



# HỆ ĐIỀU HÀNH

# Operating Systems

1

## NỘI DUNG

---

- Chương 1: Tổng quan
- Chương 2: Quản lý tiến trình
- Chương 3: Deadlock
- Chương 4: Quản lý bộ nhớ
- Chương 5: Hệ thống file**
- Chương 6: Quản lý nhập xuất

---

2

2

## Chương 5

# Hệ thống File



[www.cunghodaptrinh.com](http://www.cunghodaptrinh.com)

3

## Nội dung

---

1. File và thư mục
  2. Hiện thực Hệ thống file
- 

4

## 5.1. File và thư mục

---

- Ý nghĩa
  - File và các khái niệm liên quan
  - Các thao tác với file
  - Thư mục
- 

5

### 5.1.1 Ý nghĩa của hệ thống file

---

- Yêu cầu lưu trữ của máy tính rất lớn
  - Bộ nhớ trong:
    - Có kích thước giới hạn
    - Dữ liệu mất khi mất nguồn điện
    - Giá thành cao
  - → Cần có một không gian lưu trữ lớn, tồn tại độc lập, dễ dàng sao chép, di chuyển, giá thành rẻ: Bộ nhớ ngoài.
  - Cần tổ chức bộ nhớ ngoài để có thể lưu trữ khối lượng lớn dữ liệu, chương trình, khi thực thi → nạp vào bộ nhớ trong
  - → Hệ thống file
- 

6

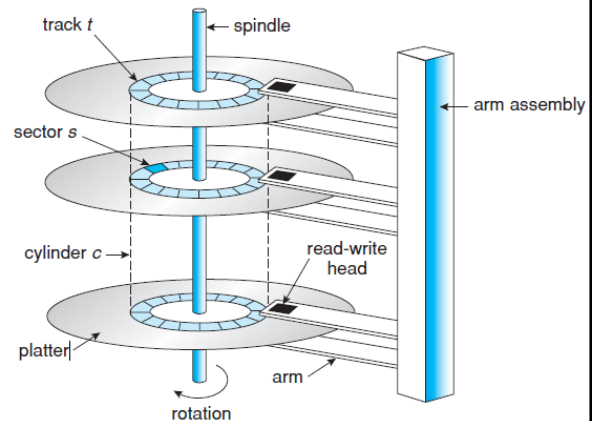
## 5.1.2. File và các khái niệm liên quan

### • Bộ nhớ ngoài:

- Là các thiết bị lưu trữ bên ngoài có khả năng lưu trữ trong thời gian dài
- Chứa lượng thông tin rất lớn

### • Đĩa cứng:

thiết bị lưu trữ thuộc bộ nhớ ngoài



7

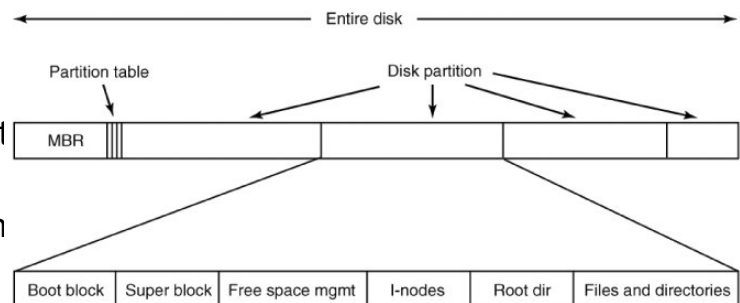
## File và các khái niệm liên quan (tt)

### • Partition (phân vùng):

- Là một tập các sector liên kề trên một đĩa, còn gọi là đĩa luận lý
- Mỗi đĩa vật lý có thể chia thành nhiều partition

### • Volume:

- Một volume tương tự một partition logic trên một đĩa
- Được tạo khi ta định dạng một đĩa hoặc một phần của đĩa theo hệ thống file NTFS.
- Một đĩa có thể có một hoặc nhiều volume độc lập với nhau
- Có thể tạo ra một volume trên đĩa vật lý khác nhau (windows NT/2000)



8

## File và các khái niệm liên quan (tt)

### •Track:

- Đĩa cứng được định dạng thành các vòng tròn đồng tâm gọi là track để ghi dữ liệu
- Track có thể thay đổi vị trí khi định dạng cấp thấp ổ đĩa (*low format*)

### •Sector:

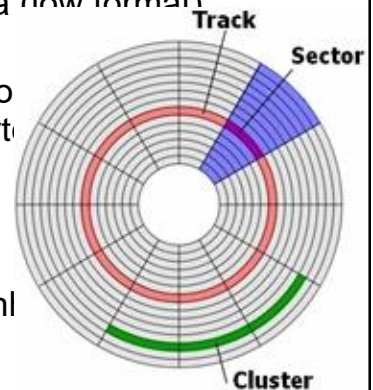
- Mỗi rãnh chia thành nhiều phần bằng nhau gọi là sector
- Thông thường thì một sector chứa dung lượng 512 byte

### •Cluster (khối):

- Gồm một hoặc nhiều sector → tăng hiệu quả truy xuất

### •Cylinder (từ trụ)

- Tập hợp các *track* cùng bán kính ở các mặt đĩa khác nhau
- Cylinder chỉ có trên các ổ đĩa cứng



9

## File và các khái niệm liên quan (tt)

### •File (tập tin):

- Là đơn vị lưu trữ thông tin của bộ nhớ ngoài.
- Các tiến trình có thể đọc hay tạo mới file nếu cần thiết.
- Thông tin trên file luôn tồn tại không bị ảnh hưởng bởi các xử lý tạo hay kết thúc các tiến trình, chỉ mất đi khi user thật sự muốn xóa.
- file được quản lý bởi hệ điều hành.

### •Hệ thống quản lý file

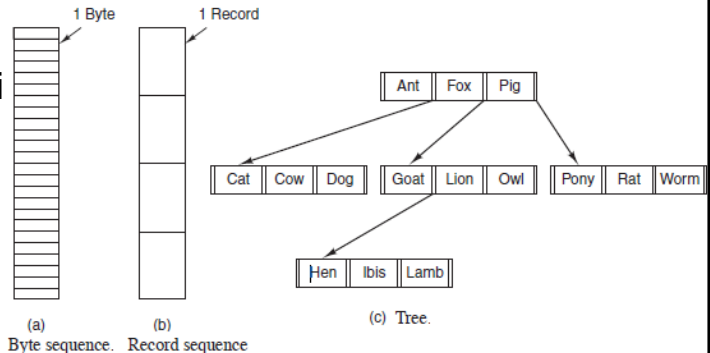
- Các file được quản lý bởi hệ điều hành với cơ chế gọi là hệ thống quản lý file.
- Bao gồm: cách hiển thị, các yếu tố cấu thành file, cách đặt tên, cách truy xuất, cách sử dụng và bảo vệ file, các thao tác trên file.
- Cách tổ chức thư mục, các đặc tính và các thao tác trên thư mục

10

## File và các khái niệm liên quan (tt)

### • Cấu trúc của file: gồm 3 loại:

1. Dãy tuần tự các byte không cấu trúc: hệ điều hành không biết nội dung của file:  
MS-DOS, UNIX, Windows
2. Dãy các record có chiều dài cố định.
3. Cấu trúc cây: gồm cây của những record, không cần thiết có cùng độ dài, mỗi record có một trường khóa giúp cho việc tìm kiếm nhanh hơn



11

## File và các khái niệm liên quan (tt)

### • Kiểu file:

- Các hệ điều hành hỗ trợ nhiều loại file khác nhau bao gồm các kiểu như:
  - Thư mục: là những file hệ thống dùng để lưu giữ cấu trúc của hệ thống file.
  - File có ký tự đặc biệt: liên quan đến nhập xuất thông qua các thiết bị nhập xuất tuần tự như màn hình, máy in, mạng.
  - File khối: dùng để truy xuất trên thiết bị đĩa.
  - File thường: được chia làm hai loại là file văn bản và file nhị phân.

12

## File và các khái niệm liên quan (tt)

### • File văn bản:

- Chứa các dòng văn bản, độ dài có thể khác nhau, cuối dòng có ký hiệu enter.
- Ưu điểm: có thể hiển thị, in hay soạn thảo với một editor thông thường.
- Được dùng để nhập xuất, chứa dữ liệu làm đầu vào và đầu ra cho cơ chế pipeline.

### • File nhị phân:

- Gồm dãy các byte, hệ điều hành chỉ thực thi file này nếu nó có cấu trúc đúng.
- Ví dụ: một file nhị phân thi hành được của UNIX: thường có năm thành phần: header, text, data, relocation bits, symbol table.
  - Header: nhận diện tập tin.
  - Sau đó là 16 bit cho biết kích thước các thành phần của tập tin, địa chỉ bắt đầu thực hiện và một số bit cờ.
  - Sau header là text và data. Nó được nạp vào bộ nhớ và định vị lại bởi những bit relocation.
  - Bảng symbol được dùng để debug

13

## 5.1.3 Thuộc tính của file

- **Tên (name)**: là thông tin được lưu ở dạng mà người dùng có thể đọc
- **Định danh (identifier)**: là thẻ duy nhất, thường là số, dùng để xác định file trong hệ thống file, người dùng không thể đọc
- **Kiểu (type)**: này được yêu cầu cho hệ thống hỗ trợ các kiểu khác nhau
- **Vị trí (location)**: con trỏ chỉ tới một thiết bị tới vị trí file trên thiết bị đó.
- **Kích thước (size)**: kích thước hiện hành của file (tính bằng byte, word hay khối) và kích thước cho phép tối đa chứa trong thuộc tính này.
- **Bảo vệ (protection)**: mức độ và quyền truy cập
- **Ngày (date) Giờ (time), và định danh người dùng (user identification)**: thông tin về việc tạo, sửa đổi gần nhất, sử dụng gần nhất → có ích cho việc bảo vệ, bảo mật, và kiểm soát việc sử dụng

14

## 5.1.4 Các thao tác với file

---

- Các thao tác cơ bản:
    - Tạo file
    - Ghi file
    - Đọc file
    - Tái định vị trong file
    - Xóa file
    - Xóa nội dung file
  - Truy xuất file
  - Bảo toàn dữ liệu file
  - Danh sách các quyền truy cập (Access Right)
- 

15

### 5.1.4.1 Các thao tác cơ bản với file

---

- **Tạo file:**
    - Tìm một khối trống cấp cho file
    - Tạo một cấu trúc chứa thông tin của file ghi vào bảng thư mục
  - **Ghi file:**
    - Tìm tên file trong bảng thư mục
    - Lấy được số hiệu khối nhớ đầu tiên cấp phát cho file, ghi dữ liệu vào file bắt đầu từ vị trí này
    - Dùng một con trỏ ghi để ghi nhận vị trí cho lần ghi kế tiếp
  - **Đọc file:**
    - Tìm tên file trong bảng thư mục
    - Biết vị trí lưu trữ file
    - Đọc file vào bộ nhớ
    - Dùng một con trỏ đọc để ghi nhận vị trí cho lần đọc kế tiếp
- 

16



## Các thao tác cơ bản với file (tt)

### •Tái định vị trong file:

- Tìm tên file trong bảng thư mục
- Dùng lệnh seek để thay đổi vị trí con trỏ đọc

### •Xóa file:

- Tìm tên file trong bảng thư mục
- Giải phóng tất cả các khối đĩa mà file chiếm dụng
- Xóa mục tương ứng chứa thông tin của file trong bảng thư mục

### •Xóa nội dung file:

- Tìm tên file trong bảng thư mục
- Thiết lập thuộc tính kích thước file = 0
- Giải phóng vùng nhớ chứa nội dung file

17

## Các thao tác cơ bản với file (tt)

### •Các thao tác trên file thông thường phải tìm file trên bảng thư mục → chậm → HĐH sử dụng bảng open-file để chứa thông tin các file đã mở

- Khi cần thao tác với file, HĐH sẽ tìm file trong bảng open-file
- Khi đóng file, phần tử tương ứng được xóa trong bảng open-file

### •Mở file:

- Lần đầu: thông tin file được đọc từ bảng thư mục và lưu vào bảng open-file trong bộ nhớ
- Các lần sau (khi file chưa đóng): lấy thông tin từ bảng open-file.
- Thủ tục do HĐH cung cấp: **Open** (*tên file cần mở, chế độ mở: đọc/ ghi/ tạo mới*)

### •Đóng file:

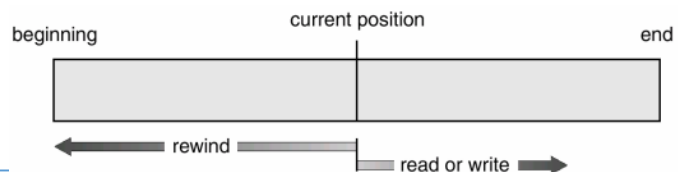
- Ghi mục tương ứng trong bảng open-file vào cấu trúc thư mục và hủy mục này trong bảng open-files
- Thủ tục do HĐH cung cấp: **Close** (*tên file cần đóng*)

18

### 5.1.4.2 Các phương pháp truy xuất file

#### • Truy xuất tuần tự:

- Thông tin trong file được xử lý có thứ tự
- Tiến trình đọc/ghi tất cả các byte trong file theo thứ tự từ đầu (các trình soạn thảo, trình biên dịch) → tự động tăng con trỏ file đến vị trí tiếp theo.
- File có thể tự khởi động lại từ vị trí đầu tiên
- Một số hệ thống file cho phép di chuyển con trỏ file đi tới/ lui n mẫu tin.
- Phù hợp với các loại băng từ
- Có hai cách truy xuất:
  - Đọc bắt đầu ở vị trí đầu file
  - SEEK đến một vị trí trong file



19

### Các phương pháp truy xuất file (tt)

#### • Truy xuất trực tiếp

- Truy xuất không theo thứ tự
- Sử dụng cho loại file được hình thành từ các mẫu tin luận lý có chiều dài không đổi
- Dựa trên mô hình của đĩa, vì đĩa cho phép truy xuất ngẫu nhiên tới bất cứ khối nào tập tin.
- File được hiển thị như một chuỗi các khối hay mẫu tin được đánh số, có thể đọc hoặc ghi các khối bất kỳ.
- Được sử dụng trong trường hợp phải truy xuất một khối lượng thông tin lớn ( ví dụ: cơ sở dữ liệu).
- Một số hệ điều hành không hỗ trợ cả hai kiểu truy xuất

20

## Các phương pháp truy xuất file (tt)

### • Các phương pháp truy xuất khác

- Được xây dựng trên cơ sở của phương pháp truy xuất trực tiếp
- Tạo chỉ mục cho file, chỉ mục chứa các con trỏ chỉ tới các khối khác.
- Để tìm một mẫu tin trong file, trước hết phải tìm chỉ mục → dùng con trỏ để truy xuất file → tìm mẫu tin.
- Với những file lớn → chỉ mục file quá lớn → tạo chỉ mục cho file chỉ mục:
  - File chỉ mục chính chứa các con trỏ chỉ tới các file chỉ mục thứ cấp chỉ tới các thành phần dữ liệu thật sự.

21

## 5.1.4.3 Bảo toàn dữ liệu file

- HĐH phải luôn tạo bản sao cho các file đang mở trên hệ thống, để có thể phục hồi lại khi cần thiết, có hai kỹ thuật là **DUMP có chu kỳ** và **DUMP Incremental**
- **DUMP theo chu kỳ:**
  - Sau một khoảng thời gian, nội dung của các file đang mở trên bộ nhớ chính sẽ được đổ (Dump/backup) ra lại đĩa.
  - Nếu hệ thống gặp sự cố thì tất cả các file đang mở sẽ được tái tạo lại kể từ trạng thái mà chúng được DUMP ra lần cuối cùng.
  - Dump làm tốn thời gian thực hiện của hệ thống.

22

## Bảo toàn dữ liệu file (tt)

### • DUMP Incremental:

- Chỉ lưu các thông tin được sửa đổi kể từ lần Dump sau cùng
- Thông tin cần lưu trữ ít hơn → hệ thống có thể thực hiện Dump thường xuyên hơn
- Sử dụng một trường KTCN dài 2 bit, chứa một trong các giá trị sau:
  - **00**: mở không cập nhật, **01**: mở có cập nhật, **10**: không thay đổi so với lần Dump trước, **11**: có thay đổi so với lần Dump trước.
- Cần thường xuyên kiểm tra bảng danh mục và cập nhật lại trường KTCN sau mỗi lần Dump → chậm tốc độ thực hiện của hệ thống.
- → cài đặt thêm một bảng danh mục để ghi nhận thông tin của các file đang được truy xuất (ghi/đọc) trên hệ thống chỉ dành cho Dump sử dụng → hệ thống Dump có thể hoạt động song song với các thao tác khác của hệ thống.
- Dump Incremental là một tiến trình có độ ưu tiên thấp, thường trú trong bộ nhớ, phân tích các bảng danh mục để tìm ra các file cần phải thực hiện Dump.

23

## 5.1.4.4 Danh sách các quyền truy cập (Access Right)

- Trong các hệ thống đa người dùng (multiuser), quyền truy cập có thể được gán cho **User**, **User Group**, **All User**.
- Khi một group được gán quyền nào đó thì tất cả các user trong group đều được gán quyền truy cập đó.

-rw-rw-r--	1	pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2	jwg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1	pbg	staff	9423	Feb 24 2012	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb 24 2012	program
drwx--x--x	4	tag	faculty	512	Jul 31 10:31	lib/
drwx-----	3	pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul 8 09:35	test/

24

## Danh sách các quyền truy cập (tt)

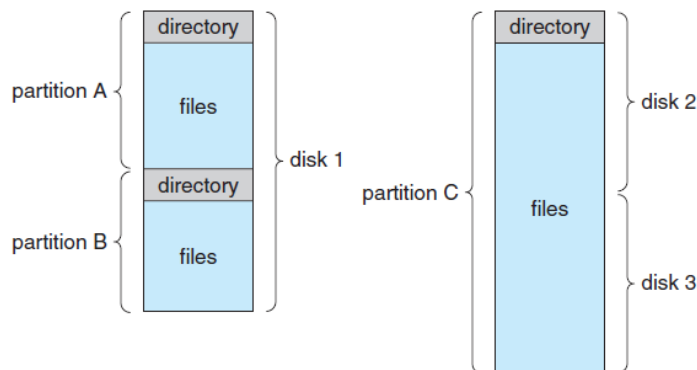
### • Các quyền truy cập của hệ điều hành:

- **None:** không được phép đọc thư mục chứa file này.
- **Knowledge:** có thể xác định được là file đang tồn tại và ai là người sở hữu file
- **Excution:** có thể thực thi một chương trình nhưng không thể copy nó.
- **Reading:** đọc, copy và thực thi. (Một vài hệ thống chỉ cho phép đọc nhưng thể copy nội dung)
- **Appending:** chỉ có thể thêm dữ liệu vào file, thường là ở cuối file,
- **Updating:** có thể thay đổi, xoá và thêm dữ liệu vào file.
- **Changing protection:** có thể thay đổi các quyền truy cập được gán đến user khác (thường chỉ được gán cho user sở hữu file)
- **Deletion:** có thể xoá được file từ hệ thống file.

25

## 5.1.5. Thư mục

- Đĩa chia thành các partition, mỗi partition được xem như một thiết bị lưu trữ riêng
- Mỗi partition chứa thông tin về các file chứa bên trong (volume table of contents)



26

## Thư mục (tt)

---

- Cấu trúc thư mục dùng để quản lý có tổ chức hệ thống tập tin, có thể đáp ứng các thao tác:
    - Tìm kiếm tập tin trong thư mục
    - Tạo một tập tin bên trong thư mục.
    - Xóa tập tin khỏi thư mục.
    - Liệt kê các tập tin trong thư mục.
    - Đổi tên một tập tin.
    - Duyệt hệ thống tập tin
- 

27

## Thư mục (tt)

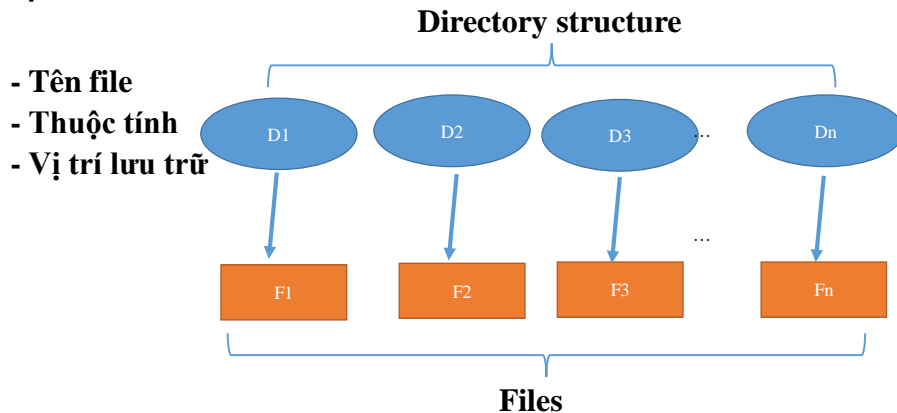
---

- Cấu trúc thư mục được ghi trên đĩa và gồm nhiều mục (directory entry), mỗi mục ghi thông tin một file (thuộc tính file, danh sách các số hiệu khối đĩa lưu trữ file)
  - Khi truy xuất file, HĐH tìm file trong bảng thư mục, lấy thông tin file lưu vào bộ nhớ (bảng open-file) dành cho các lần truy xuất sau
  - HĐH cung cấp các lời gọi hệ thống thao tác trên thư mục như: Create, Delete, Open, Close, Read, rename, Link, Unlink,...
- 

28

## Thư mục (tt)

- Mô hình cấu trúc thư mục, mỗi mục ( $D_i$ ) quản lý một file ( $F_i$ ) hoặc thư mục con

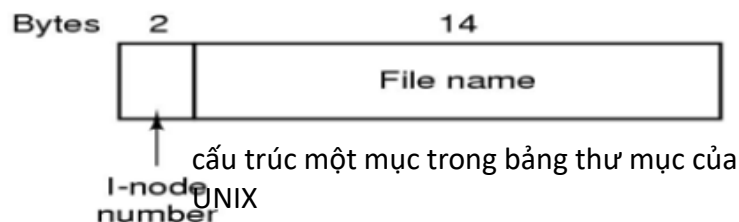


29

## Thư mục (tt)

### • Bảng thư mục (Directory table)

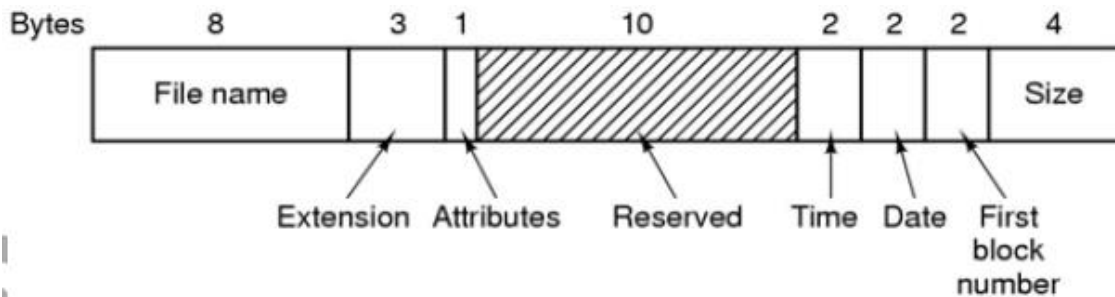
- Mảng một chiều, mỗi phần tử (directory entry) lưu thông tin của một file/thư mục:
  - Thuộc tính file
  - Địa chỉ trên đĩa của file (số hiệu khối đầu tiên chứa file/số I-node của file)
- Mỗi đĩa có một bảng thư mục gọi là bảng thư mục gốc (**RDET** – Root Directory Entry Table), cài đặt ở phần đầu của đĩa và có thể có nhiều bảng thư mục con (**SDET** – Sub Directory Entry Table)



30

## Thư mục (tt)

### •Bảng thư mục (tt)



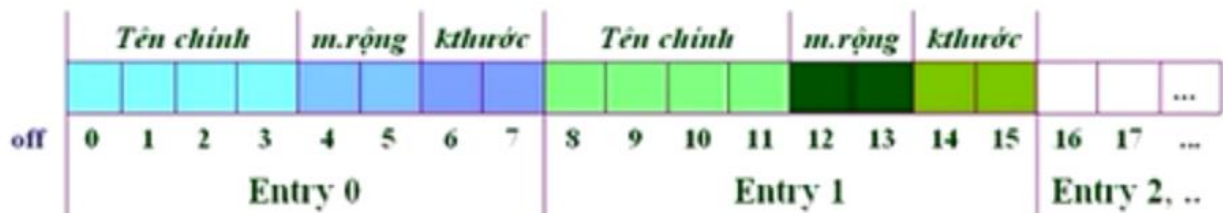
cấu trúc một mục trong bảng thư mục của MSDOS/WINDOWS (FAT)

31

## Thư mục (tt)

•Ví dụ: nếu mô hình thuộc tính file được thiết kế chỉ gồm các thành phần:

- tên file ( $\leq 4$  kí tự)
- phần mở rộng ( $\leq 2$  kí tự)
- kích thước (số nguyên 2 byte) → bảng RDET:



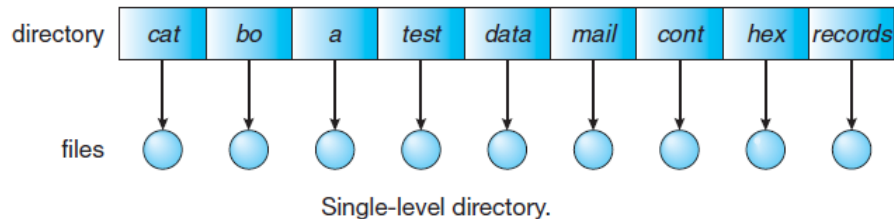
32



## Thư mục (tt)

### • Cấu trúc thư mục dạng đơn cấp

- Tất cả file được chứa trong cùng thư mục
- Cấu trúc thư mục chứa của tất cả file của tất cả người dùng



### • Hạn chế:

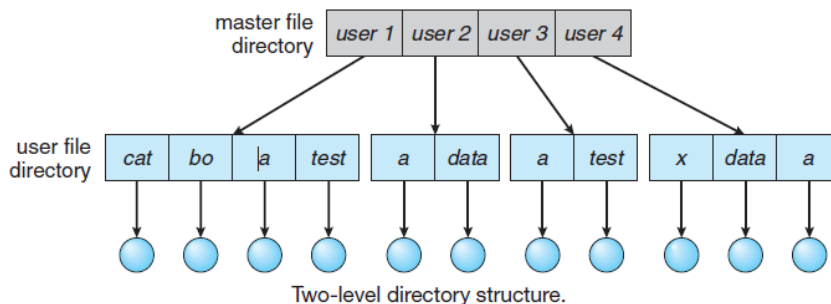
- Thư mục chứa số lượng lớn file → khó nhớ, tìm kiếm chậm
- Nhiều người dùng → lẫn lộn tên tập tin

33

## Thư mục (tt)

### • Cấu trúc thư mục dạng hai cấp

- Mỗi người dùng có thư mục riêng (user file directory-UFD)
- Mỗi UFD có cấu trúc giống nhau nhưng chỉ quản lý file của một người dùng
- Khi người dùng đăng nhập, hệ thống MFD (master file directory) tìm kiếm load UFD dựa vào id tài khoản người dùng



34

## Thư mục (tt)

### •Cấu trúc thư mục dạng hai cấp (tt)

- Để tạo một file cho một người dùng, HĐH chỉ tìm UFD của người dùng đó để xác định một file khác cùng tên có tồn tại hay không.
- Để xóa một tập tin, HĐH chỉ tìm kiếm trong UFD cục bộ → không thể xóa nhầm file của người dùng khác có cùng tên.
- Ưu điểm:
  - Tên file của các người dùng khác nhau có thể trùng
  - Tìm kiếm nhanh
- Hạn chế: các người dùng khó có thể truy xuất các file của người dùng khác

35

## Thư mục (tt)

### •Cấu trúc thư mục đa cấp (dạng cây)

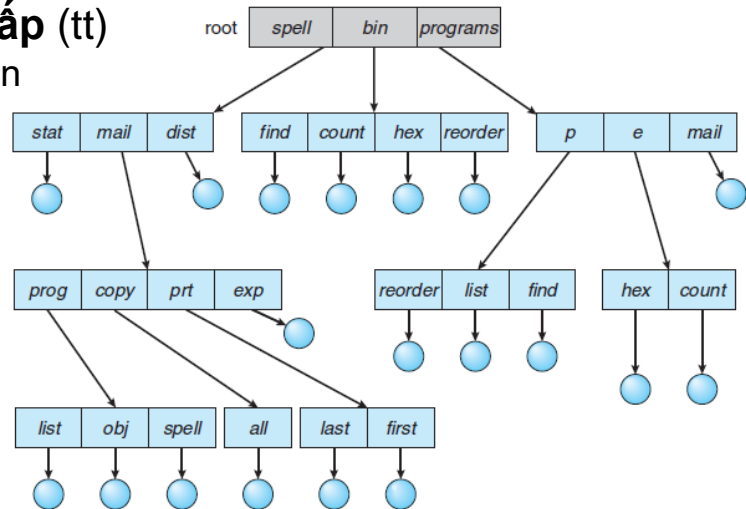
- Là trường hợp tổng quát của thư mục hai cấp
- Cho phép người dùng tạo thư mục con và tổ chức các tập tin
- Một thư mục (hay thư mục con) chứa tập hợp các file hay thư mục con
- Mỗi người dùng có **thư mục hiện hành**
- Tên đường dẫn có hai kiểu: **tên đường dẫn tuyệt đối** và **tên đường dẫn tương đối**
- Người dùng có thể truy xuất tới các file của người dùng khác bằng cách xác định tên đường dẫn của chúng (đường dẫn tương đối hay tuyệt đối).

36

## Thư mục (tt)

### • Cấu trúc thư mục đa cấp (tt)

- Người dùng có thể chuyển thư mục hiện hành tới thư mục của người dùng khác và truy xuất các file bằng tên của chúng



Tree-structured directory structure.

37

## Thư mục (tt)

### • Cấu trúc Acyclic-Graph

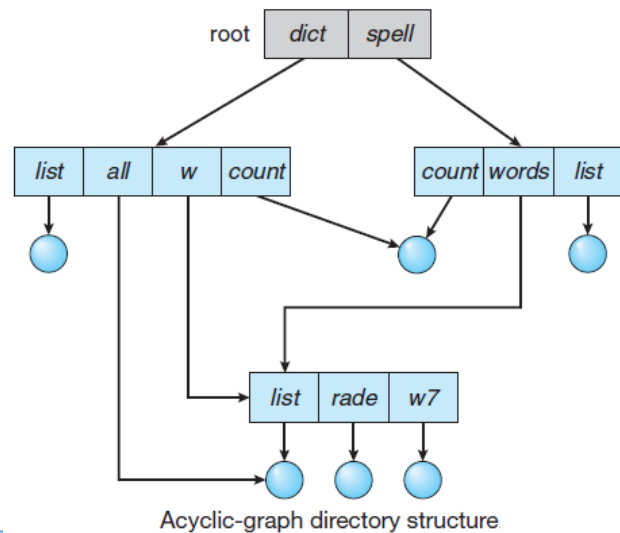
- Cho phép chia sẻ thư mục chung cho các user
- UFD của các user được chia sẻ sẽ chứa thư mục dùng chung
- Cách chia sẻ file/thư mục:
  - Symbolic link: một con trỏ đến file hoặc thư mục khác
  - Hard link: nhân bản thông tin trong thư mục chia sẻ

38

## Thư mục (tt)

### • Cấu trúc Acyclic-Graph (tt)

- Linh hoạt nhưng phức tạp: một file chia sẻ có thể có nhiều
- Vấn đề trong thao tác xóa.  
VD: xóa file /dict/w/list



39

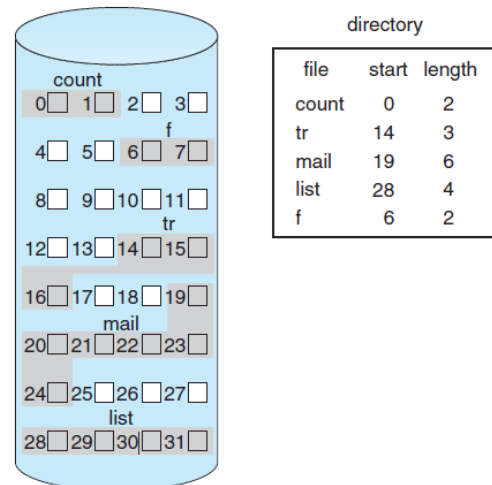
## 5.2. Hiện thực Hệ thống file

- Các phương pháp cấp phát không gian cho file
  - Cấp phát liên tục
  - Cấp phát theo danh sách liên kết
  - FAT
  - Cấp phát dùng chỉ số index
  - I-nodes
  - NTFS
- Quản lý các khối trống
- Hệ thống file trong MS-DOS

40

## 5.2.1 Cấp phát liên tục

- Lưu trữ file trên dãy các khối liên tiếp.
- Mỗi phần tử trong bảng thư mục:
  - Tên file
  - Vị trí khối (cluster) đầu tiên.
  - Kích thước (số khối)
- Ưu điểm: Đơn giản, dễ cài đặt, dễ thao tác vì toàn bộ file được đọc từ đĩa bằng thao tác đơn giản không cần định vị lại.

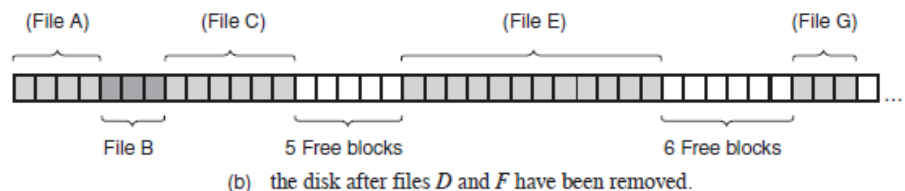
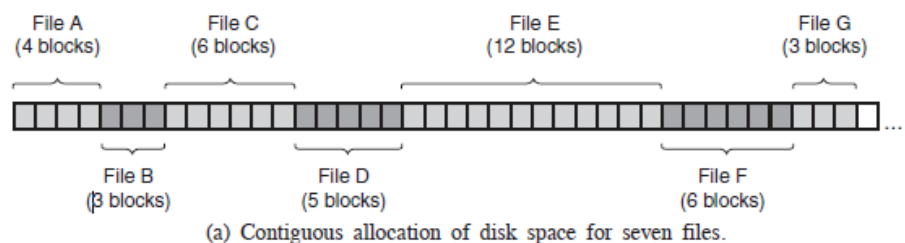


Contiguous allocation of disk space.

41

## Cấp phát liên tục (tt)

- Hạn chế:
  - Không linh động.
  - Phân mảnh ngoại vi



42

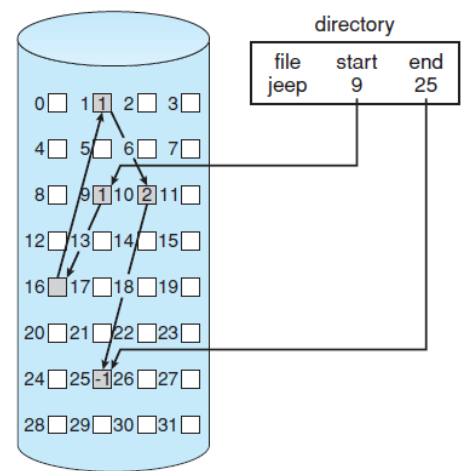
## Cấp phát liên tục (tt)

- Bài tập:
  - Cluster có kích thước 1K
  - Các file cần lưu trữ gồm:
    - baitap (0,2006): bắt đầu:cluster 0, kích thước: 2006
    - baigiang (6,2007)
    - lythuyet (14, 2050)
    - thuchanh (19, 6000)
    - diem (28,4096)
  - Cho biết cách phân bổ các cluster để có thể lưu trữ các file trên?

43

## 5.2.2 Cấp phát dùng danh sách liên kết

- Các khối của file có thể nằm ở các vị trí không liên tục nhau
- Mỗi mục trong bảng thư mục:
  - Tên file
  - Vị trí khối đầu tiên (start)
  - Vị trí khối kết thúc (end)
- Mỗi khối trên đĩa:
  - Con trỏ (~ 4 bytes) trở đến vị trí khối tiếp theo của file
  - Dữ liệu của file



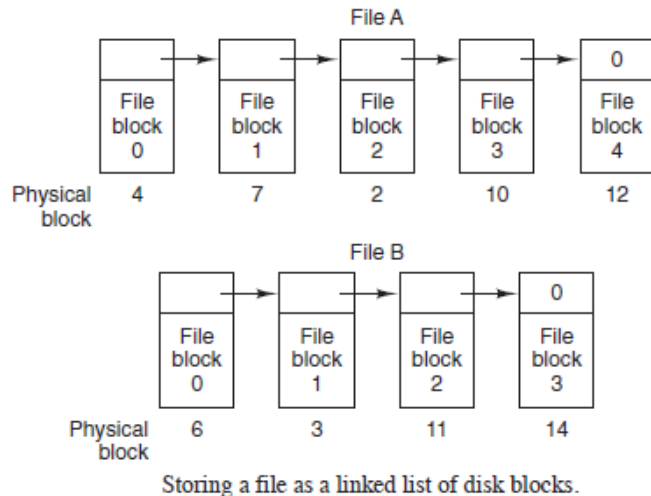
Linked allocation of disk space.

44

## Cấp phát dùng danh sách liên kết

### •Nhận xét:

- Không phân mảnh ngoại vi
- Truy xuất ngẫu nhiên chậm
- Tốn bộ nhớ lưu các con trỏ
- Một con trỏ lỗi → cả file

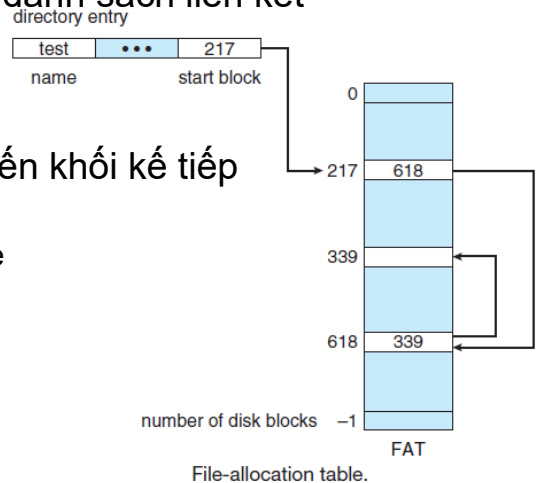


45

## 5.2.3 FAT (File Allocation Table)

### •Mô hình cấp phát không liên tục, sử dụng FAT:

- Giải quyết hạn chế của cấp phát dùng danh sách liên kết
- Mỗi mục trong bảng thư mục lưu số hiệu của khối đầu tiên
- Các khối còn lại của file lưu trong bảng FAT, mỗi phần tử có con trỏ trỏ đến khối kế tiếp
- Ưu điểm:
  - Dễ quản lý các số hiệu khối đã cấp cho file
  - Truy xuất ngẫu nhiên dễ dàng
  - Kích thước file dễ mở rộng
- Hạn chế:
  - Bảng FAT có kích thước giới hạn



46

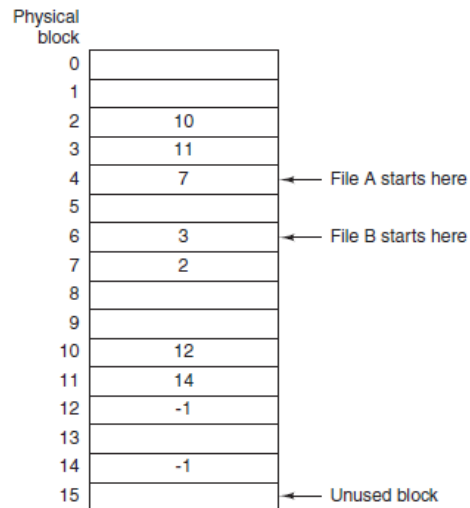
## FAT (tt)

- Ví dụ 2: file A và B được cấp phát các khối như sau:

- A: 4, 7, 2, 10, 12
- B: 6, 3, 11, 14

Bảng thư mục

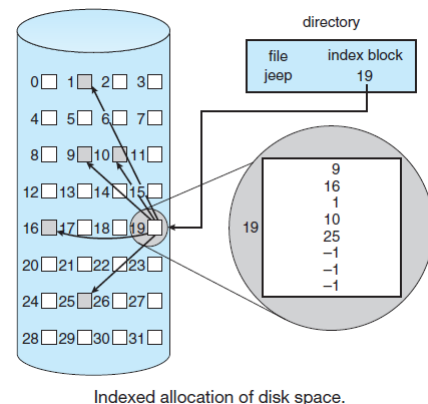
...	(A, 4)	...	(B, 6)	...
-----	--------	-----	--------	-----



47

## 5.2.4 Cấp phát dùng index (bảng chỉ mục)

- Sử dụng bảng chỉ mục (một mảng) chứa tất cả vị trí các khối của file
- Một mục trong bảng thư mục:
  - Tên file
  - Vị trí khối của bảng chỉ mục của file (phần tử i quản lý cluster i)
- Ưu điểm
  - Hỗ trợ truy cập trực tiếp → Truy cập ngẫu nhiên tốt
  - Không phân mảnh ngoại vi
- Nhược điểm
  - Tốn không gian lưu trữ bảng index khi file có kích thước chỉ vài block
  - Nếu khối lưu bảng index bị bad → mất cả tập tin



48



## 5.2.5 Bảng I-nodes

- Sử dụng trong UNIX
- Mỗi file được quản lý bằng một cấu trúc gọi là I-nodes

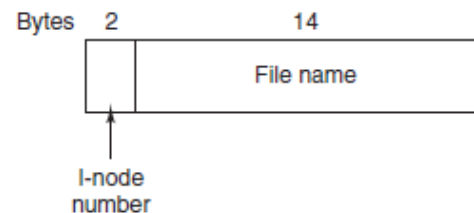
- Mỗi I-nodes bao gồm:

- Phần lưu trữ thuộc tính file

- Phần thứ 2 gồm

- LINUX: 12 (UNIX v7:10) con trỏ đầu trỏ đến các khối dữ liệu đầu tiên của file

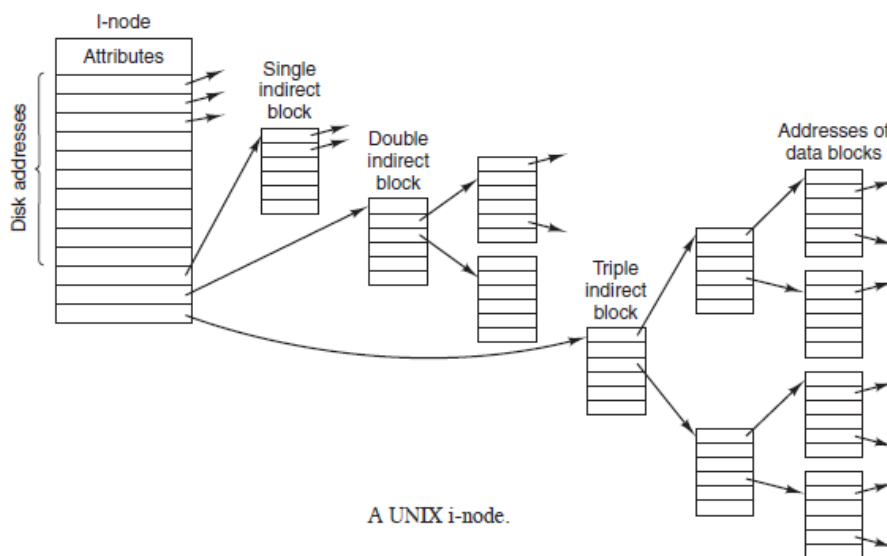
- Ba phần tử còn lại trỏ đến các bảng thứ cấp: single, double, w triple



A UNIX V7 directory entry.

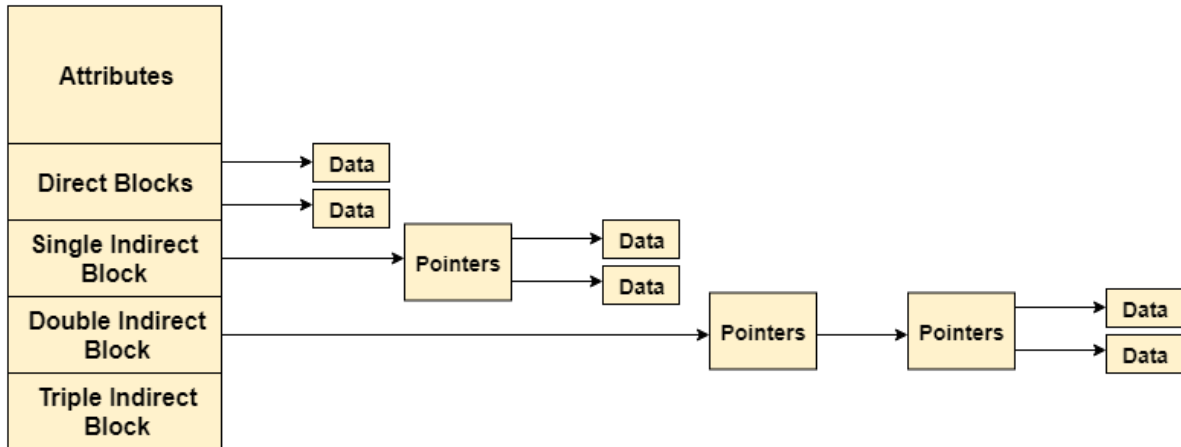
49

## Bảng I-nodes (tt)



50

## Bảng I-nodes (tt)



51

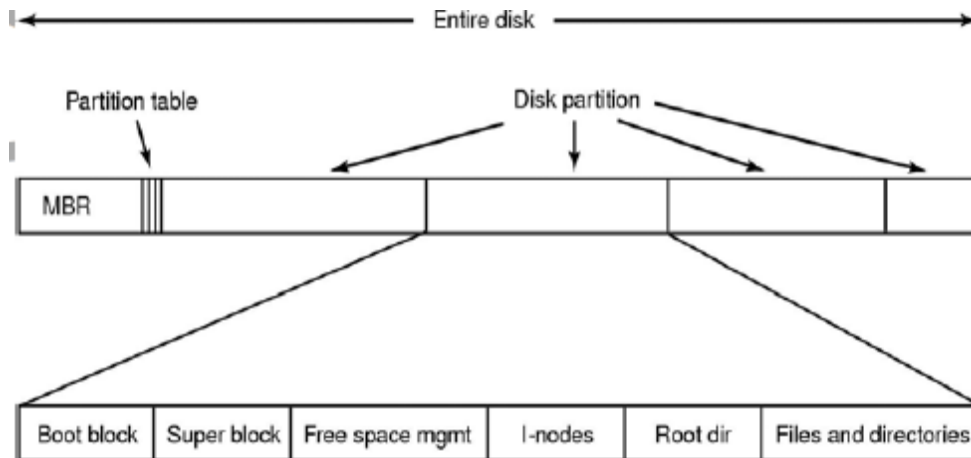
## Bảng I-nodes (tt)

- Tổ chức quản lý đĩa bằng I-nodes:
  - MBR (Master Boot Record): sector đầu tiên chứa thông tin về đĩa
  - Partition Table: bảng phân vùng chứa các thông tin về mỗi phân vùng
  - Tổ chức một phân vùng gồm: boot block, supper block, free space mgmt, I-nodes, root dir, files, directories
    - I-nodes: bảng chứa các i-node ghi thông tin mỗi file
    - Mỗi mục của bảng thư mục gốc (root-dir) ghi tên file và số hiệu i-node của file
  - Ưu điểm: quản lý hiệu quả khi hệ thống file lớn

52

## Bảng I-nodes (tt)

- Tổ chức quản lý đĩa bằng I-nodes:



53

## Bảng I-nodes (tt)

- Ví dụ: đọc file /usr/ast/mbox

Thư mục gốc	I-node 6 dành cho /user	Block 132 là thư mục /usr	I-node 26 dành cho /usr/ast	Block 406 là thư mục /usr/ast																																																		
<table><tr><td>1</td><td>.</td></tr><tr><td>1</td><td>..</td></tr><tr><td>4</td><td>bin</td></tr><tr><td>7</td><td>dev</td></tr><tr><td>14</td><td>lib</td></tr><tr><td>9</td><td>etc</td></tr><tr><td>6</td><td>usr</td></tr><tr><td>8</td><td>tmp</td></tr></table> <p>Tra cứu usr tìm được i-node 6</p>	1	.	1	..	4	bin	7	dev	14	lib	9	etc	6	usr	8	tmp	<table><tr><td>Mode size times</td></tr><tr><td>132</td></tr><tr><td></td></tr></table> <p>I-node 6 cho biết thông tin /user được lưu tại block 132</p>	Mode size times	132		<table><tr><td>6</td><td>.</td></tr><tr><td>1</td><td>..</td></tr><tr><td>19</td><td>dick</td></tr><tr><td>30</td><td>erik</td></tr><tr><td>51</td><td>jim</td></tr><tr><td>26</td><td>ast</td></tr><tr><td>45</td><td>bal</td></tr></table> <p>/usr/ast là i-node 26</p>	6	.	1	..	19	dick	30	erik	51	jim	26	ast	45	bal	<table><tr><td>Mode size times</td></tr><tr><td>406</td></tr><tr><td></td></tr></table> <p>I-node cho thấy /usr/ast ở tại block 406</p>	Mode size times	406		<table><tr><td>26</td><td>.</td></tr><tr><td>6</td><td>..</td></tr><tr><td>64</td><td>grants</td></tr><tr><td>92</td><td>books</td></tr><tr><td>60</td><td>mbox</td></tr><tr><td>81</td><td>minix</td></tr><tr><td>17</td><td>src</td></tr></table> <p>/usr/ast/mbox là i-node 60</p>	26	.	6	..	64	grants	92	books	60	mbox	81	minix	17	src
1	.																																																					
1	..																																																					
4	bin																																																					
7	dev																																																					
14	lib																																																					
9	etc																																																					
6	usr																																																					
8	tmp																																																					
Mode size times																																																						
132																																																						
6	.																																																					
1	..																																																					
19	dick																																																					
30	erik																																																					
51	jim																																																					
26	ast																																																					
45	bal																																																					
Mode size times																																																						
406																																																						
26	.																																																					
6	..																																																					
64	grants																																																					
92	books																																																					
60	mbox																																																					
81	minix																																																					
17	src																																																					

54

## 5.2.6 NTFS (New Technology File System)

### • Đặc điểm của NTFS

- NTFS là một hệ thống file mạnh và linh động, những đặc điểm nổi bật là :
  - Khả năng phục hồi
  - An toàn
  - Quản lý được đĩa dung lượng lớn và kích thước file lớn
  - Quản lý hiệu quả.
  - Nén, mã hóa
  - Hard link, symbolic link

55

## NTFS (tt)

### • Cấu trúc volume của NTFS

Partition Boot Sector	Master File Table	Các tập tin hệ thống	Vùng các tập tin
-----------------------	-------------------	----------------------	------------------

- Partition Boot Sector: các sector khởi động của partition (có thể đến 16 sectors) bao gồm các thông tin về cấu trúc của volume, cấu trúc của hệ thống file cũng như những thông tin và mã nguồn khởi động.
- Các file hệ thống: có kích khoảng 1Mb bao gồm:
  - MFT2 : bản sao của MFT
  - Log file : thông tin về các giao tác dùng cho việc phục hồi.
  - Cluster bitmap : biểu diễn thông tin lưu trữ của các cluster
  - Bảng định nghĩa thuộc tính : định nghĩa các kiểu thuộc tính hỗ trợ cho volume đó
- Vùng các tập tin: chứa nội dung thực sự của các tập tin.

56

## NTFS (tt)

### • Cấu trúc volume của NTFS (tt)

#### • Master File Table (MFT):

- Lưu các thông tin về tất cả file và thư mục trên volume NTFS, kể cả các vùng trống.
- MFT được tổ chức thành nhiều dòng.
  - Mỗi dòng (1024 bytes) mô tả cho một file hoặc một thư mục trên volume.
  - Nếu kích thước file nhỏ thì toàn bộ nội dung của file được lưu trong dòng này.
  - Mỗi dòng cũng lưu những thuộc tính cho file hay thư mục mà nó quản lý.

57

## NTFS (tt)

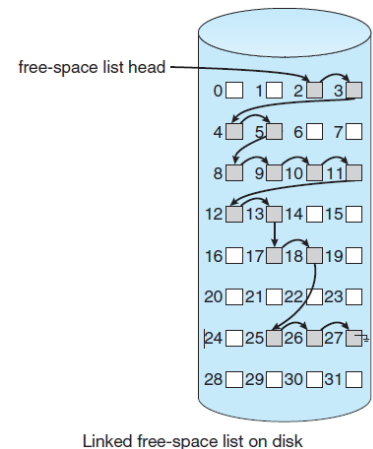
#### • Master File Table (tt):

Kiểu thuộc tính	Mô tả
Thông tin chuẩn	Thuộc tính truy xuất, nhãn thời gian, chỉ số liên kết
Danh sách thuộc tính	Trường hợp tất cả thuộc tính vượt quá một dòng của MFT
Tên tập tin	
Mô tả an toàn	Thông tin về người sở hữu và truy cập
Dữ liệu	
Chỉ mục gốc	Dùng cho thư mục
Chỉ mục định vị	
Thông tin volume	Version, tên volume
Bitmap	Hiện trạng các dòng trong MFT

58

## 5.2.7 Quản lý các khối trống trên đĩa

- Dừng danh sách liên kết: mỗi phần tử trong DSLK là một khối chứa một bảng gồm các số hiệu khối trống, phần tử cuối của bảng lưu số hiệu khối tiếp theo trong danh sách
- Nhận xét:
  - đĩa hoàn toàn trống: tốn nhiều khối nhớ cho DSLK
  - đĩa gần đầy: tốn ít khối nhớ cho DSLK

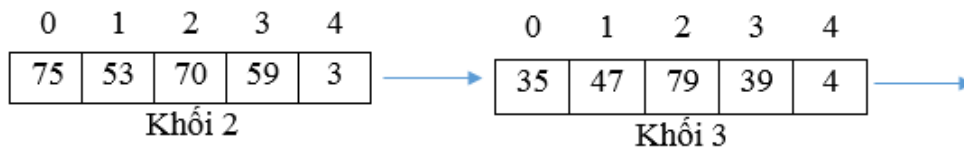


59

## Quản lý các khối trống trên đĩa (tt)

- Dừng danh sách liên kết (tt)

• Ví dụ:



- Khối 2, 3 là là các khối lưu số hiệu các khối trống
- Khối 2 là khối đầu tiên trong DSLK

60

## Quản lý các khối trống trên đĩa (tt)

- Dùng vector bit (dãy bit)
  - bit thứ  $i = 1$ : khối thứ  $i$  trống
  - bit thứ  $i = 0$ : khối thứ  $i$  có dữ liệu
  - Vector bit được lưu trên một hoặc nhiều khối đĩa, khi cần sẽ được đọc vào bộ nhớ để xử lý
  - Ưu điểm: Ít tốn khối nhớ so với DSLK
  - Hạn chế:
    - kích thước vector là cố định
    - HĐH cần đồng bộ vector bit trong bộ nhớ và vector bit trên đĩa

1001101101101100
0110110111110111
1010110110110110
0110110110111011
1110111011101111
1101101010001111
0000111011010111
1011101101101111
1100100011101111
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
0111011101110111
1101111101110111

A bitmap

61

## 5.2.8 Hệ thống file trong MSDOS

- Nội dung một file được chia thành các block file lưu trữ tại các cluster trên đĩa, các cluster này có thể không nằm kề nhau.
- Bảng FAT:
  - Quản lý danh sách các cluster chứa nội dung của một file của tất cả các file
  - Ghi nhận trạng thái của các cluster trên đĩa: còn trống, đã cấp phát cho các file, bị bad không thể sử dụng hay dành riêng cho hệ điều hành.
- Trong quá trình khởi động máy tính HĐH nạp bảng FAT vào bộ nhớ để chuẩn bị cho việc đọc/ghi các file sau này.

62

## Hệ thống file trong MSDOS (tt)

- Khi cần đọc/ghi file, HĐH phải dựa vào bảng FAT
- Bảng FAT hỏng → không thể ghi/đọc các file trên đĩa
- → HĐH DOS tạo ra hai bảng FAT hoàn toàn giống nhau là FAT1 và FAT2 (dự phòng)
- Nếu FAT1 bị hỏng thì DOS sẽ sử dụng FAT2 để khôi phục lại FAT1.
- **Trong hệ thống file FAT32:** nếu FAT1 bị hỏng thì HĐH sẽ chuyển sang sử dụng FAT2, sau đó mới khôi phục FAT1, và ngược lại.

63

## Hệ thống file trong MSDOS (tt)

- Bảng FAT bao gồm nhiều phần tử, các phần tử được đánh địa chỉ bắt đầu từ 0
- Mỗi phần tử trong bảng FAT cho biết trạng thái của một cluster tương ứng trên vùng dữ liệu.
- HĐH DOS định dạng hệ thống file theo một trong 2 loại FAT là FAT12 và FAT16.
  - Mỗi phần tử trong FAT12 rộng 12 bit (1.5 byte)
  - Mỗi phần tử trong FAT16 rộng 16 bit (2 byte).

64



## Hệ thống file trong MSDOS (tt)

- Bảng giá trị các phần tử trong FAT
  - **000h** (0000h): cluster tương ứng còn để trống.
  - **FF7h** (FFF7h): cluster tương ứng bị bad.
  - **FF0h** (FFF0h) - **FF6h** (FFF6h): cluster tương ứng dành riêng cho hệ điều hành.
  - **FF8h** (FFF8h) - **FFFh** (FFFFh): cluster tương ứng là cluster cuối cùng trong dãy các cluster chứa nội dung của một file.
  - **002h** (0002h) – **FFEh** (FFFEh): số hiệu của cluster trong bảng FAT, cho biết cluster tiếp theo trong dãy các cluster chứa nội dung của một file.

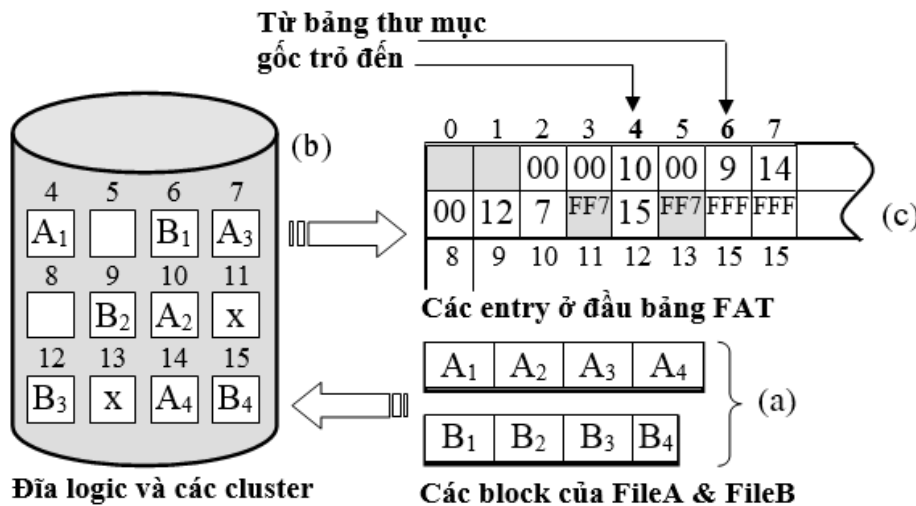
65

## Hệ thống file trong MSDOS (tt)

- Trong bảng FAT, hai phần tử đầu tiên (00 và 01) được sử dụng để chứa một giá trị nhận biết khuôn dạng đĩa, được gọi là byte định danh (byte ID) của đĩa, đây là byte đầu tiên của bảng FAT.
  - Đối với đĩa cứng thì byte ID = F8h.
- Để đọc được nội dung của một file trên đĩa, HĐH phải tìm được dãy các cluster chứa nội dung của một file.
  - Cluster đầu tiên được chứa trong bảng thư mục gốc
  - Các cluster còn lại được chứa trong bảng FAT

66

## Hệ thống file trong MSDOS (tt)



67

## Hệ thống file trong MSDOS (tt)

### • Nhận xét:

- Thao tác đọc file của DOS kém hiệu quả
  - đọc nội dung của file tại các cluster trên vùng data của đĩa
  - đọc và phân tích bảng FAT để các cluster chứa nội dung của một file.
- Khắc phục:
  - lưu danh sách các cluster chứa nội dung của một file vào một danh sách
  - khi đọc file hệ điều hành chỉ cần đọc nội dung của các cluster trên đĩa theo danh sách
  - Lưu danh sách các cluster còn trống trên đĩa → thì tốc độ ghi file nhanh
  - Windows98, windowsNT/2000

68

## Hệ thống file trong MSDOS (tt)

### •Root Directory:

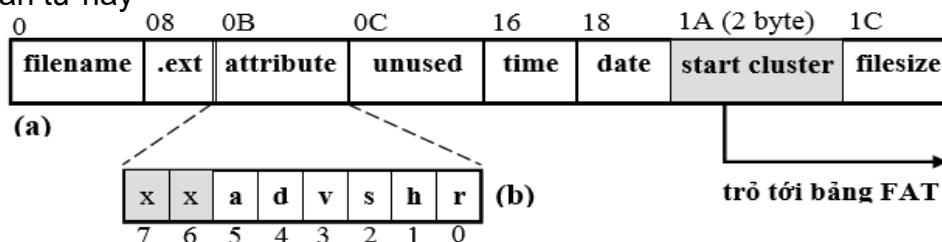
- Quản lý thông tin của các file và các thư mục đang được lưu trữ trên thư mục gốc của đĩa mềm hoặc đĩa logic trên đĩa cứng
- Số lượng phần tử trong bảng thư mục gốc được DOS quy định trước trong quá trình Format đĩa và được ghi tại word tại offset 11h trong boot sector, giá trị này không thể thay đổi.
- → tổng số file và thư mục con chứa trên thư mục gốc của đĩa là có giới hạn. Đây là một hạn chế của DOS.

69

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

- Mỗi phần tử trong bảng thư mục gốc có kích thước 32 byte, dùng để chứa thông tin về một file/thư mục đang lưu trên thư mục gốc của đĩa.
  - Khi một file/thư mục được tạo ra trên thư mục gốc của đĩa thì HĐH dùng một phần tử trong bảng thư mục gốc để chứa các thông tin liên quan của nó
  - Khi một file/thư mục bị xoá/di chuyển khỏi thư mục gốc thì HĐH sẽ thu hồi lại phần tử này



Cấu trúc của một phần tử trong bảng thư mục gốc của DOS

70

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

- Byte đầu tiên (offset 00) của một phần tử trong thư mục gốc (byte trạng thái) chứa một trong các giá trị đặc biệt sau đây:
  - 0: cho biết phần tử chưa được sử dụng.
  - E5h: phần tử là của một file đã được tạo nhưng đã bị xóa.
  - 05h: kí tự đầu tiên của tên file này thực tế là E5h, nếu file được tạo có tên bắt đầu là kí tự có mã ascii là E5h thì sẽ được thay bằng kí tự có mã là 05h, để phân biệt file này với các file đã bị xoá.
  - 2Eh (kí tự '.'): phần tử chứa thông tin của một thư mục con, nếu byte thứ 2 cũng là 2Eh ( "..") thì trường start cluster sẽ chứa số hiệu cluster đầu tiên của thư mục cha, nếu là thư mục gốc thì là 0000h.

71

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

- Các trường trong một phần tử của bảng thư mục gốc

Offset	Nội dung	Độ lớn
00h	Tên chính của file (ASCII)	8 byte
08h	Phần mở rộng của tên File (ASCII)	3 byte
0Bh	Thuộc tính file (attribute) (0, 0, A, D, V, S, H, R)	1 byte
0Ch	Dự trữ, chưa được sử dụng (Unused)	10 byte
16h	Giờ thay đổi tập tin cuối cùng (giờ 5bit, phút 6bit, giây/2 5bit)	2 byte
18h	Ngày thay đổi tập tin cuối cùng (ngày 5 bit, tháng 4 bit năm -1980: 7 bit)	2 byte
1Ah	Cluster đầu tiên của File (start cluster)	2 byte
1Ch	Kích thước của File (filesize)	4 byte

72

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

- Ví dụ: một phần tử lưu tập tin có dãy byte như sau:

44 45 54 48 49 20 20 20 44 4F 43 20 00 00 00 00  
00 00 00 00 00 00 16 4A BB 34 14 00 1E 0A 00 00

- → các thuộc tính tập tin là:

- Tên file: DETHI.DOC
- Kích thước: offset 1Ch: 0A1E = 2590
- Cluster bắt đầu: offset 1Ah: 14h = 20
- Ngày cập nhật: offset 18h: 34BBh = 0011010 0101 11011b = 27/5/2006
- Giờ cập nhật: offset 16h: 4A16h = 01001 010000 10110h = 9:16:44

73

73

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

- Bài tập: giả sử một phần tử lưu tập tin có dãy byte như sau:

42	61	69	74	61	70	20	20	63	70	70	20	00	00	00	00
00	00	00	00	00	00	F9	41	52	47	AA	01	DE	09	00	00

- Xác định các thuộc tính tập tin

74

74

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

#### •Trường hợp xóa file:

- HÐH thay kí tự đầu tiên của tên file tại phần tử trong bảng thư mục gốc bằng giá trị E5h
- → có thể khôi phục lại được file này, bằng các thay kí tự mã E5h ở byte đầu tiên của tên file bằng một kí tự khác
- Khi duyệt bảng thư mục gốc gặp các phần tử có byte đầu = E5h HÐH biết đây là phần tử của file đã bị xóa nên không in ra màn hình.
- Trường hợp không khôi phục được: do sau một thời gian, HÐH đã sử dụng phần tử trong thư mục gốc, các phần tử trong bảng FAT và các cluster trên vùng data của file đã bị xóa, cấp phát cho các file mới.

75

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

#### •Trường hợp ghi file: HÐH DOS thực hiện các bước sau:

1. Tìm phần tử mà byte đầu tiên của nó chứa giá trị 00. **Giả sử tìm được phần tử thứ 105.**
2. Ghi tên file, phần mở rộng, thuộc tính của file, ngày giờ tạo file vào các trường tương ứng tại **phần tử 105** trong bảng thư mục gốc.
3. Tìm một phần tử trong bảng FAT chứa giá trị 000h, giả sử tìm được **207 → cluster 207** trên vùng data còn trống.
4. Ghi giá trị **207**, vào trường **start cluster** tại offset **1Ah** của phần tử **105** trong bảng thư mục gốc.
5. Ghi block đầu tiên của file vào **cluster 207** trên vùng data. Nếu nội dung của file chứa vừa đủ trong 1 cluster, thì DOS sẽ thực hiện bước 9, ngược lại DOS tiếp tục thực hiện bước 6.

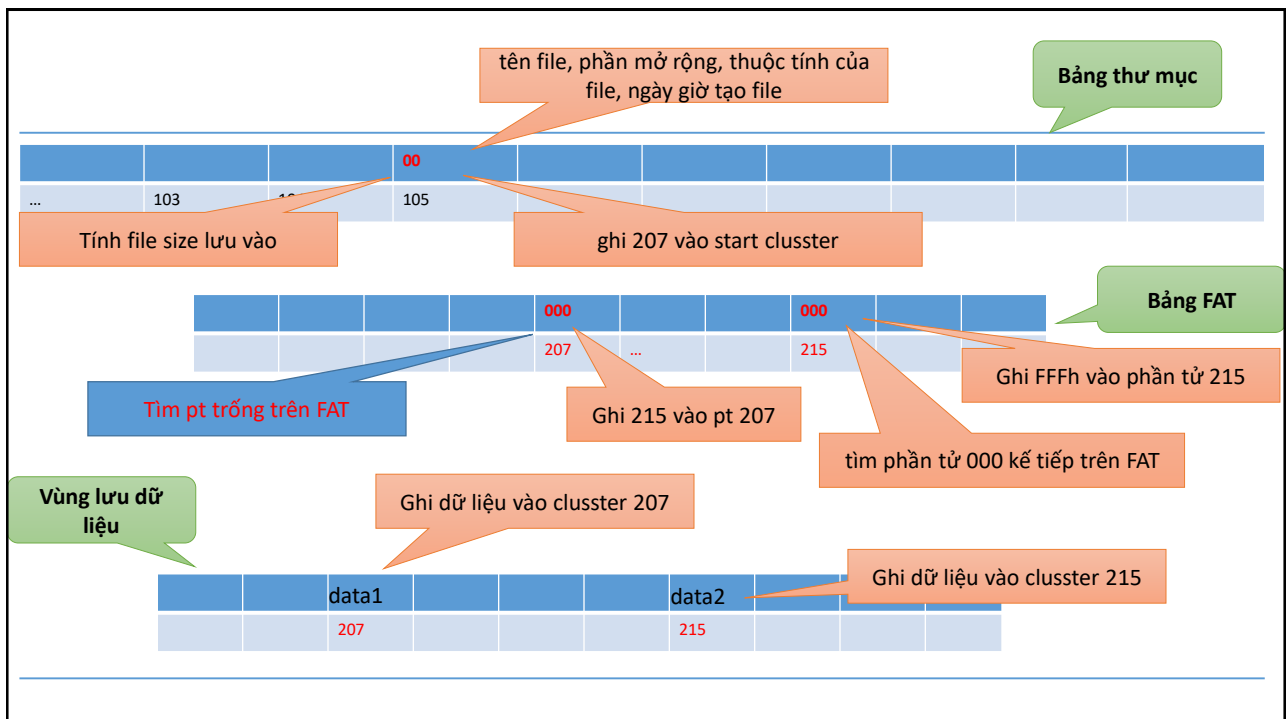
76

## Hệ thống file trong MSDOS (tt)

### •Root Directory (tt)

6. Tiếp tục tìm một phần tử trong bảng FAT chứa giá trị 000h, giả sử tìm được phần tử 215 → cluster 215 trên vùng data còn trống.
7. Ghi giá trị 215 vào phần tử 207 trong bảng FAT và ghi block thứ hai của file vào cluster 215 trên vùng data.
8. Lặp lại các bước 6 và 7 cho đến khi ghi hết các block của file vào các cluster trên vùng data. Giả sử block cuối cùng của file được ghi vào cluster **n**.
9. Bước cuối cùng: ghi giá trị FFFh vào phần tử **n** trong bảng FAT.
10. Tính kích thước của file và ghi vào trường filesize của phần tử 105 trong bảng thư mục gốc.

77



78

## Hệ thống file trong MSDOS (tt)

- Thư mục con (subdirectory):
  - Số phần tử trong bảng thư mục gốc là cố định, không thể mở rộng, được DOS quy định trong quá trình định dạng đĩa. → khái niệm thư mục con.
  - Cấu trúc của một phần tử trong bảng thư mục của thư mục con tương tự cấu trúc của một phần tử trong bảng thư mục gốc.
    - Đầu chứa thông tin của một thư mục (con) gồm:
      - trường Filename là tên của thư mục (con)
      - trường Attribute = 16 (00010000)
      - trường Start cluster = cluster đầu tiên của thư mục (con)
      - trường Filesize = 0
      - ...
    - Hai phần tử đầu tiên trong bảng thư mục của thư mục con chứa các giá trị đặc biệt.

79

## Hệ thống file trong MSDOS (tt)

- Thư mục con (tt)
  - Đặc điểm thư mục con trong DOS:
    - Được lưu trữ giống các file khác trên đĩa, muốn đọc được thư mục con HĐH phải dò trong bảng FAT.
    - Bảng thư mục của thư mục con
      - có số lượng phần tử không giới hạn
      - có thể định vị tại một vị trí bất kỳ trên vùng data của đĩa, nếu tạo quá nhiều thư mục con trên đĩa thì bảng thư mục của các thư mục con sẽ chiếm hết nhiều không gian đĩa trên vùng data.
      - Virus khó có thể tấn công bảng thư mục của thư mục con vì nó không cố định
    - Kích thước của thư mục con là kích thước của tất cả các file trong thư mục con, dựa vào bảng thư mục của thư mục con và bảng FAT

80



## Hệ thống file trong MSDOS (tt)

---

- Thư mục con (tt)
    - Tạo thư mục con: HĐH tạo bảng thư mục cho nó và khởi tạo 2 phần tử đầu tiên trong bảng thư mục này:
      - Phần tử thứ nhất:
        - byte đầu tiên của trường filename chứa mã ascii của kí tự dấu chấm (.)
        - phần tử này chỉ đến chính thư mục hiện hành.
        - Trường Start cluster cho biết cluster bắt đầu của thư mục này.
      - Phần tử thứ hai:
        - 2 byte đầu tiên của trường filename chứa 2 mã ascii của 2 kí tự dấu chấm (..)
        - phần tử này chỉ đến thư mục cha của nó.
        - Trường Start cluster cho biết cluster bắt đầu của thư mục cha của nó, nếu cha của nó là thư mục gốc thì trường này chứa giá trị 0.
-