Chương 2

HỆ THỐNG FILE

Nội dung chương 1

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. Cấu trúc hệ thống file
10. Hệ thống file FAT

Các khái niệm

- File được định nghĩa như tập hợp các thông tin liên quan đến nhau được đặt tên và được lưu trữ trên bộ nhớ ngoài.
- Thuộc tính của file:
  - 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.

Các khái niệm

- Cấu trúc file:
  - Các thông tin trong file có thể rất khác nhau. Vì vậy, cấu trúc của file cũng rất khác nhau và phụ thuộc vào thông tin chứa trong file.
  - Các HDH chỉ coi file là tập hợp các byte không cấu trúc.

Các phương pháp truy cập file

- Truy cập tuần tự:
  - ✓ Thông tin được đọc, ghi theo từng byte/ bản ghi lần lượt từ đầu file.
  - ✓ Dụng một con trỏ để định vị vị trí hiện thời trong file.
- 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.
- Truy cập dựa trên chỉ số:
  - ✓ File chứa một 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.

Các thao tác với file

- Tạo file: Tạo file trống chưa có data; được dành một chỗ trong thư mục.
- Xóa file:
  - ✓ Giải phóng không gian mà dữ liệu của file chiếm.
  - ✓ Giải phóng chỗ của file trong thư mục.
- Mở file:
  - ✓ Thực hiện trước khi ghi và đọc file.
  - ✓ Đọc các thuộc tính của file vào bộ nhớ để tăng tốc độ.
- Đóng file: Xóa các thông tin về file ra khỏi bảng trong bộ nhớ.
- Ghi vào file.
- Đọc file.

Thư mục

- Số lượng file lưu trữ trên đĩa rất lớn nên phải tổ chức để dễ dàng quản lý, truy cập các file.
- Không gian trên đĩa được chia thành các phần (partition/volume) gọi là đĩa logic.
- Để quản lý file trên các đĩa logic, thông tin về file được lưu trong thư mục của đĩa.
- Khoản mục chứa các thông tin về file: tên, kích thước, vị trí, kiểu file,… hoặc con trỏ tới nơi lưu trữ thông tin này.
- Coi thư mục như 1 bảng, mỗi dòng là khoản mục ứng với 1 file.

Thư mục

- 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 nên kích thước khoản mục, thư mục lớn.
  - ✓ Thư mục chỉ lưu thông tin tối thiểu cần thiết cho việc tìm kiếm vị trí file trên đĩa nên kích thước giảm.

Thư mục

- Mở file:
  - Hệ điều hành 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 trỏ tới cấu trúc dữ liệu khác chứa thuộc tính file, cấu trúc này sẽ được đọc vào bảng.

Thư mục

- Tìm kiếm file: cấu trúc thư mục phải cho phép tìm kiếm file theo tên file.
- Tạo file: tạo khoản mục mới và thêm vào thư mục
- Xóa file: thông tin về file và khoản mục tương ứng 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 file: chỉ thực hiện với thư mục mà không liên quan đến dữ liệu của file.

Thư mục

- Thư mục một mức:
  - Đơn giản nhất
  - Chỉ 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 hai 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.

Thư mục

- Thư mục cấu trúc cây:
  - Con của một thư mục là các file hoặc thư mục khác.
  - Hệ thống thư mục được biểu diễn phân cấp như một cây: cành là thư mục, lá là file.
  - Phân biệt khoản mục file và khoản mục của thư mục con: thêm bit đặc biệt trong khoản mục:
    - ✓ 0: khoản mục của file,
    - ✓ 1: Khoản mục của thư mục mức dưới.
  - Tại mỗi thời điểm, người dùng làm việc với thư mục hiện thời (current directory).
  - Tổ chức cây thư mục cho từng đĩa:
    - ✓ FAT và DOS: Mỗi đĩa một cây thư mục.
    - ✓ Linux: Một cây thư mục cho toàn hệ thống

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.
  - Mở rộng của cấu trúc 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.
  - Mềm dẻo nhưng phức tạp.

Thư mục

- Đường dẫn:
  - 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
    - ✓ C:\Program Files\WinRAR\WinRAR.exe
  - Đường dẫn tương đối:
    - ✓ Tính từ thư mục hiện thời
    - ✓ Thêm 2 khoản mục đặc biệt trong thư mục: “.”, “..”

Thư mục

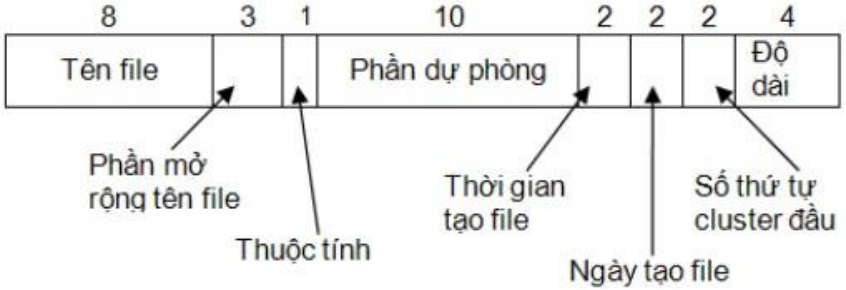
- Tổ chức bên trong của thư mục:
  - Tổ chức thư mục dạng danh sách các khoản mục.
  - Tìm kiếm khoản mục được thực hiện bằng cách duyệt lần lượt danh sách.
  - Thêm file mới vào thư mục:
    - ✓ Duyệt cả thư mục để kiểm tra xem khoản mục với tên file như vậy đã có chưa.
    - ✓ Khoản mục mới được thêm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

Thư mục

- Tổ chức bên trong của thư mục:
  - 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 => kích thước bảng cố định.

Thư mục

- Tổ chức bên trong của thư mục:
  - Tổ chức thư mục của DOS:
    - ✓ Mỗi đĩa logic có cây một thư mục, gốc là thư mục 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 hoặc thư mục khác.
    - ✓ Được tổ chức dưới dạng bảng: mỗi khoản mục chiếm 1 dòng trong bảng và có kích thước cố định 32 bytes



Thư mục

- Tổ chức bên trong của thư mục:
  - 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 I-node
    - ✓ Thông tin còn lại về các thuộc tính file và vị trí các khối dữ liệu được lưu trên I-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

Cấp phát không gian cho file

- Phép ánh xạ file: chỉ ra vị trí file dựa vào tên file.
- Sơ bộ về tổ chức đĩa:
  - ✓ Thông tin được đọc/ghi theo từng khối sector
  - ✓ Nhóm các sector thành block hay cluster (khối)
- Một file gồm một tập các khối. Hệ điều hành chịu trách nhiệm cấp phát các khối cho file:
  - ✓ Không gian trên đĩa phải được cấp phát cho file.
  - ✓ Quản lý không gian trống để sẵn sàng cấp phát
- Một số vấn đề:
  - ✓ Không gian tối đa một lần cần cấp phát cho file?
  - ✓ Không gian cấp phát cho file gọi là phần (portion). Kích thước phần như thế nào?
  - ✓ Theo dõi các phần được gán cho 1 file.

Cấp phát không gian cho file

- Cấp phát trước và cấp phát động:
  - ✓ Cấp phát trước: phải xác định kích thước tối đa của file lúc tạo; khó xác định, gây lãng phí.
  - ✓ Cấp phát động: cấp không gian cho 1 file theo các phần như mong muốn.
- Kích thước phần:
  - ✓ Tính liên tục của không gian làm tăng hiệu năng.
  - ✓ Nhiều phần nhỏ làm tăng kích thước bảng quản lý thông tin cấp phát.
  - ✓ Các phần có kích thước cố định: đơn giản hóa quá trình cấp phát lại không gian.
  - ✓ Các phần có kích thước thay đổi hoặc nhỏ và cố định: tối thiểu hóa lãng phí không gian chưa sử dụng

1

Cấp phát không gian cho file

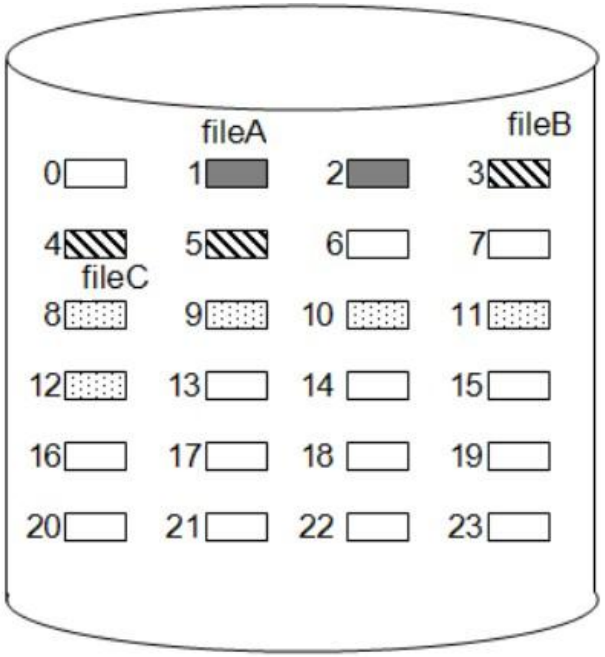
- Có hai lựa chọn:
  - ✓ Các phần liên tục, lớn, thay đổi: hiệu năng tốt, tiết kiệm, bảng cấp phát nhỏ; khó sử dụng lại không gian.
  - ✓ Các khối: các phần nhỏ cố định; linh hoạt cao; cần bảng lớn, cấu trúc phức tạp để quản lý.
- Các lựa chọn trên phù hợp với cấp phát trước, hoặc cấp phát động. Với các phần kích thước thay đổi, có thể chọn:
  - ✓ Phù hợp đầu tiên: chọn nhóm các khối liên tục chưa sử dụng đầu tiên có kích thước phù hợp.
  - ✓ Phù hợp nhất: nhóm chưa sử dụng nhỏ nhất có kích thước phù hợp.
  - ✓ Phù hợp gần nhất: nhóm có kích thước phù hợp, chưa sử dụng và gần với cấp phát cho file đó lần trước nhất.

Cấp phát không gian cho file

- Cấp phát khối liên tiếp:
  - Được cấp phát 1 khoảng không gian gồm các khối liên tiếp trên đĩa.
  - Vị trí file trên đĩa được xác định bởi vị trí khối đầu tiên và độ dài (số khối) mà file đó chiếm.
  - Khi có yêu cầu cấp phát, HDH sẽ chọn 1 vùng trống có số lượng khối đủ cấp cho file đó.
  - Bảng cấp phát file chỉ cần 1 khoản mục cho 1 file, chỉ ra khối bắt đầu, và độ dài của file tính bằng khối.
  - Là cấp phát trước, sử dụng kích thước phần thay đổi.

Cấp phát không gian cho file

- Cấp phát khối liên tiếp:
  - Ưu điểm:
    - ✓ Cho phép truy cập trực tiếp và tuần tự
    - ✓ Đơn giản, tốc độ cao
  - Nhược điểm:
    - ✓ Phải biết trước kích thước file khi tạo Khó
    - ✓ tìm chỗ cho file Gây
    - ✓ phân mảnh ngoài: thay đổi.



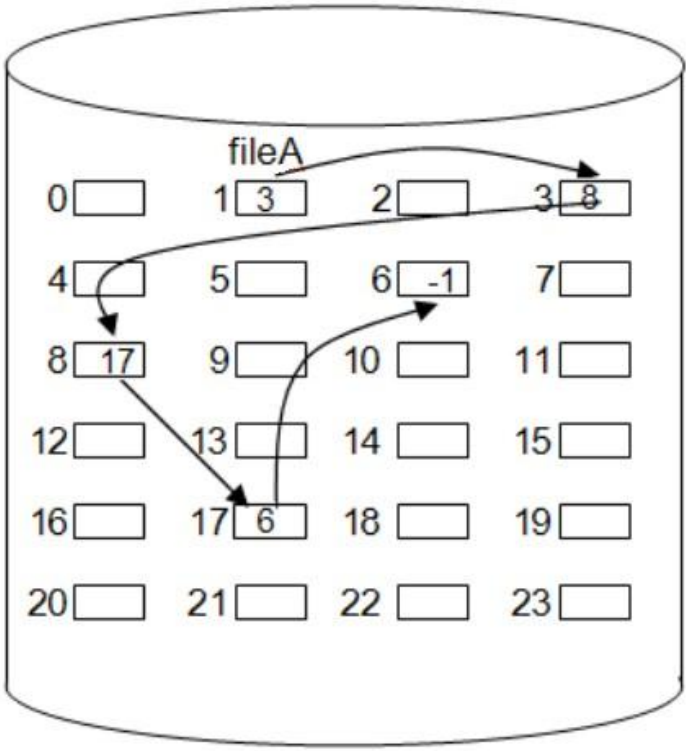
Thư mục		
Tên file	Bắt đầu	Độ dài
fileA	1	2
fileB	3	3
fileC	8	5

Cấp phát không gian cho file

- Sử dụng danh sách kết nối:
  - Các khối được kết nối với nhau thành danh sách kết nối: phần đầu mỗi khối chứa con trỏ trỏ tới khối kế tiếp.
  - Các khối thuộc về 1 file có thể nằm ở vị trí bất kì trên đĩa.
  - Khoản mục của thư mục chứa con trỏ tới khối đầu tiên của file.
  - Khi file được cấp thêm khối mới, khối đó được thêm vào cuối danh sách.
  - Hệ điều hành đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo.

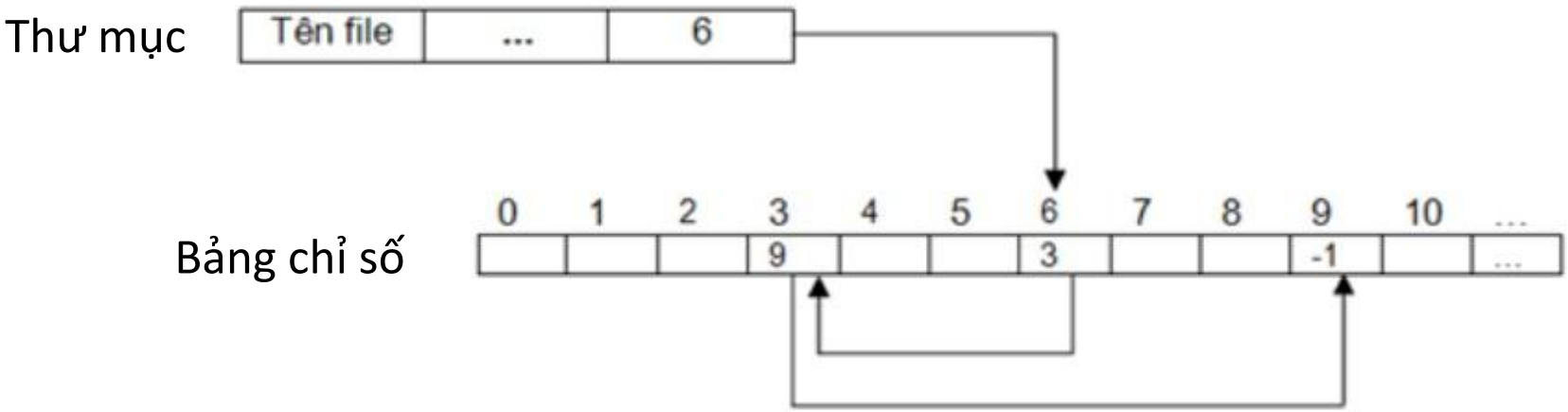
Cấp phát không gian cho file

- Sử dụng danh sách kết nối:
  - Ưu điểm:
    - ✓ Không bị phân mảnh ngoài
    - ✓ Không yêu cầu biết trước kích thước file lúc tạo
    - ✓ Dễ tìm vị trí cho file, khoản mục đơn giản
  - Nhược điểm:
    - ✓ Không hỗ trợ truy cập trực tiếp
    - ✓ Tốc độ truy cập không cao
    - ✓ Giảm độ tin cậy và tính toàn vẹn của hệ thống file



Cấp phát không gian cho file

- Sử dụng danh sách kết nối trên bảng chỉ số:
  - Bảng chỉ số: mỗi ô của bảng ứng với một khối của đĩa
  - Con trỏ tới khối tiếp theo của file được chứa trong ô tương ứng của bảng.
  - Mỗi đĩa logic có 1 bảng chỉ số được lưu ở vị trí xác định
  - Kích thước mỗi ô trên bảng phụ thuộc vào số lượng khối trên đĩa.



Cấp phát không gian cho file

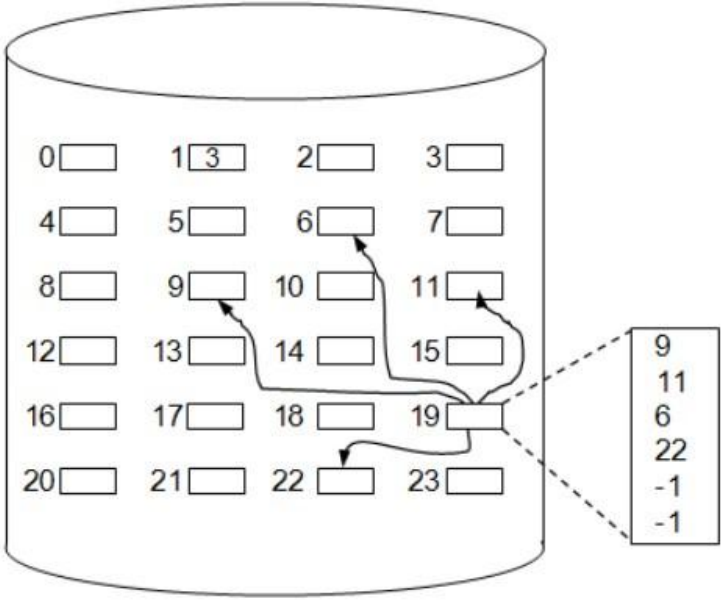
- Sử dụng danh sách kết nối trên bảng chỉ số:
  - Cho phép tiến hành truy cập file trực tiếp: đi theo chuỗi con trỏ chứa trong bảng chỉ mục.
  - Bảng FAT (File Allocation Table): được lưu ở đầu mỗi đĩa logic sau sector khởi động.
  - FAT12, FAT16, FAT32: mỗi ô của bảng có kích thước 12, 16, 32 bit.

Cấp phát không gian cho file

- Sử dụng khối chỉ mục:
  - Tất cả con trỏ tới các khối thuộc về 1 file được tập trung 1 chỗ.
  - Mỗi file có một mảng riêng của mình chứa trong một khối gọi là khối chỉ mục (I-node).
  - Mảng chứa thuộc tính của file và vị trí các khối của file trên đĩa.
  - Ô thứ i của mảng chứa con trỏ tới khối thứ i của file.
  - Khoản mục của file trong thư mục chứa con trỏ tới khối chỉ mục này

Cấp phát không gian cho file

- Sử dụng khối chỉ mục:
  - Chọn kích thước I-node:
    - ✓ Nhỏ: tiết kiệm không gian nhưng không đủ con trỏ tới các khối nếu file lớn.
    - ✓ Lớn: với file nhỏ chỉ chiếm 1 vài ô thì lãng phí.
    - ✓ Giải pháp:.
      - Thay đổi kích thước i-node = sử dụng danh sách kết nối.
      - Sử dụng I-node có cấu trúc nhiều mức

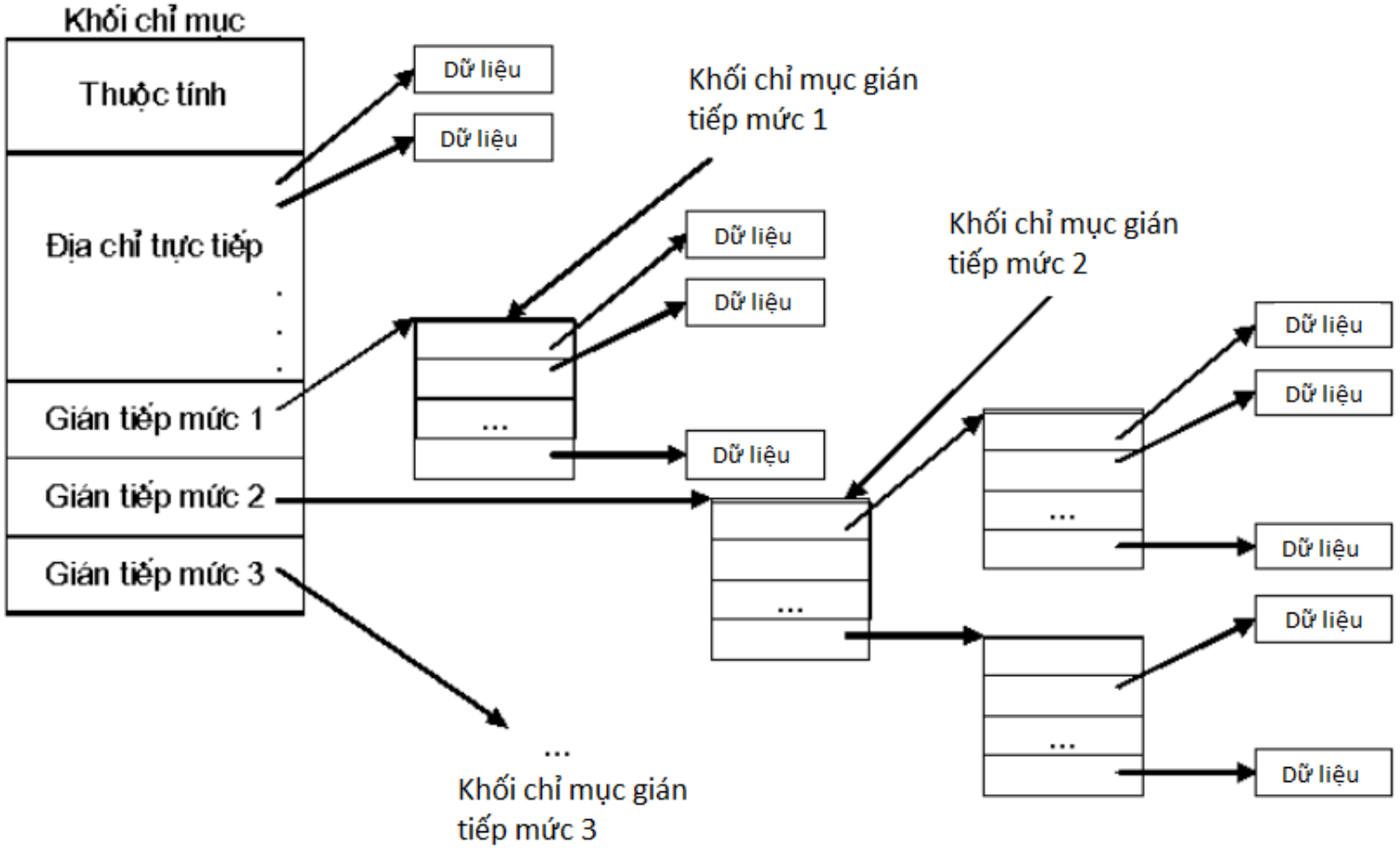


Thư mục

Tên file	Khối chỉ số
fileA	19

Cấp phát không gian cho file

- Sử dụng khối chỉ mục:



Cấp phát không gian cho file

- Sử dụng khối chỉ mục:
  - Ưu điểm:
    - ✓ Cho phép truy cập trực tiếp.
    - ✓ Các khối thuộc 1 file không cần nằm liên tiếp nhau.
  - Nhược điểm:
    - ✓ Tốc độ truy cập file chậm

Quản lý không gian trống trên đĩa

- Kích thước khối:
  - Kích thước khối lớn:
    - ✓ Giảm kích thước bảng chỉ mục, tăng tốc độ đọc file;
    - ✓ Bị phân mảnh trong.
  - Kích thước khối nhỏ:.
    - ✓ Mỗi file chiếm nhiều khối nhớ, nằm rải rác trên đĩa.
    - ✓ Thời gian đọc file lâu.
  - Chọn kích thước khối tùy thuộc:
    - ✓ Kích thước đĩa: đĩa lớn, chọn kích thước khối lớn => thời gian truy cập nhanh, đơn giản hóa việc quản lý.
    - ✓ Kích thước file: hệ thống sử dụng nhiều file lớn, kích thước tăng và ngược lại.
  - Kích thước khối thường là lũy thừa 2 của sector và nằm trong khoảng từ 512B tới 32 KB.

Quản lý không gian trống trên đĩa

- Danh sách kết nối:
  - Vector bit là mảng 1 chiều.
  - Mỗi ô có kích thước 1 bit tương ứng với một khối trên đĩa.
  - Khối được cấp phát có bít tương ứng là 0, khối trống: 1
    hoặc ngược lại.
  - Dễ tìm 1 hoặc nhóm các khối trống liên tiếp.
  - Với đĩa có kích thước lớn, đọc toàn bộ vector bit vào bộ nhớ có thể đòi hỏi khá nhiều không gian nhớ.

Quản lý không gian trống trên đĩa

- Danh sách vùng trống:
  - Các khối nằm liền nhau thường được cấp phát và giải phóng đồng thời.
  - Lưu vị trí khối trống đầu tiên của vùng các khối trống liên tiếp và số lượng các khối trống nằm liền sau đó.
  - Thông tin trên được lưu vào danh sách riêng.

Độ tin cậy của hệ thống file

- Phát hiện và loại trừ các khối hỏng:
  - Phương pháp 1:
    - ✓ Một sector trên đĩa được dành riêng chứa danh sách các khối hỏng.
    - ✓ Một số khối không hỏng được dành riêng để dự trữ.
    - ✓ Các khối hỏng sẽ được thay thế bởi các khối dự trữ bằng cách thay thế địa chỉ.
    - ✓ Truy cập tới khối hỏng thành truy cập tới khối dự trữ.
  - Phương pháp 2:
    - ✓ Tập trung tất cả các khối hỏng thành 1 file.
    - ✓ Vì thế, xem như đã cấp phát và không được sử dụng nữa.

Độ tin cậy của hệ thống file

- Sao dự phòng:
  - Tạo ra một bản sao của đĩa trên một vật mang khác.
  - Sao lưu toàn bộ (full backup):
    - ✓ Ghi toàn bộ thông tin trên đĩa ra băng từ.
    - ✓ Chắc chắn nhưng tốn nhiều thời gian.
  - Sao lưu tăng dần (incremental backup):
    - ✓ Dùng sau khi đã tiến hành full backup ít nhất 1 lần.
    - ✓ Chỉ ghi lại các file có thay đổi sau lần sao lưu cuối.
    - ✓ Hệ thống lưu trữ thông tin về các lần lưu trữ file.
    - ✓ DOS: file thay đổi, archive bit =1.
  - Kết hợp:
    - ✓ Full backup: hàng tuần/ tháng.
    - ✓ Incremental baclup: hàng ngày.

Độ tin cậy của hệ thống file

- Kiểm tra tính toàn vẹn của hệ thống file:
  - Hệ thống file chứa nhiều CTDL có mối liên kết nên nếu thông tin về liên kết bị hư hại, tính toàn vẹn của hệ thống bị phá vỡ.
  - Các khối không có mặt trong danh sách các khối trống, đồng thời cũng không có mặt trong một file nào.
  - Một khối có thể vừa thuộc về một file nào đó vừa có mặt trong danh sách khối trống.
  - Hệ điều hành có các chương trình kiểm tra tính toàn vẹn của hệ thống file, được chạy khi hệ thống khởi động, đặc biệt là sau sự cố

Độ tin cậy của hệ thống file

- Đảm bảo tính toàn vẹn bằng cách sử dụng giao tác:
  - Giao tác (transaction) là một tập hợp các thao tác cần phải được thực hiện trọn vẹn cùng với nhau. Với
  - hệ thống file: mỗi giao tác sẽ bao gồm những thao tác thay đổi liên kết cần thực hiện cùng nhau. Toàn
  - bộ trạng thái hệ thống file được ghi lại trong file log. Nếu giao tác không được thực hiện trọn vẹn, hệ điều
  - hành sử dụng thông tin từ log để khôi phục hệ thống file về trạng thái không lỗi trước khi thực hiện giao tác.

Bảo mật cho hệ thống file

- Đảm bảo tính toàn vẹn bằng cách sử dụng giao tác:
  - Ngăn cản việc truy cập trái phép các thông tin lưu trữ trong file và thư mục.
  - Hạn chế các thao tác truy cập tới file hoặc thư mục.
  - Dùng mật khẩu:
    - Người dùng phải nhớ nhiều mật khẩu.
    - Mỗi khi thao tác với tài nguyên lại gõ mật khẩu.

Bảo mật cho hệ thống file

- Đảm bảo tính toàn vẹn bằng cách sử dụng giao tác:
  - Dùng danh sách quản lý truy cập ACL (Access Control List):
  - Mỗi file được gán danh sách đi kèm, chứa thông tin định danh người dùng và các quyền người đó được thực hiện với file.
  - ACL thường được lưu trữ như thuộc tính của file/thư mục.
  - Thường được sử dụng cùng với cơ chế đăng nhập.
  - Các quyền truy cập cơ bản:
    - ✓ Quyền đọc (r).
    - ✓ Quyền ghi, thay đổi (w).
    - ✓ Quyền xóa.
    - ✓ Quyền thay đổi chủ file (change owner)

Cấu trúc hệ thống file

Trình ứng dụng
Hệ thống file logic
Module tổ chức file
Hệ thống file cơ sở
Quản lý nhập/xuất
Thiết bị nhớ ngoài

- Quản lý vào ra:
  - Gồm các chương trình điều khiển thiết bị (driver) và chương trình xử lý ngắt cứng.
  - Lưu chuyển thông tin và dữ liệu giữa bộ nhớ và thiết bị nhớ ngoài.
  - Nhận các lệnh đọc/ghi thông tin từ lớp trên; dịch các yêu cầu;

Cấu trúc hệ thống file

- Hệ thống file cơ sở:
  - Sinh ra các lệnh đọc/ghi khối nhớ cụ thể cho lớp dưới.
  - Trung chuyển các khối tin giữa lớp dưới và bộ nhớ.
  - Không cần hiểu nội dung thông tin và tổ chức file.
- Modul tổ chức file:
  - Ánh xạ giữa khối logic và vật lý.
  - Quản lý các khối trống chưa được cấp phát trên đĩa.
- Hệ thống file logic:
  - Quản lý thư mục.
  - Cung cấp thông tin vị trí file.
  - Thực hiện các chức năng liên quan đến bảo mật và đảm bảo toàn vẹn cho hệ thống file.

Hệ thống file FAT (File Allocation Table)

- Có 3 phiên bản: FAT12, FAT16, FAT32. Chữ số chỉ kích thước ô bảng FAT tương ứng là 12, 16 và 32 bit.
- Đơn vị cấp phát không gian trên đĩa (khối logic) là cluster (lũy thừa 2 của số lượng sector).

Boot sector và các khối dự phòng	Bảng FAT1	Bảng FAT2	Thư mục gốc (chỉ có trên FAT12 và FAT16)	Phần còn lại cho tới cuối đĩa chứa các file và thư mục của đĩa lô gic

Hệ thống file FAT (File Allocation Table)

- Boot sector:
  - Sector đầu tiên của đĩa logic.
  - Chứa thông tin mô tả cấu trúc đĩa logic: kích thước sector, cluster, kích thước bảng FAT.
  - Chứa mã chương trình mồi để tải hệ điều hành nếu đĩa logic là đĩa khởi động.
- FAT: bảng chỉ số quản lý cấp phát khối cho file.
- Thư mục gốc ROOT.
- Vùng dữ liệu: chứa các file và thư mục của đĩa logic

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
EAST ASIA UNIVERSITY OF TECHNOLOGY

Hệ thống file FAT (File Allocation Table)

- Boot sector - 32 byte đầu tiên:

Vj tri	Oqd ai	Y nghia
0	3	Lenh Jump. Chi thi cho CPU bo qua phfin thong tin va nht y t6i thuc hi□n phdn ma m6i cua h□di□u hanh nfu day la dia m6i h□dieu hanh.
.)	8	Ten hang san xuat, b6 sung dfiu trang Ccu6i cho du 88 . Vi d11: IBM 3.3, MSDOS5.0.v.v.
11	2	Bytes per sector. Kich thuoc sector tinh bing byte. Gia rri thulmg g□ la 512 d6i v6i dia cirng. E>iiy cilng la vi tri biit dfiu cua Kh6i Thong s6 BIOS (BIOS Parameter Block, vi□ tilt la BPB)
1B	I	Sectors per cluster. SB sector trong m9t cluster, luon la lily thira cua 2 va khong lan hon 12 8.
14	2	Reserved sectors. S6 hrong sector danh cho vung dfiu dia dtln truoc FAT, bao g6m boot sector va cac sector d\f phong.
16	I	S6 hr ong bang FAT. Thuong bilng 2.
17	2	S6 khoan rnuc t6i da trong thu muc g6c ROOT. Chi sir dung cho FAT12 va FATl6 . Bilng O v6i FAT32.
19	2	Total sector. T6ng s6 sector rren dia. Neu bilng khong thi s6 hreng sector dugc ghi bing 4 byte t:ii vi trl Ox20.
21	I	Mo ta lo{li dia. Vi du OxFO la dia 016 111 3.5" hai m□ v6i 80 r anh tren moi m{it, OxFI la dia orng .v.v.
22	2	Sectors per FAT. Kich thu6c FAT tinh bfing sector (d6i v6i FAT12/16)
24	2	Sectors per track. S6 sector trcn m{jt ranh.
26	2	Number of heads. S6 IIr(Jllg dfiu doc (m□ dia dul,JC su dung)
28	4	Hidden sectors. S6 IIr(Jllg sector i\n.
32	4	Total sector. T6ng s6 sector tren dia cho rruong hep c6 nhi□u hon 65535.

Hệ thống file FAT (File Allocation Table)

- Boot sector – Các byte tiếp theo với FAT12/16:

Vị trí	Độ dài	Ý nghĩa
36	1	Số thứ tự vật lý của đĩa (0: đĩa mềm, 80h: đĩa cứng .v.v.)
37	1	Dự phòng
38	1	Dấu hiệu của phần mã mồi. Chứa giá trị 0x29 (ký tự ')') hoặc 0x28.
39	4	Số xê ri của đĩa (Volume Serial Number) được tạo lúc format đĩa
43	11	Volume Label. Nhãn của đĩa được tạo khi format.
54	8	Tên hệ thống file FAT, ví dụ"FAT12 ", "FAT16 ".
62	448	Mã mồi hệ điều hành, đây là phần chương trình tải hệ điều hành khi khởi động.
510	2	Dấu hiệu Boot sector (0x55 0xAA)

Hệ thống file FAT (File Allocation Table)

- Boot sector – Các byte tiếp theo với FAT32:

Vị trí	Độ dài	Ý nghĩa
36	4	Sectors per FAT. Kích thước FAT tính bằng sector.
0x28	2	Cờ của FAT
0x2a	2	Version. Phiên bản.
0x2c	4	Số thứ tự của cluster đầu tiên của thư mục gốc root.
0x30	2	Số sector của Information Sector. Đây là phần nằm trong số sector dự phòng ngay sau boot sector.
0x32	2	Số thứ tự sector đầu tiên của bản sao của boot sector (nếu có)
0x34	12	Dự phòng
0x40	1	Số thứ tự vật lý của đĩa
0x41	1	Dự phòng
0x42	1	Dấu hiệu của phần mã mồi mở rộng.
0x43	4	Số xê ri của đĩa (Volume Serial Number)
0x47	11	Volume Label
0x52	8	"FAT32 "
0x5a	420	Mã mồi hệ điều hành
0x1FE	2	Dấu hiệu Boot sector (0x55 0xAA)

Hệ thống file FAT (File Allocation Table)

- Bảng FAT:
  - Quản lý các cluster trên đĩa và các file theo nguyên tắc:
    - ✓ Các khối thuộc cùng 1 file được liên kết thành 1 danh sách.
    - ✓ Con trỏ được chứa trong ô tương ứng của bảng FAT.
  - Mỗi ô trong bảng FAT tương ứng với một cluster trên đĩa, chứa một trong các thông tin:
    - ✓ Stt cluster tiếp theo trong danh sách các khối của file.
    - ✓ Dấu hiệu kết thúc nếu ô tương ứng với cluster cuối cùng của file.
    - ✓ Ký hiệu đánh dấu cluster hỏng, không được sử dụng.
    - ✓ Dấu hiệu đánh dấu cluster dự phòng.
    - ✓ Bằng 0 nếu cluster trống, chưa cấp phát cho file nào.

Hệ thống file FAT (File Allocation Table)

- Bảng FAT:
  - Cluster đầu tiên của vùng dữ liệu được đánh STT là 2.
  - 2 ô đầu tiên của bảng FAT không dùng để quản lý

FAT12	FAT16	FAT32	Ý nghĩa
0x000	0x0000	0x00000000	Cluster trống
0x001	0x0001	0x00000001	Cluster dự phòng, không được sử dụng
0x002–0xFEF	0x0002–0xFFEF	0x00000002–0x0FFFFFEF	Cluster đã được cấp cho file. Chứa số thứ tự cluster tiếp theo của file.
0xFF0–0xFF6	0xFFF0–0xFFF6	0x0FFFFFF0–0x0FFFFFF6	Cluster dự phòng
0xFF7	0xFFF7	0x0FFFFFF7	Cluster hỏng.
0xFF8–0xFFF	0xFFF8–0xFFFF	0x0FFFFFF8–0x0FFFFFFF	Cluster cuối cùng của file

Hệ thống file FAT (File Allocation Table)

- Root – Thư mục gốc:
  - Mỗi thư mục được lưu trong bảng thư mục, thực chất là 1 file đặc biệt chứa các khoản mục của thư mục.
  - Mỗi khoản mục chứa thông tin về một file hoặc thư mục con của thư mục đang xét.
  - Với FAT12/16, thư mục trên cùng của đĩa được chứa trong 1 vùng đặc biệt gọi là thư mục gốc.
  - Các thư mục mức thấp hơn/ thư mục gốc của FAT32 được chứa trong vùng dữ liệu trên đĩa cùng với các file.
  - Mỗi thư mục gồm các khoản mục 32 byte xếp liền nhau.

Hệ thống file FAT (File Allocation Table)

- Đọc FAT:
  - Vị trí sector bắt đầu: reserved sector (byte 14, 15 trong bootsector).
  - Tổng số sector cần đọc: sectors per FAT (byte 22, 23).
  - Nội dung bảng FAT đã được đọc ra vùng buf.
  - FAT16: ô FAT thứ n = buf+n*2.
  - Hàm đọc đĩa:

```
int absread(int drive, int nsects, long lsect, void *buffer);
```

    - drive: ổ đĩa cần đọc, A: 0, B:1, C:2.
    - nsects: số sector cần đọc.
    - lsect: vị trí sector bắt đầu đọc.
    - buffer: vùng nhớ lưu nội dung thông tin cần đọc.