quản lý không gian trống trên đĩa
- ▶ Độ tin cậy của hệ thống file
- ▶ Bảo mật cho hệ thống file
- ▶ Hệ thống file FAT