



# HỆ ĐIỀU HÀNH

## BÀI 3: HỆ THỐNG TẬP TIN FAT

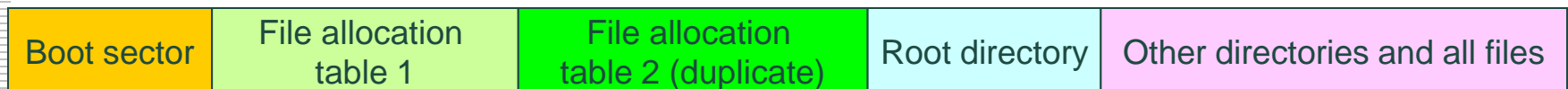
Ths. Lê Viết Long

# Nội dung

- ❖ Giới thiệu hệ thống tập tin FAT
- ❖ Vùng Boot Sector
- ❖ Bảng thư mục gốc (RDET)
- ❖ Bảng FAT
- ❖ Vùng dữ liệu
- ❖ Bảng thư mục con

# Giới thiệu hệ thống tập tin FAT

- ❖ FAT là hệ thống tập tin được sử dụng trên HĐH MS-DOS và Windows 9x (trên Windows họ NT có thêm hệ thống NTFS)
- ❖ Có 3 loại FAT
  - FAT12
  - FAT16
  - FAT32
- ❖ Tổ chức thành 2 vùng
  - Vùng hệ thống
    - Vùng Boot Sector
    - Bảng FAT
    - Bảng thư mục gốc (có thể nằm trên vùng dữ liệu)
  - Vùng dữ liệu



# Vùng Boot Sector

❖ Gồm một số sector đầu tiên của phân vùng (partition), trong đó:

- Sector đầu tiên (Boot Sector):
  - Chứa các thông số quan trọng của phân vùng
  - Chứa một đoạn chương trình nhỏ để nạp HĐH khi khởi động máy
- Các sector còn lại (nếu có):
  - Chứa các thông tin hỗ trợ cho việc xác định tổng số cluster trống & tìm kiếm cluster trống được hiệu quả
  - Chứa một sector bản sao của Boot sector

# Boot Sector của FAT12 và FAT16

Offset (hex)	Số byte	Ý nghĩa
0	3	Lệnh nhảy đến đầu đoạn mã Boot (qua khỏi vùng thông số)
3	8	Tên công ty /version của HĐH
B	2	Số byte của sector, thường là 512
D	1	Số sector của cluster ( $S_C$ )
E	2	Số sector trước bảng FAT ( $S_B$ )
10	1	Số lượng bảng FAT ( $N_F$ ), thường là 2
11	2	Số Entry của RDET ( $S_R$ ), thường là 512 với FAT16
13	2	Số sector của volume ( $S_V$ ), bằng 0 nếu $S_V > 65535$
15	1	Kí hiệu loại volume
16	2	Số sector của FAT ( $S_F$ )
18	2	Số sector của track
1A	2	Số lượng đầu đọc (side)
1C	4	Khoảng cách từ nơi mô tả vol đến đầu vol
20	4	Kích thước volume (nếu số 2 byte tại offset 13h là 0)
24	1	Ký hiệu vật lý của đĩa chứa vol (0 : mềm, 80h: cứng)
25	1	Dành riêng
26	1	Ký hiệu nhận diện HĐH
27	4	SerialNumber của Volume
2B	B	Volume Label
36	8	Loại FAT, là chuỗi "FAT12" hoặc "FAT16"
3E	1CF	Đoạn chương trình Boot nạp tiếp HĐH khi khởi động máy
1FE	2	Dấu hiệu kết thúc BootSector /Master Boot (luôn là AA55h)

# Boot Sector của FAT32

Offset	Số byte	Nội dung
0	3	Jump_Code: lệnh nhảy qua vùng thông số (như FAT)
3	8	OEM_ID: nơi sản xuất – version, thường là “MSWIN4.1”
B	2	Số byte trên Sector, thường là 512 (như FAT)
<b>D</b>	<b>1</b>	<b>S<sub>C</sub>: số sector trên cluster (như FAT)</b>
<b>E</b>	<b>2</b>	<b>S<sub>B</sub>: số sector thuộc vùng Bootsector (như FAT)</b>
<b>10</b>	<b>1</b>	<b>N<sub>F</sub>: số bảng FAT, thường là 2 (như FAT)</b>
11	2	Không dùng, thường là 0 (số entry của RDET – với FAT)
13	2	Không dùng, thường là 0 (số sector của vol – với FAT)
15	1	Loại thiết bị (F8h nếu là đĩa cứng - như FAT)
16	2	Không dùng, thường là 0 (số sector của bảng FAT – với FAT)
18	2	Số sector của track (như FAT)
1A	2	Số lượng đầu đọc (như FAT)
1C	4	Khoảng cách từ nơi mô tả vol đến đầu vol (như FAT)
<b>20</b>	<b>4</b>	<b>S<sub>V</sub>: Kích thước volume (như FAT)</b>
<b>24</b>	<b>4</b>	<b>S<sub>F</sub>: Kích thước mỗi bảng FAT</b>
28	2	bit 8 bật: chỉ ghi vào bảng FAT active (có chỉ số là 4 bit đầu)
2A	2	Version của FAT32 trên vol này
<b>2C</b>	<b>4</b>	<b>Cluster bắt đầu của RDET</b>
<b>30</b>	<b>2</b>	<b>Sector chứa thông tin phụ (về cluster trống), thường là 1</b>
<b>32</b>	<b>2</b>	<b>Sector chứa bản lưu của Boot Sector</b>
34	C	Dành riêng (cho các phiên bản sau)
40	1	Kí hiệu vật lý của đĩa chứa vol (0 : mềm, 80h: cứng)
41	1	Dành riêng
42	1	Kí hiệu nhận diện HĐH
43	4	SerialNumber của Volume
47	B	Volume Label
<b>52</b>	<b>8</b>	<b>Loại FAT, là chuỗi “FAT32”</b>
5A	1A4	Đoạn chương trình khởi tạo & nạp HĐH khi khởi động máy
1FE	2	Dấu hiệu kết thúc BootSector /Master Boot (luôn là AA55h)

# Bảng thư mục gốc (RDET – Root Directory Entry Table)

- ❖ Nằm trên vùng hệ thống (FAT12 & FAT16) hoặc nằm trên vùng dữ liệu (FAT32)
- ❖ Gồm một dãy các phần tử (gọi là entry), mỗi phần tử có kích thước 32 bytes chứa các thông tin của 1 tập tin hoặc một thư mục

Entry	1	2	...	16	17	18	...	32	33	...	208	209	210	...	224	225	226	...
Sector	1				2				...				14				...	

- ❖ Thông tin của mỗi tập tin/ thư mục có thể chiếm 1 hay nhiều entry
- ❖ Byte đầu tiên của mỗi entry cho biết trạng thái của entry này
  - 0 – entry trống
  - E5h – tập tin chiếm entry này đã bị xóa
  - Giá trị khác – đang chứa thông tin của tập tin/ thư mục
- ❖ Có 2 loại entry
  - Entry chính: chứa các thông tin của tập tin
  - Entry phụ: chỉ chứa tên của tập tin

# Cấu trúc bảng thư mục gốc

...	
Entry chính	} 32 bytes
Entry phụ N	
...	} 32 bytes
Entry phụ 2	
Entry phụ 1	
Entry chính	
Entry chính	
...	



# Entry chính

Offset (hex)	Số byte	Ý nghĩa
0	8	Tên chính /tên ngắn - lưu bằng mã ASCII
8	3	Tên mở rộng – mã ASCII
B	1	Thuộc tính trạng thái (0.0.A.D.V.S.H.R)
C	1	Dành riêng
D	3	Giờ tạo (miligiây:7; giây:6; phút:6; giờ:5)
10	2	Ngày tạo (ngày: 5; tháng: 4; năm-1980: 7)
12	2	Ngày truy cập gần nhất (lưu như trên)
14	2	Cluster bắt đầu – phần Word (2Byte) cao
16	2	Giờ sửa gần nhất (giây/2:5; phút:6; giờ:5)
18	2	Ngày cập nhật gần nhất (lưu như trên)
1A	2	Cluster bắt đầu – phần Word thấp
1C	4	Kích thước của phần nội dung tập tin

7	0	
6	0	
5	X	Archive
4	X	Directory
3	X	VolLabel
2	X	System
1	X	Hidden
0	X	ReadOnly

# Entry phụ

Offset	Số byte	Ý nghĩa
0	1	Thứ tự của entry (bắt đầu từ 1)
1	A (10d)	5 ký tự UniCode – bảng mã UTF16
B (11d)	1	Dấu hiệu nhận biết (luôn là <b>0Fh</b> )
E (14d)	C (12d)	6 ký tự kế tiếp
1C (28d)	4	2 ký tự kế tiếp

# Bảng FAT

- ❖ Nằm trên vùng hệ thống
- ❖ Thường có 2 bảng: 1 bảng chính và 1 bảng dự phòng
- ❖ Lưu vị trí của các tập tin/ thư mục theo kiểu danh sách liên kết

Giá trị	<b>X</b>	<b>X</b>	<b>3</b>	<b>4</b>	<b>EOF</b>	<b>7</b>	<b>EOF</b>	<b>6</b>
Phần tử	0	1	2	3	4	5	6	7

- ❖ Kích thước mỗi phần tử FAT phụ thuộc vào loại FAT
  - FAT12: kích thước mỗi phần tử là 12 bits ~ 1.5 bytes
  - FAT16: kích thước mỗi phần tử là 16 bits ~ 2 bytes
  - FAT32: kích thước mỗi phần tử là 32 bits ~ 4 bytes

# Bảng FAT (tt)

- ❖ Phần tử thứ k trên bảng FAT (đánh số từ 0) cho biết trạng thái của cluster thứ k trên vùng dữ liệu (đánh số từ 2) → 2 phần tử đầu của bảng FAT không dùng

Trạng thái của cluster k trên vùng dữ liệu	Giá trị của phần tử k trên bảng FAT			Ghi chú
	FAT12	FAT16	FAT32	
Trống	0	0	0	= FREE
Hư	FF7	FFF7	0FFFFFFF7	= BAD
Cluster cuối của file	FFF	FFFF	0FFFFFFF	= EOF
Chứa nội dung file	2 .. FEF	2 .. FFEF	2..0FFFFFFEF	

- ❖ FAT 12 quản lý được tối đa 4078 (FEEh) cluster
- ❖ FAT 16 quản lý được tối đa 65518 (FFEEh) cluster
- ❖ Nếu số cluster quá 65518 thì dùng FAT 32

# Làm sao để truy xuất các phần tử FAT ? FAT

## ❖ Lưu trữ bảng là dãy byte

Giá trị	<b>F0</b>	<b>FF</b>	<b>FF</b>	<b>03</b>	<b>40</b>	<b>00</b>	<b>FF</b>	<b>7F</b>	<b>FF</b>	<b>AB</b>	<b>CD</b>	<b>EF</b>
Byte	0	1	2	3	4	5	6	7	8	9	A	B

## ❖ Truy xuất theo FAT 32 (mỗi phần tử 4 bytes)

Giá trị	<b>F0</b>	<b>FF</b>	<b>FF</b>	<b>03</b>	<b>40</b>	<b>00</b>	<b>FF</b>	<b>7F</b>	<b>FF</b>	<b>AB</b>	<b>CD</b>	<b>EF</b>
Byte	0	1	2	3	4	5	6	7	8	9	A	B

Giá trị	<b>03 FF FF F0</b>				<b>7F FF 00 40</b>				<b>EF CD AB FF</b>			
Ptử FAT	0				1				2			

## ❖ Truy xuất theo FAT 16 (mỗi phần tử 2 bytes)

Giá trị	<b>F0</b>	<b>FF</b>	<b>FF</b>	<b>03</b>	<b>40</b>	<b>00</b>	<b>FF</b>	<b>7F</b>	<b>FF</b>	<b>AB</b>	<b>CD</b>	<b>EF</b>
Byte	0	1	2	3	4	5	6	7	8	9	A	B

Giá trị	<b>FFF0</b>	<b>03FF</b>	<b>0040</b>	<b>7FFF</b>	<b>ABFF</b>	<b>EFCD</b>
Ptử FAT	0	1	2	3	4	5

# Làm sao để truy xuất các phần tử FAT ? (tt)

## ❖ Truy xuất theo FAT 12 (mỗi phần tử 1.5 bytes)

Giá trị	F0	FF	FF	03	40	00	FF	7F	FF	AB	CD	EF
Byte	0	1	2	3	4	5	6	7	8	9	A	B

Giá trị	FF0	FFF	003	004	FFF	FF7	DAB	EFC
Ptử FAT	0	1	2	3	4	5	6	7

▪ Phần tử chẵn: F0 FF → FF0

▪ Phần tử lẻ: FF FF → FFF

## ❖ Công thức tương quan giữa phần tử thứ k và byte thứ i trên bảng FAT

$$i = k * \text{<kích thước phần tử FAT>}$$

# Vùng dữ liệu

- ❖ Mỗi phần tử trên vùng dữ liệu, gọi là **cluster**, có kích thước  $2^n$  sector, tùy thuộc vào người dùng khi format
- ❖ Cluster trên vùng dữ liệu đánh số từ 2
- ❖ Công thức tương quan giữa cluster thứ k trên vùng dữ liệu và sector thứ i trên phân vùng

$$i = S_B + S_F * N_F + [S_{RDET}] + (k - 2) * S_C$$

# Bảng thư mục con SDET – Sub Directory Entry Table

- ❖ Chứa thông tin các tập tin/ thư mục con của một thư mục
- ❖ Nằm trên vùng dữ liệu, có cấu trúc hoàn toàn giống bảng thư mục gốc
- ❖ Mỗi SDET luôn có 2 entry ‘.’ và ‘..’ ở đầu bảng mô tả về chính thư mục này và thư mục cha của nó



# Cấu trúc bảng thư mục con

.	
..	} 32 bytes
Entry chính	
Entry phụ N	} 32 bytes
...	
Entry phụ 2	
Entry phụ 1	
Entry chính	
Entry chính	
...	

# Một số thao tác trên hệ thống tập tin FAT

## ❖ Đọc nội dung tập tin (TYPE)

- Xác định entry chính trong bảng thư mục (RDET/ SDET) chứa thông tin của tập tin dựa vào phần tên và phần mở rộng (lưu ý trường hợp tên dài)
- Từ entry chính tìm được, ta có được chỉ số cluster/ phần tử FAT đầu tiên
- Từ phần tử FAT đầu tiên này, vào bảng FAT, xác định các phần tử còn lại của tập tin, tương ứng có được các cluster của tập tin này → các sector của tập tin
- Đọc các sector nội dung của tập tin

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Liệt kê nội dung thư mục (DIR)

- Xác định entry chính trong bảng thư mục (RDET/ SDET) chứa thông tin của thư mục dựa vào phần tên (lưu ý trường hợp tên dài)
- Từ entry chính tìm được, ta có được chỉ số cluster/ phần tử FAT đầu tiên
- Từ phần tử FAT đầu tiên này, vào bảng FAT, xác định các phần tử còn lại của tập tin, tương ứng có được các cluster của tập tin này  
→ các sector của tập tin
- Đọc các sector nội dung tìm được theo từng entry (32 bytes) và hiển thị thông tin của các tập tin và thư mục con của thư mục này

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Tạo tập tin (COPY CON)

- Tìm đủ số entry trống liên tiếp nhau trên bảng thư mục (RDET/ SDET) để chứa thông tin của tập tin (lưu ý trường hợp tên dài)
- Kiểm tra trên bảng FAT xem còn đủ số cluster trống để chứa nội dung của tập tin không
- Lưu thông tin của tập tin vào các entry trống tìm được
- Ghi giá trị vào các phần tử FAT trống tìm được theo dạng danh sách liên kết, đồng thời lưu nội dung tập tin vào các cluster tương ứng (theo chỉ số sector)

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Tạo thư mục (MD)

- Tìm đủ số entry trống liên tiếp nhau trên bảng thư mục (RDET/ SDET) để chứa thông tin của thư mục (lưu ý trường hợp tên dài)
- Kiểm tra trên bảng FAT xem còn cluster trống nào để chứa nội dung của thư mục không
- Lưu thông tin của thư mục vào các entry trống tìm được
- Ghi giá trị kết thúc vào phần tử FAT trống tìm được, đồng thời tạo 2 thư mục "." và ".." chiếm 2 entry đầu tiên trong cluster tương ứng

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Xóa tập tin (DELETE)

- Xác định entry chính trong bảng thư mục (RDET/ SDET) chứa thông tin của tập tin dựa vào phần tên và phần mở rộng (lưu ý trường hợp tên dài)
- Đặt giá trị E5h vào byte đầu tiên của entry chính và tất cả các entry phụ của tập tin (nếu có)
- Từ entry chính tìm được, ta có được chỉ số cluster/ phần tử FAT đầu tiên. Vào bảng FAT, xác định được các phần tử còn lại của tập tin
- Đặt tất cả các phần tử FAT của tập tin về giá trị 0
- Lưu ý, hoàn toàn không thay đổi gì phần nội dung của tập tin

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Xóa thư mục (RD)

- Thực hiện xóa đệ qui tất cả các tập tin và thư mục con từ cấp sâu nhất ra. Xóa thư mục rỗng tương tự như xóa tập tin

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Sao chép tập tin (COPY)

- Tìm đủ số entry trống liên tiếp nhau trên bảng thư mục (RDET/ SDET) để chứa thông tin của tập tin đích (lưu ý trường hợp tên dài)
- Kiểm tra trên bảng FAT xem còn đủ số cluster trống để chứa nội dung của tập tin đích không
- Copy thông tin (các entry) của tập tin nguồn sang các entry tìm được của tập tin đích
- Ghi giá trị vào các phần tử FAT trống tìm được theo dạng danh sách liên kết, đồng thời copy các sector nội dung tập tin nguồn vào các sector nội dung tương ứng tìm được của tập tin đích



# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Di chuyển tập tin (MOVE)

- Tìm đủ số entry trống liên tiếp nhau trên bảng thư mục (RDET/ SDET) để chứa thông tin của tập tin đích (lưu ý trường hợp tên dài)
- Copy thông tin (các entry) của tập tin nguồn sang các entry tìm được của tập tin đích
- Xóa thông tin của tập tin nguồn

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Đổi tên tập tin/ thư mục (REN)

- Xác định entry chính trong bảng thư mục (RDET/ SDET) chứa thông tin của tập tin/ thư mục dựa vào phần tên và phần mở rộng (lưu ý trường hợp tên dài)
- Nếu tên tập tin không cần thêm các entry phụ
  - Cập nhật lại phần tên và phần mở rộng
- Nếu tên tập tin cần thêm các entry phụ
  - Tìm đủ số entry trống liên tiếp nhau trên bảng thư mục (RDET/ SDET) để chứa thông tin của tập tin đích (lưu ý trường hợp tên dài)
  - Copy thông tin (các entry) của tập tin nguồn sang các entry tìm được của tập tin đích

# Một số thao tác trên hệ thống tập tin FAT (tt)

## ❖ Quick format

- Giữ lại các thông số cũ của phân vùng,
- Cập nhật lại trạng thái các cluster đang chứa dữ liệu thành trống và cho tất cả entry trên bảng thư mục gốc về trạng thái trống.
- Chức năng này tương đương với việc xóa tất cả mọi tập tin & thư mục đang tồn tại trên phân vùng, nhưng thời gian thi hành rất nhanh, có thể nhanh hơn thời gian xóa một tập tin

## ❖ Full format

- Các thông số của từng thành phần trên phân vùng sẽ được xác định lại.
- Để tạo ra những dạng thức mới phù hợp hơn cho phân vùng. Chức năng này dĩ nhiên cũng được dùng cho những phân vùng chưa được định dạng.

# Ví dụ

- ❖ Xét đĩa mềm 1.44MB (có 2880 sector), để các tập tin trên vol có thể truy xuất nhanh & an toàn hơn ta có thể cho  $SC = 4$  (sector),  $SB = 1$  (sector),  $SR = 32$  (entry) = 2 (sector),  $nF = 2$ .

Thay các giá trị trên vào đẳng thức  $SB + nF \cdot SF + SR + SD = SV$  ta được  
 $1 + 2SF + 2 + SD = 2880$  (sector), hay  $2SF + SD = 2877$  (sector) (\*)

(\*)  $\Rightarrow SD < 2877$  (sector) = 719.25 (cluster) (vì  $SC = 4$  sector).

$\Rightarrow$  Loại FAT tối ưu nhất (về kích thước) là **FAT12**, vì  $SD < 4079$  (cluster)

- **Giả sử  $SF = 1$**  (sector): (\*)  $\Rightarrow SD = 2875$  (sector) = 718.75 (cluster)

$\Rightarrow$  Vùng dữ liệu có 718 cluster, nên bảng FAT phải có  $718 + 2 = 720$  phần tử, do đó  $SF = (720 \cdot 1.5) / 512 = 2.1x$  (sector)

Bảng FAT phải chiếm 3 sector – mâu thuẫn với giả thiết  $SF = 1$ .

Vậy kích thước bảng FAT của vol này không thể là 1 sector

- **Giả sử  $SF = 2$**  (sector): tương tự, ta vẫn thấy mâu thuẫn, tức kích thước bảng FAT phải lớn hơn 2 sector.

- **Giả sử  $SF = 3$**  (sector): (\*)  $\Rightarrow SD = 2871$  (sector) = 717.75 (cluster).

$\Rightarrow$  Vùng dữ liệu có 717 cluster, nên bảng FAT phải có  $717 + 2 = 719$  phần tử, do đó  $SF = (719 \cdot 1.5) / 512 = 2.1x$  (sector)

$\Rightarrow$  Bảng FAT phải chiếm 3 sector – phù hợp với giả thiết  $SF = 3$ .

Vậy kích thước bảng FAT của vol này là 3 sector.

# Fragmentation

- ❖ Một bảng FAT gọi là bị phân mảnh nếu xảy ra ít nhất một trong 2 điều kiện sau:
  - Các phần tử FAT của 1 tập tin không liên tiếp nhau
  - Các phần tử FAT của các tập tin không liên tiếp nhau
- Truy xuất chậm
- Defragmentation

