# NGUYÊN LÝ HỆ ĐIỀU HÀNH

Phạm Đăng Hải haipd@soict.hust.edu.vn

Bộ môn Khoa học Máy tính Viện Công nghệ Thông tin & Truyền Thông



Ngày 14 tháng 2 năm 2020

### Chương 4 Quản lý hệ thống file

2 / 108

### Giới thiệu

- Bộ nhớ ngoài (*đĩa từ, băng từ, đĩa quang,..*): dung lượng lớn và cho phép lưu trữ lâu dài
  - Được người dùng sử dụng lưu trữ dữ liệu và chương trình
  - Dữ liệu và chương trình được lưu dưới dạng file  $(t\hat{q}p\ tin/t\hat{q}p)$ ⇒ Tạo nên hệ thống file

  - Hệ thống file gồm 2 phần riêng biệt
     Các file: Chứa dữ liệu/chương trình của hệ thống/người dùng
     Cấu trúc thư mục: Cung cấp các thông tin về file
- ullet Hệ thống file lớn  $\Rightarrow$  Quản lý như thế nào?
  - Các thuộc tính của file, thao tác cần phải cung cấp?
- Lưu trữ và truy xuất dữ liệu trên thiết bị lưu trữ như thế nào? Phương pháp cung cấp không gian lưu trữ, quản lý vùng tự do ⇒Khó khăn phải trong suốt với người dùng (*tính thuận tiện*)
- $\bullet$  Các file dữ liệu /chương trình có thể sử dụng chung
  - Đảm bảo tính toàn vẹn dữ liệu và loại bỏ truy nhập bất hợp lệ?
- ullet Dữ liệu không lưu trữ tập trung  $\Rightarrow$  hệ thống file phân tán
  - Truy nhập file từ xa, đảm bảo tính toàn vẹn...



Notes			
Notes			
Notes			

Chương 4: Quản lý hệ thống file		
Nội dung chính		Notes
1 Hệ thống file		
2 Cài đặt hệ thống file		
3 Tổ chức thông tin trên đĩa từ		
4 Hệ thống FAT		
	101	
4 / 108		
Chương 4: Quản lý hệ thống file  1. Hệ thống file		Notes
Nội dung chính		Notes
1 Hệ thống file		
2 Cài đặt hệ thống file		
3 Tổ chức thông tin trên đĩa từ		
4 Hệ thống FAT		
5 / 108	_	
Chương 4: Quản lý hệ thống file 1. Hệ thống file		N
1.1 Khái niệm file		Notes
1 Hệ thống file		
<ul><li>Khái niệm file</li><li>Cấu trúc thư mục</li></ul>		
• Cau truc tilu mije		

### lệ thống file Giới thiệu • Thông tin lưu trữ trên nhiều phương tiện/thiết bị lưu trữ khác nhau Ví dụ: Đĩa từ, băng từ, đĩa quang... Thiết bị lưu trữ được mô hình như một mảng của các khối nhớ • File là tập thông tin ghi trên thiết bị lưu trữ. • File là đơn vị lưu trữ của hệ điều hành trên bộ nhớ ngoài File bao gồm dãy các bits, bytes, dòng, bản ghi,... mang ý nghĩa được định nghĩa bởi người tạo ra Cấu trúc của file được định nghĩa theo loại file • File văn bản: Chuỗi ký tự tổ chức thành dòng • File đối tượng: Bytes được tổ chức thành khối để chương trình liên kết (linker) hiểu được • File thực thi: Chuỗi các mã lệnh có thể thực hiện trong bộ nhớ 7 / 108 Các thuộc tính file • Tên file (Name): Chuỗi ký tự (hello.c) Thông tin lưu dưới dạng người dùng có thể đọc được Có thể phân biệt chữ hoa/chữ thường • Đảm bảo tính độc lập của file với tiến trình, người dùng.. A tạo file hello.c bằng notepad trên hệ Windows B dùng emacs trên linux sửa lại file bởi xác định tên hello.c • Định danh (*Identifier*): Thẻ xác định duy nhất một file • Kiểu (Type): Dùng cho hệ thống hỗ trợ nhiều kiểu file • Có thể xác định kiểu file dựa trên một phần của tên file • Ví dụ: .exe, .com/ .doc, .txt/ .c, .jav, .pas/ .pdf, .jpg,... • Dựa trên kiểu, HĐH sẽ thao tác trên tập tin phù hợp Thực hiện file thực thi mà file nguồn đã sửa ⇒ Dịch lại Nháy đúp vào một file văn bản (\*.doc)⇒ Gọi word processor • Vị trí (Position): Trỏ tới thiết bị và vị trí của file trên đó • Kích thước (Size): Kích thước hiện thời/ tối đa của file • Bảo vệ (Protection): Điều khiển truy nhập: Ai có thể đọc/ghi... • Thời gian (*Time*): Thời điểm tạo, sửa đổi, sử dụng cuối ... Các thuộc tính file (tiếp tục) • Thuộc tính file được lưu trong cấu trúc dữ liệu: Bản ghi file • Có thể chỉ chứa tên file và định danh file; định danh file xác định các thông tin còn lại Kích thước từ vài bytes lên tới kilobytes • Các bản ghi file được lưu giữ trong **Thư mục file** • Kích thước có thể đạt tới Megabytes • Thường được lữu trữ trên thiết bị nhớ ngoài Được đưa từng phần vào bộ nhớ khi cần thiết

Thư mục file

Không gian lưu trữ

\ hdh.pdf

hello c

Notes		
Notes		
	-	

### Chương 4: Quản lý hệ thống file 1. Hệ thống file

1. He thong me

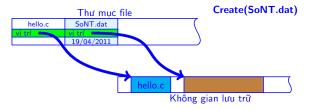
### Các thao tác cơ bản

- Tạo file (*Create*)
- Ghi file (Write)
- Doc file (Read)
- Thay đổi vị trí trong file (Seek)
- Xóa file (Delete)
- Thu gọn file (*Truncate*)
- **0** . . .



10 / 10

### Chương 4: Quản lý hệ thống file 1. Hệ thống file 1.1 Khái niệm file Các thao tác cơ bản : Tạo file



- Tìm vùng tự do trong không gian lưu trữ của hệ thống file
   Cung cấp vùng trống như thế nào?
- Tạo một phần tử mới trong thư mục file
- Lưu tên file, vị trí của file và các thông tin khác

11 / 108



### Chương 4: Quản lý hệ thống file 1. Hệ thống file 1.1 Khái niệm file Các thao tác cơ bản : Ghi file

	Thư mục	file	Write(SoNT.dat, 17)
hello.c	SoNT.dat	kiemtra.pdf	Write(SoNT.dat, 19)
vị trí	vị trí		\ Write(SoNT.dat, 23)
	19/04/2011		Write(SoNT.dat, 29)
		_ \	wayayay ?
		hello.c	Không gian lưu trữ

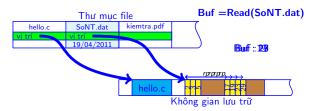
- Lời gọi hệ thống Write() yêu cầu tên file và dữ liệu được ghi
- Dùng tên file, tìm kiếm file trong thư mục file
- Dựa vào trường vị trí, tìm vị trí của file trên thiết bị lưu trữ
- Hệ thống lưu con trỏ ghi (write pointer) để chỉ ra vị trí ghi
  - Con trỏ ghi thay đổi sau mỗi thao tác ghi



N	O.	te	S

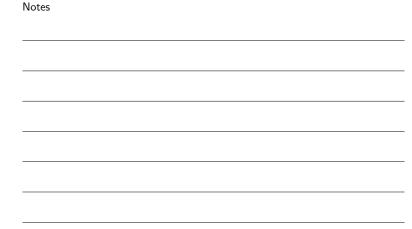
Notes

# Các thao tác cơ bản : Đọc file

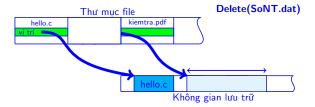


- Lời gọi hệ thống Read() yêu cầu tên file và vùng đệm ghi KQ
- Dùng tên file, tìm kiếm file trong thư mục file
- Dựa vào trường vị trí, tìm vị trí của file trên thiết bị lưu trữ
- Hệ thống lưu con trỏ đọc (read pointer) chỉ ra vị trí được đọc • Con trỏ đọc thay đổi sau mỗi thao tác đọc dữ liệu
- $\bullet$  Dùng một con trỏ cho cả thao tác đọc và ghi: con trỏ file  $_{13\,/\,108}$





# Các thao tác cơ bản : Xóa file



- Dùng tên file, tìm kiếm file trong thư mục file
- Vùng nhớ được xác định bởi 2 trường vị trí và kích thước được giải phóng để có thể sử dụng lại bởi các file khác
- Xóa phần tử tương ứng trong thư mục file
- Xóa logic / xóa vật lý



Chương 4: Quản lý hệ thống file	
1. Hệ thống file	
1.1 Khái niệm file	
Các thao tác cơ bản : Thay đổi vị trí trong file và thu gọn file	
	Π

- Thay đổi vị trí trong file
  - Duyệt thư mục để tìm phần tử tương ứng

  - Con trỏ file được thay bằng giá trị thích hợp
    Thao tác này không yêu cầu một hoạt động vào/ra
- Thu gọn file
  - Được sử dụng khi người sử dụng muốn xóa nội dung file nhưng vẫn giữ nguyên các thuộc tính

    Tìm kiếm file trong thư mục file

  - Đặt kích thước file về 0
  - Giải phóng vùng nhớ dành cho file

4	r.
	Щ

Notes			

### Các thao tác cơ bản : Một số thao tác khác

- Ngoài các thao tác cơ bản, còn tồn tại nhiều thao tác khác
  - Thêm dữ liệu vào cuối file (append)
  - $\bullet$  Lấy/đặt thông tin thuộc tính file
  - Đổi tên file
- $\bullet$  Có thể được đảm bảo thông qua các thao tác cơ bản. Ví dụ copy file
  - Tạo file mới
  - Đọc dữ liệu từ file cũ
  - Ghi ra file mới



16 / 108

### Các thao tác cơ bản: Đóng mở file

- ullet Các thao tác file phải duyệt thư mục file  $\Rightarrow$  Lãng phí thời gian
- Để giải quyết, các tiến trình phải thực hiện mở file (open) trước khi thao tác với file

  - Thao tác mở file: tìm kiếm file trong thư mục file
    Chép phần tử tương ứng vào bảng file mở
    Chứa thông tin về các file đang được mở
- Trả lại con trỏ của phần tử tương ứng trong bản file mở
- Khi có yêu cầu, HĐH tìm kiếm trong bảng file mở
  - Dùng con trỏ trả về của thao tác mở file
- Khi không sử dụng file nữa cần phải đóng (close) file.
- HDH sẽ loại bỏ phần tử tương ứng trong bảng file mở
   Thao tác đóng/mỏ file trong môi trường đa người dùng

  - Dùng 2 loại bảng file mở: Cho từng tiến trình và cho hệ thống • Ghi lại số tiến trình đang mở file (File Open Counter)

    - Tăng/Giảm bộ đếm khi có tiến trình mở/đóng file
      Xóa p/tử tương ứng trong bảng file mở mức hệ thống khi bộ đếm bằng không



17 / 108



- Khái niệm file
- Cấu trúc thư mục

	-
	T(
•	

Notes			
Notes			
Notes			

# Các phân vùng (Partition) Thư mục Phân vùng A Files Δ Phân vùng Files

- 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

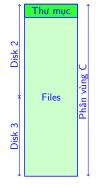


Notes

# Notes



### Các phân vùng (Partition)



20 / 108

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



	4	
П	Į	gn
7		M

Chương 4: Quản lý h	thống file
1. Hệ thống file	
1.2 Cấu trúc thư	

### Các thao tác với thư mục

- Mỗi một phân khu lưu các thông tin về file trong nó
  - Các thông tin file được lưu trữ trong thư mục thiết bị thư muc
- Thư mục là bảng chuyển cho phép ánh xạ từ một tên (file) thành một phần tử trong thư mục
  - Thư mục có thể được cài đặt bằng nhiều cách khác nhau
    - Yêu cầu các thao tác chèn, tạo mới, xóa, duyệt danh sách
- Các thao tác
  - $\bullet$  Tìm kiếm file: Tìm phần tử ứng với một file xác định
  - Tạo file: Tạo file mới cần tạo phần tử trong thư mục

  - Xóa file: Khi xóa file, xóa phần tử tương ứng trong thư mục
     Liệt kê thư mục: Liệt kê files và nội dung phần tử tương ứng trong thu muc

  - trong thu mục

     Đổi tên file: Thay đổi tên file, vị trí trong cấu trúc thư mục

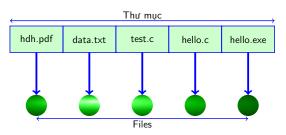
     Duyệt hệ thống file: Truy nhập tất cả thư mục và nội dung tất cả các files trong thư mục (backup dữ liệu lên băng từ)



Notes				

### Chương 4: Quản lý hệ thông file 1. Hệ thống file 1.2 Cấu trúc thư mục

### Thư mục một mức

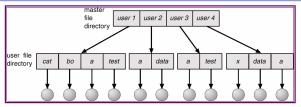


- Cấu trúc đơn giản nhất, các file nằm trong cùng một thư mục
- Số người dùng và số file lớn, khả năng trùng tên file cao
  - Mỗi người dùng một thư mục riêng



22 / 108

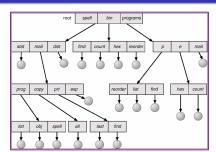
# Chương 4: Quân lý hệ thống file 1. Hệ thống file 1. Cầu trúc thư mục Thư mục hai mức



- Mỗi người sử dụng có một thư mục riêng, khi làm việc với file chỉ duyệt thư mục riêng
- Khi log in, hệ thống sẽ kiểm tra và cho phép người sử dụng làm việc với thư mục riêng
- Khi thêm một người dùng
  - Hệ thống tạo phần tử mới trong Master file directory
  - Tao ra User file directory
- 23 🕭 1 🍪 iả quyết v/đề trùng tên; Hiệu quả khi người dùng độc lập
  - Khó khăn khi muốn dùng chung file



### Chương 4: Quản lý hệ thống file 1. Hệ thống file 1.2 Cấu trúc thư mục **Thư mục cấu trúc cây**



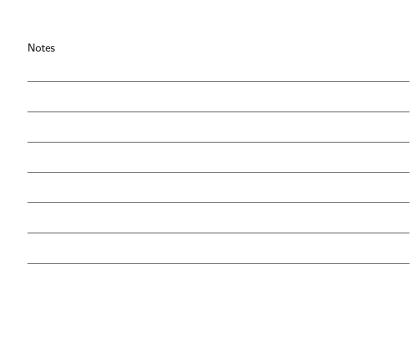
- Tồn tại một đường dẫn (*tương đối/tuyệt đối*) đến một file
- Thư mục con là file được xử lý đặc biệt (bit đánh dấu)
- - Xóa thư mục con ⇒ Xóa hết các cây con của nó



NΙ	- 4
IN	OTES

Notes

# 



Chương 4: Quản lý hệ thống file	
2. Cài đặt hệ thống file	
Nội dung chính	
Hệ thống file	
2 Cài đặt hệ thống file	
3 Tổ chức thông tin trên đĩa từ	
4 Hệ thống FAT	
	•

Notes			

Chương 4: Quản lý hệ thống fil 2. Cài đặt hệ thống file 2.1 Cài đặt thư mục

26 / 108

### 2 Cài đặt hệ thống file

- Cài đặt thư mục
- Các phương pháp phân phối vùng lưu trữ
- Quản lý vùng lưu trữ tự do

Notes			



### Phương pháp

- 1 Danh sách tuyến tính với con trỏ tới các khối dữ liệu
  - Đơn giản cho lập trình
  - Tốn thời gian khi thực hiện các thao tác với thư mục
    - Phải duyệt toàn bộ danh sách  $\Leftarrow$  Dùng cây nhị phân?
- 8 Bảng băm Bảng băm với danh sách tuyến tính

  - Giảm thời gian duyệt thư mục
    Đòi hỏi có một hàm băm hiệu quả

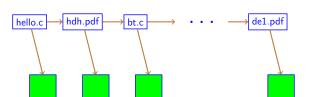
$$\textit{h(Name)} = \frac{\sum_{i=1}^{\textit{Len(Name)}} \textit{ASCII(Name[i])}}{\textit{Table\_Size}}$$

- Vấn đề đụng độ  $\leftarrow$  hàm băm trả về cùng một kết quả với 2 tên file khác nhau Vấn đề kích thước cố định  $\rightarrow$  Tăng kích thước phải tính toán lại những phần đã tồn tại



28 / 108

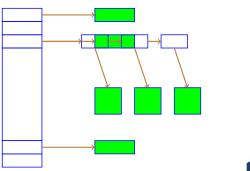
# Danh sách tuyến tính





29 / 108

# Bảng băm



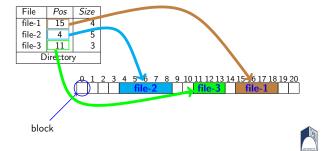


NΙ	_	+~
ıν	O	Les

Notes

Phân phối liên tục

Nguyên tắc: File được phân phối các khối nhớ liên tiếp nhau



### Phân phối liên tục (tiếp tục)

- ullet File có độ dài n và bắt đầu ở khối b sẽ chiếm các khối  $b, b+1, \ldots, b+n-1$ 
  - ullet Hai khối b và b+1 liên tiếp nhau
  - $\Rightarrow$  Không phải dịch chuyển đầu từ khi đọc (trừ sector cuối)
  - ⇒ Tốc độ truy nhập nhanh
  - $\Rightarrow$  truy nhập khối b+i-1 trên thiết bị lưu trữ
- Lựa chọn vùng trống khi có yêu cầu lưu trữ?
  - Các chiến lược First-Fit /Worst Fit /Best Fit
  - Hiện tượng phân đoạn ngoài
- Khó khăn khi muốn tăng kích thước của file

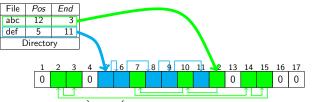


Notes

34 / 108

### ơng 4: Quản lý hệ thống file Cài đặt hệ thống file 2.2 Các phương pháp phân phối vùng lưu trữ Phân phối liên kết

Nguyên tắc: File được phân phối các khối nhớ không liên tục. Cuối mỗi khối là con trỏ, trỏ tới khối tiếp theo



File abc gồm 7 khối: 12, 10, 7, 14, 15, 2, 3 File def gồm 5 khối: 5, 6, 8, 9, 11



35 / 108

# Phân phối liên kết(tiếp tục)

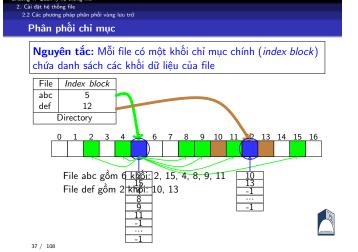
- Chỉ áp dụng hiệu quả cho các file truy nhập tuần tự
- ullet Để truy nhập khối thứ n, phải duyệt qua n-1 khối trước đó
  - Các khối không liên tục, phải định vị lại đầu từ
  - Tốc độ truy nhập chậm
- Các khối trong file được liên kết bởi con trỏ. Nếu con trỏ lỗi?
  - Bị mất dư liệu do mất liên kết tới khối
  - Liên kết tới khối không có dữ liệu hoặc khối của file khác

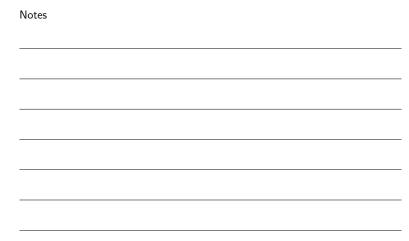
**Giải quyết:** Sử dụng nhiều con trỏ trong mỗi khối  $\Rightarrow$ Tốn nhớ

- Áp dụng: FAT
  - Được sử dụng như danh sách liên kết
  - Gồm nhiều phần tử, mỗi phần tử ứng với một khối
  - Mỗi phần tử trong FAT, chứa khối tiếp theo của file
  - Khối cuối cùng có giá trị đặc biệt (FFFF)
  - Khối bị hỏng có giá trị (FFF7)
  - Khối chưa sử dụng có giá trị (0)
  - Trường vị trí trong bản ghi file, chứa khối đầu tiên của file

	LCT
7	A
•	1

NI .	
Notes	
Notes	





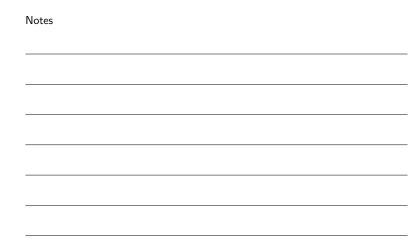
### ơng 4: Quản lý hệ thống file . Cài đặt hệ thống file 2.2 Các phương pháp phân phối vùng lưu trữ Phân phối Phân phối chỉ mục (tiếp tục)

ullet Phần tử thứ i của khối chỉ mục trỏ tới khối thứ i của file

- ullet Đọc khối i dùng con trỏ được khi tại p/tử i của khối chỉ mục
- Tạo file, các phần tử của khối chỉ mục có giá trị null (-1)
- ullet Cần thêm khối i, địa chỉ khối được cấp, được đưa vào p/tử i
- Nhận xét
  - Không gây hiện tượng phân đoạn ngoài
  - Cho phép truy nhập trực tiếp
  - Cần khối chỉ mục: file có k/thước nhỏ, vẫn cần 2 khối

 Khối cho dữ liệu
 Khối chi khối chỉ mục (chỉ dùng 1 phần tử)
 Giải quyết: Giảm kích thước khối ⇒ Giảm phí tổn bộ nhớ ⇒ Vấn đề về kích thước file có thể lưu trữ.

- Sơ đồ liên kết
  - Liên kết các khối chỉ mục lại
  - $\bullet~{\rm P/t} \vec{\rm u}$  cuối của khối chỉ mục trỏ tới khối chỉ mục khác nếu cần
- Index nhiều mức
  - Dùng một khối chỉ mục trỏ tới các khối chỉ mục khác



# Sơ đồ kết hợp (UNIX)

mode	
owners (2)	
timestamps (3)	→ data
size block	
count	→ data
direct blocks —	data : data data
single indirect -	
double indirect	data data
triple indirect	□ data
	. data

- 12 direct block tro tới data block
- Single indirect block chứa địa chỉ khối direct block
- Double indirect block chứa địa chỉ khối Single indirect block
- Triple indirect block chứa địa chỉ khối Double indirect

IOI	
_	•

Notes			

ng 4: Quản lý hệ thống file
Tổ chức thông tin trên đĩa từ

Nội dung chính

1 Hệ thống file
2 Cài đặt hệ thống file
3 Tổ chức thông tin trên đĩa từ

4 Hệ thống FAT



### 3 Tổ chức thông tin trên đĩa từ

- Cấu trúc vật lý của đĩa
- Cấu trúc logic của đĩa



43 / 108

### Đĩa mềm $5\frac{1}{4}$





### 44 / 108

### Đĩa mềm $3\frac{1}{2}$





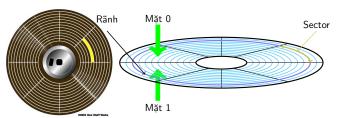
### Notes

-		

### Notes

_			
-			
-			
-			
_			
_			

### Cấu trúc vật lý đĩa mềm

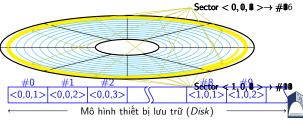


- Mặt đĩa. Mỗi mặt đĩa được đọc bởi một đầu đọc (Header)
- Các đầu từ được đánh số 0, 1
  Rãnh đĩa (*Track*): Các vòng tròn đồng tâm
- Được đánh số 0, 1,... từ ngoài vào trong
  Cung từ (Sector)
- Được đánh số **1**, 2,...



### Định vị thông tin trên đĩa mềm

- Sector đơn vị thông tin hệ thống dùng làm việc với đĩa
- Sector xác định qua tọa độ 3 chiều: Header, Track, Sector • Ví dụ: Boot Sector của đĩa mềm: Sector <0, 0, 1>
- Sector được xác định qua số hiệu sector (*tọa độ 1 chiều*)
  - Vị trí tương đối so với sector đầu tiên của đĩa



47 / 108

Notes

Notes

_			
_			
-			
-			

### Đĩa cứng

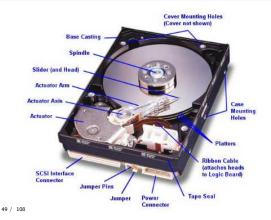




Ν	otes
I۷	Otes

# Chương 4: Quản lý hệ thống file 3. Tổ chức thông tin trên đĩa từ 3.1 Cấu trúc vật lý của đĩa

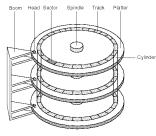
### Dĩa cứng





### Chương 4: Quản lý hệ thống file 3. Tổ chức thông tin trên đĩa từ 3.1 Cấu trúc vật lý của đĩa

### Cấu trúc vật lý đĩa cứng



### Cấu trúc

- Gồm nhiều mặt đĩa, được đánh số từ 0,1
- Các rãnh cùng bán kính tạo nên cylinder, được đánh số từ 0, 1,...
- Các sector trên mỗi mặt của mỗi cylinder, được đánh số từ 1,2,...

### Định vị thông tin

- Tọa độ 3 chiều (H, C, S)
- Tọa độ 1 chiều: Số hiệu sector

 $_{\rm 50\,/\ 108}$  • Nguyên tắc như với đĩa mềm: Sector $\rightarrow$ Header $\rightarrow$ Cylinder



Notes

Notes

### Chương 4: Quản lý hệ thống file 3. Tổ chức thông tin trên đĩa từ 3.1 Cấu trúc vật lý của đĩa

### Truy nhập sector trên đĩa

- Sector là đơn vị thông tin máy tính dùng để làm việc với đĩa từ
- Có thể truy nhập (doc/ghi/format/...) tới từng sector
- Truy nhập sử dụng ngắt BIOS 13h (chức năng 2, 3, 5,...)
  - Không phụ thuộc hệ điều hành
  - Sector được xác định theo địa chỉ <H,C,S>
- Truy nhập sử dụng lời gọi hệ thống
  - Ngắt của hệ điều hành
    - Ví dụ: MSDOS cung cấp ngất 25h/26h cho phép đọc/ghi các sector theo địa chỉ tuyến tính
  - Sử dụng hàm WIN32 API
    - CreateFile()/ReadFile()/WriteFile()...



No	otes			
_				
_				

# Sử dụng ngắt 13h

Thanh ghi	Ý nghĩa
AH	2h:Đọc secror; 3h: Ghi Sector
AL	Số sector cần đọc Các sector phải trên cùng một mặt, một rãnh
DH	Số hiệu mặt đĩa
DL	Số hiệu ổ đĩa. 0h:A; 80h: Đĩa cứng thứ nhất; 81h Đĩa cứng thứ 2
СН	Số hiệu Track/Cylinder (Sử dụng 10 bit, trong đó lấy 2 bit cao của CL)
CL	Số hiệu sector ( <i>chỉ sử dụng 6 bit thấp</i> )
ES:BX	Trỏ tới vùng đệm, nơi sẽ chứa dữ liệu đọc được (khi AH=2h) hoặc dữ liệu ghi ra đĩa (Khi AH=3h)
CarryFlag	CF=0 không có lỗi; CL chứa số sector đọc được CF=1 Có lỗi, AH chứa mã lỗi



Notes

SylinXP hạn chế sử dụng ngắt 13h để truy nhập trực tiếp

# Sử dụng ngắt 13h (Ví dụ)

```
#include <stdio.h>
#include <dos.h>
int main(int argc, char *argv[]){
 union REGS regs;
 struct SREGS sregs;
 int Buf[512];
 int i;
 regs.h.ah = 0x02; regs.h.al = 0x01;
 regs.h.dh = 0x00; regs.h.dl = 0x80;
 regs.h.ch = 0x00; regs.h.cl = 0x01;
 regs.x.bx = FP_OFF(Buf);
 sregs.es = FP_SEG(Buf);
 int86x(0x13,&regs,&regs,&sregs);
 for(i=0;i<512;i++) printf("%4X",Buf[i]);</pre>
 return 0;
```

Notes			

### ơng 4: Quản lý hệ thống file Tổ chức thông tin trên đĩa từ 3.1 Cấu trúc vật lý của đĩa Sử dụng WIN32 API

- HANDLE CreateFile(...): Mở file/thiết bị vào ra
  - ullet LPCTSTR IpFileName,  $\Rightarrow$  Tên file/thiết bị vào ra
    - "\\\.\\C:" Phân vùng /  $\mathring{O}$  đĩa C
    - "\\\.\\PhysicalDrive0"  $\mathring{O}$  đĩa cứng thứ nhất

  - OWORD dwDesiredAccess, ⇒ Than tác với thiết bị
     DWORD dwShareMode, ⇒ Cho phép dùng chung
     LPSECURITY\_ATTRIBUTES IpSecurityAttributes (NULL),
     DWORD dwCreationDisposition, ⇒ Hành động thực hiện

  - $\bullet \ \, \textbf{DWORD} \ \, \text{dwFlagsAndAttributes}, \Rightarrow \text{Thuộc tính} \\$
- HANDLE hTemplateFile (NULL)
   BOOL ReadFile(...)
- - **HANDLE** hFile,⇒File muốn đọc

  - LPVOID lpBuffer, ⇒ Vùng đệm chứa dữ liệu
     DWORD nNumberOfBytesToRead,⇒, số byte cần đọc
  - LPDWORD lpNumberOfBytesRead,⇒ số byte đọc được
  - LPOVERLAPPED IpOverlapped (NULL)
- BOOL WriteFile(...) ⇒Tham số tương tự ReadFile()



Notes		

```
Sử dụng WIN32 API (Ví dụ)
```

#include <windows.h> #include <stdio.h> int main(int argc, char \*argv[]){ HANDLE hDisk; BYTE Buf [512]; int byteread,i;  $\verb|hDisk=CreateFile("\\\.\PhysicalDriveO",GENERIC_READ,\\$ FILE\_SHARE\_READ | FILE\_SHARE\_WRITE, NULL, OPEN\_EXISTING,O,NULL); if (hDisk==INVALID\_HANDLE\_VALUE) printf("Loi thiet bi"); else { ReadFile(hDisk,Buf,512,&byteread,NULL);
for(i=0;i<512;i++) printf("%4X",Buf[i]);</pre> CloseHandle(hDisk); return 0;

Notes		

ơng 4: Quản lý hệ thống file . Tổ chức thông tin trên đĩa từ 3.1 Cấu trúc vật lý của đĩa

### Kết quả thực hiện

H18C77567222BB05584F92E500000FF 8374646CF78BB0827723400001FF -497857868614664F470000F410 10046DBBAA6604110916620000FFAA 0E 7466 AA 823 CBB 113 69 00 00 00 00 00 00 00 00 00 00 00 8806EEFF92A1CA2C3E0000710 F1021C42EE460410990000F6

Notes

56 / 108

### 3 Tổ chức thông tin trên đĩa từ

- Cấu trúc vật lý của đĩa
- Cấu trúc logic của đĩa

1	Notes		
-			
_			
-			
_			



# Cấu trúc logic

- Đĩa mềm: Mỗi hệ điều hành có một chiến lược quản lý riêng
- Đĩa cứng (Có dung lượng lớn)
  - Được chia thành nhiều phân vùng (Partitions, Volumes,..)
    - Mỗi vùng là tập hợp các Cylinder liên tiếp nhau
  - Người dùng ấn định kích thước (Ví dụ dùng: fdisk) • Mỗi phân vùng có thể được quản lý bởi một HĐH riêng

    - HDH format phân vùng theo định dạng được sử dụng
      Tồn tại nhiều hệ thống khác nhau: FAT, NTFS, EXT3,...
  - Trước tất cả các phân vùng là các sector bị che
    - Master Boot Record (MBR): Sector đầu tiên của đĩa

Sector ẩn với phân vùng 2





Notes

Notes

58 / 108

# ơng 4: Quản lý hệ thống file Tổ chức thông tin trên đĩa từ 3.2 Cấu trúc logic của đĩa

### Master Boor Record

- Sector quan trọng nhất của đĩa
- Là sector đầu tiên trên đĩa (Số hiệu 0 hoặc địa chỉ <0, 0, 1>)
- Cấu trúc gồm 3 phần

### Chương trình nhận biết

CT nhận biết

Bảng phân chương

• Đọc bảng phân chương để biết • Vị trí các phân vùng

Phân vùng tích cực (chứa HDH)

Đọc và thực hiện sector đầu tiên của phân vùng tích cực

### Bảng phân chương (64bytes)

- Gồm 4 phần tử, mỗi phần tử 16 bytes
- Mỗi phần tử chứa thông tin một vùng
  - Vị trí, kích thước, hệ thống chiếm



59 / 108

~	
5	^
gii	0
_	1

	-
	_

Cấu trúc một phần tử bảng phân chương	
3.2 Cấu trúc logic của đĩa	
3. Tổ chức thông tin trên đĩa từ	
ương 4: Quản lý hệ thống file	

	Stt	Ofs	Size	Ý nghĩa								
	1	0	1B	Phân vùng tích cực? 80h nếu đúng; 0: Data								
	2	1	1B	Số hiệu mặt đĩa đầu của phân vùng								
fàu	3	2	1W	Số hiệu sector và cylinder đầu của phân vùng								
3 2 1W Số hiệu sector và cylinder đầu của phân vì  F E D C B A 9 8 7 6 5 4 3 2 1  Co Co Co Co Co Co Co So												
	4	4	1B	Mã nhận diện hệ thống. 05/0F: Partition mở								
				rộng; 06:Big Dos; 07:NTFS; 0B: FAT32,								
uôi	5	5	1B	Số hiệu đầu đọc cuối								
đ/c cuối	6	6	1W	Số hiệu sector và cylinder cuối của phân								
d,				vùng. (Số hiệu sector chỉ dùng 6 bit thấp)								
	7	8	1DW	Địa chỉ đầu, tính theo số hiệu sector								
	8	12	1DW	Số sector trong phân vùng								

Notes			

# Ví dụ 1

00 01 01 00	07 FE 3F F8	3F 00 00 00	7A 09 3D 00
80 00 01 F9	0B FE BF 30	B9 09 3D 00	38 7B 4C 00
00 00 81 EB	0F FE FF FF	2B 1D B7 00	72 13 7A 00
00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
55 AA			

Giải mã

Boot	V	trí đầ	iu	V	ị trí cuố	i	#sector	số sector		
Door	Hdr	Cyl	Sec	HdR	Cyl	Cyl Sec "		SO SECTOR		
No	1	0	1	254	248	63	63	4000122		
Yes	0	249	1	254	560	63	4000185	5012280		
No	0	747	1	254	1023	63	12000555	8000370		
-	0	0	0	0	0	0	0			

61 / 108

3. Tổ c 3.2 C	3.2 Cấu trúc logic của đĩa														
V	í dụ	2													
80	01	01	00	07	FE	FF	FF	3F	00	00	00	2C	92	00	02
00	00	C1	FF	0F	FE	$\mathbf{F}\mathbf{F}$	$\mathbf{F}\mathbf{F}$	31	41	8A	03	ØE	D3	1D	01
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
55	AA														

55	AA	טט	ยย	Ø6	1 6	OU	И	1	שט	שט	l Ø	Ø	ИИ	00	טט	טט	טט	שט
+	+		 Dowin			+-		+-						D. 1		-+	Numbau	+
  Active 	Hdr	+		-+-		+	Sys	+-		+		+				1	Number Of Sector	ł
YES NO	1   1   0	+   0  1023	 ( 0 (3697	•													3591852 8731790	
		: 0 : 0		)¦ )¦	0	l		i	0	: @			0		0 0	-	0 0	

62 / 108

### Bảng phân chương mở rộng

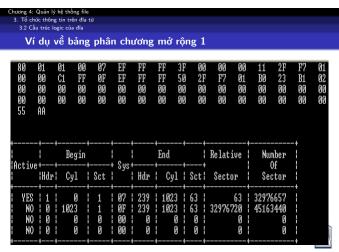
- Khi trường nhận diện có giá trị 05 hoặc 0F, partition tương ứng là partition mở rộng
- ứng là partition mở rộng
  Partition mở rộng được tổ chức như một đĩa cứng vật lý
  Sector đầu tiên là MBR, chứa thông tin về các phân vùng trong partition mở rộng này
  Các phần tử trong partition mở rộng có thể là partition rộng
  Cho phép tạo hơn 4 ổ đĩa logic

		<del></del>	Phâr	n vùng i	1	<b>→</b>	Pł	nân vù	ng m	ở rộng	
MBR						МІ	ŖR				
_4		7					L				$\longrightarrow$
	Act	Size	Sys	#Secto	r		Act	Size	Sys	#Secto	or 🥿
	80		07			A	00		0B		
	00		0B				00		05		1
	00		0F								
											1
	Bá	ảng ph	ân c	nương	•	Βả	ng	ohân c	hươn	g mở ri	ộng
62 / 100											

Notes			

Notes			

Notes				



64 / 100

3. Tổ ch 3.2 Cấ	ức thông u trúc lo	hệ thống fil g tin trên đĩa ogic của đĩa về bảng	từ	n ch	ương	mở rớ	ing 2			
		rtition								
00 00	01 00	C1 FF C1 FF	06 05	EF EF	FF FF	FF 3 FF 9			51 E8 20 3B	76 I
00	00	00 00	00	00	00	00 0	0 00	00 00	00 00	00
00 55	00 AA	00 00	00	00	00	00 0	0 00	00 00	00 00	00
!	!	Begin	+					Relative		- <del>+</del> !
Active 	Hdr:		Sct					Sector	l Of Sector	
NO	1	1023	1	06	239	1023	63	63	24569937	
NO.	: 0 : 0	1023 1 0	1 ! 0	1 05 1 00	239 10	1023 1023	163   10	24570000 A	1 20593440 1 0	
: NII										

65 / 108

3. Tổ chứ 3.2 Cấu	c thông trúc lo	hệ thống fil g tin trên đĩa ogic của đĩa về bảng	từ	n ch	ương	mở rớ	òng 3				
00 00 00	d Pai 01 00 00 00 AA	rtition C1 FF 00 00 00 00	(Sector 0B 00 00 00	r num) EF 00 00 00	per 579 FF 00 00 00	546720) FF 3 00 0 00 0	F 00 0 00 0 00	) 00 00 ) 00 00	E1 3A 00 00 00 00 00 00	3A 00 00 00	01 00 00 00
	Hdr	+ Begin +   Cvl		 + Sys:			·	Relative Sector	+   Number   Of   Sector	+	
NO NO NO	<del> </del>	1023	1   1   0   0	   0B   00   00   00	·	+	63		20593377 0 0 0	-+	

Notes			

Notes			

Global Unique Identifier Partation Table - GPT

• LBA: Logical Block Addressing

• UEFI: Unified Extensible Firmware Interfafe

• BIOS : Basic Input Output System

- Địa chỉ một sector bị giới hạn 32 bit  $\Rightarrow$  Kích thước phân vùng bị giới hạn
- GPT
  - LBA 0: Protective Master Boot Record
  - LBA 1: Header GPT
  - LBA 2→33: Bảng phân chương, chiếm 32 sector
    - Gồm 128 phần tử, mỗi phần tử 128 byte
    - Sử dụng 8 byte để ghi đia chỉ (LBA) đầu và cuối của phân vùng
  - $\bullet$  LBA 34  $\to \ldots$  Các phân vùng
  - $\bullet$  LBA -33  $\to$  -2 32 sector, backup cho bảng phân chương
  - LBA -1 Sector cuối, backup cho Header GPT





Notes

Chương	4:	Quả	n lý	hệ	thống	file
4 Há	th	ốnσl	FAT			

### Nội dung chính

- Hệ thống file
- Cài đặt hệ thống file
- 3 Tổ chức thông tin trên đĩa từ
- 4 Hệ thống FAT



68 / 108

### Các hệ thống file

Tồn tại nhiều hệ thống file khác nhau

- Hệ thống FAT
  - FAT 12/ FAT16 dùng cho MSDOS

  - FAT32 dùng từ WIN98
    12/16/32: Số bit dùng để định danh cluster
- Hệ thống NTFS

  - Sử dụng trong WINNT, WIN2000
    Dùng 64 bit để xác định một cluster
  - Ưu việt hơn FAT trong bảo mật, mã hóa, nén dữ liệu,...
- Hệ thống EXT3
  - Sử dụng trong Linux
- Hệ thống CDFS
  - Hệ thống quản lý file trong CDROM
  - Hạn chế về độ sấu cây thư mục và kích thước tên
- Hệ thốngs UDF
  - Phát triển từ CDFS cho DVD-ROM, hỗ trợ tên file dài

	_
	li.
	Ц
7	

lotes			
lotes			

# Notes Cấu trúc phân vùng cho FAT FAT12/16 ullet Số cluster lớn nhất FAT12: $2^{12}-18$ ; FAT16 : $2^{16}-18$ • K/thước max: FAT12: 32MB; FAT16: 2GB/4GB (32K/64K Cluster) FAT32 ullet Chỉ dùng 28 bit $\Rightarrow$ Số cluster lớn nhất $2^{28}-18$ • K/thước max: 2TB/8GB/16TB (8KB/32KB/64KB Cluster) Cấu trúc logic của hệ thống FAT FAT16 Vùng hệ thống Vùng dữ liệu BS FAT1 FAT2 ROOT | FILES | | | FAT32 FAT1 FAT2 Vùng hệ thống Vùng dữ liệu Notes 4 Hệ thống FAT Boot sector • Bảng FAT (File Allocation Table) • Thư mục gốc 71 / 108 Notes Cấu trúc Bảng Tham số Chương trình mồi

Sector đầu tiên của phân vùngCấu trúc gồm 3 phần

Gau truc goin 3 phan
Bảng tham số đĩa (BPB: Bios Parameter Block)
Chương trình mồi (Boot strap loader)
72 / 108
Chữ ký hệ thống (luôn là 55AA)

### Chương 4: Quân lý hệ thông file 4. Hệ thống FAT 4.1 Boot sector Cấu trúc bảng tham số đĩa - Phần chung

Stt	Ofs	Kt	Giá tri mẫu	Ý nghĩa
1	0	3B	EB 3C 90	Nhảy đến đầu chương trình mồi
2	3	8B	MSDOS5.0	Tên hệ thống file đã format đĩa
3	11	1W	00 02	K/thước 1 sector, thường là 512
4	13	1B	40	Số sector cho một cluster (32K-Cluster)
5	14	1W	01 00	Số scts đứng trước FAT/Số scts để dành
6	16	1B	02	Số bảng FAT
7	17	1W	00 02	Số phần tử của ROOT. FAT32: 00 00
8	19	1W	00 00	$\sum$ sector trên đĩa (< 32M) hoặc 0000
9	21	1B	F8	Khuôn dạng đĩa (F8:HD, F0: Đĩa1.44M)
10	22	1W	D1 09	Số sector cho một bảng FAT(209)
11	24	1W	3F 00	Số sector cho một rãnh (63)
12	26	1W	40 00	Số đầu đọc ghi ( <i>64</i> )
13	28	1DW	3F 00 00 00	Số sector ẩn- Sectors trước volume 63
14	32	1DW	41 0C 34 00	Tổng số sector trên đĩa (3411009)

Notes			

### Chương 4: Quản lý hệ thống file 4. Hệ thống FAT 4.1 Boot sector Cấu trúc bảng tham số đĩa - Phần dành cho FAT12/FAT16

Stt	Ofs	Kt	Giá trị mẫu	Ý nghĩa
15	36	1B	80h	Số hiệu ổ đĩa vật lý 0: ổ A; 80h: ổ C
16	37	1B	00	Để dành/Byte cao cho trường <i>#ổ đĩa</i>
17	38	1B	29h	Boot sector mở rộng 29h
18	39	1DW	D513 5B24	Volumn Serial number(245B-13D5)
19	43	11B	NO NAME	Volumn Label: nhãn đĩa ( <i>không dùng</i> )
20	54	8B	FAT16	Để dành, thường là đoạn text miêu tả
				dạng FAT
21	62	-		Bootstrap loader

### Ví dụ

$\mathbf{E}\mathbf{B}$	3C	90	4D	53	44	4F	53	35	2E	30	00	02	02	06	00
02	00	02	00	00	F8	F5	00	3F	00	FF	00	3F	00	00	00
C1	EB												20		
	45							54					20	33	C9
	/4 / II	18													

Notes \_\_\_\_\_

Chuong 4: Quản lý hệ thống file 4. Hệ thống FAT 4.1 Boot sector Ví dụ giả mã bảng tham số đĩa của FAT16



Notes			

### Chương 4: Quản lý hệ thống file 4. Hệ thống FAT 4.1 Boot sector Cấu trúc bảng tham số đĩa - Phần dành cho FAT32

Stt	Ofs	Kt	Giá trị mẫu	Ý nghĩa
15	36	1DW	C9 03 00 00	Tổng số sector cho bảng FAT
16	40	1W	00 00	Flags: #FAT chính(Không dùng)
17	42	1W	00 00	Version: Phiên bản FAT32 ( <i>Không dùng</i> )
18	44	1DW	02 00 00 00	Số hiệu cluster bắt đầu của ROOT
19	48	1W	01 00	#sector chứa File System information
20	50	1W	06 00	Số hiệu sector dùng backup Bootsector
21	52	12B	00 00	Để dành
22	64	1B	00	Số hiệu ổ đĩa vật lý 0: ổ A; 80h: ổ C
23	65	1B	00	Để dành/Byte cao cho trường #Driver
24	66	1B	29	Boot sector mở rộng. Luôn có giá trị 29h
25	67	1DW	62 0E 18 66	Volumn Serial number
26	71	11B	NO NAME	Volumn Label: Nhãn đĩa (Ko s/dụng)
27	82	8B	FAT32	Để dành, thường là đoạn text miê
				dạng FAT

76 / 108

Notes			

Chươn	ıg 4: Quản	lý hệ th	ống file	:											
4. F	lệ thống FA	NT.													
	1 Boot sec														
	Ví dụ	Pos	+	otor	ci'i a	mât	hâ	thấn	~ d2	na E	ΑТЭ	2			
	vi uụ	DUU	il Se	ctor	cua	ΠΙŲ	ıııė	LIIOII	g uu	ilig r	AIJ				
ER	58	90	4D	53	44	4F	53	35	2E	30	00	02	10	24	00
И2	שש	ИИ	øø	00	F8	00	00	3F	00	FØ	00	3F	00	00	00
E1	3A	3A	01	3E	27	00	00	00	00	00	00	02	99	00	00
01	99	<b>Ø</b> 6	00	99	99	99	99	99	00	99	99	00	99	00	00
80	00	29	D9	DF	92	BC	4E	4F	20	4E	41	4D	45	20	20
20 7B	20 8E	46 C1	41 8E	54 D9	33 RD	32 00	20 70	20 88	20 4E	33 92	C9 8A	8E 56	D1 40	BC B4	F4 08
CD	8E 13	73	8E 05	B9	FF	FF	8A	88 F1	4E 66	02 0F	В6	C6	40	66	08 0F
B6	Di	<b>ล์</b> ดี	E2	3F	F7	E2	86	CD	СØ	ED	06	41	66	ØF	B7
Č9	66	F7	ĔĨ	66	89	46	F8	83	ŽĔ	16	йŏ	<del>7</del> 5	38	83	7Ė
ŽÁ	ĕĕ	77	32	66	8B	46	1Ĉ	66	83	CØ	йč	ВĎ	00	80	B9
01	00	E8	2B	00	E9	48	03	AØ	FA	7D	B4	7D	8B	FØ	AC
84	ĊØ	74	17	3C	FF	74	09	B4	ØE	$\mathbf{B}\mathbf{B}$	07	00	CD	10	EB
EE	ĄØ	FB	7D	EB	E5	<u>AØ</u>	F9	7D	EB	E0	98	CD	16	CD	19
66 53	60	66 68	3B 10	46 00	F8 Ø1	0F 00	82 80	4A 7E	00 02	66 00	6A ØF	00	66 20	50 00	06 B4
41	66 BB	68 AA	55	8A	56	40	CD	13	02 0F	82	1C	85 00	20 81	FB	55
41	ØF	85	14	00	F6	C1	01	٥F	84	ЙĎ	òŏ	FE	46	02	B4
42	8A	56	40	8B	F4	čĎ	13	BØ	F9	66	58	66	58	66	58
66	58	EB	20	66	33	Ď2	66	ЙF	ΒŹ	4E	18	66	F7	F1	FE
C2	8Ä	CA	66	8B	DØ	66	Ċ1	ĒΑ	10	F7	76	1A	86	D6	8A
56	40	8A	E8	CØ	E4	96	ØA	CC	B8	91	02	CD	13	66	61
0F	82	54	FF	81	C3	00	02	66	40	49	0F	85	71	FF	C3
4E	54	4C	44	52	20 คค	20 00	20 คค	20	20	20 คค	00 88	00	00 88	00 88	00
99 99	00 00	00 00	00 00	00 00	99 90	00 00	NN NN	00 00	00 00	99 90	99 90	00 00	99 90	00 00	00 00
99	00	00	00	00	00	00	00	00	00	99	00	ØD	00 0A	52	65
60	6F	76	65	20	64	69	73	6B	73	20	6F	72	20	6F	74
68	65	72	20	6D	65	64	69	61	2Ĕ	FF	ØĎ	ΘÃ	44	69	73
6B	20	65	72	72	6F	72	FF	ØD	ØA	50	72	65	73	73	20 61
61	6E	79	20	6B	65	79	20	74	6 F	20	72	65		74	61

Notes			

Chương 4: Quản lý hệ thống file		
4. Hệ thống FAT		
4.1 Boot sector		
Kết quả giải mã hệ thống FAT32 bằng chư	ơng t	rình
BIOS PARAMETER BLOCK (BPB) OEM Name		MSDOS5.0
Bytes per sector	•	512

BIOS PARAMETER BLOCK (BPB)	
OEM Name :	MSDOS5.0
Bytes per sector :	512
Sectors per cluster :	16
Sectorss before the first FAT :	36
Number of copies of FAT :	2
Media Desctiptor :	F8h
Sectors per Tracks :	63
Number of Header :	240
Number of Hiden Scts in Volume:	63
Number of Sectors in Volume :	20593377
Number of Sectors per FAT :	10046
Cluster num. of start of ROOT :	2
Sct number of FileSystem Info :	1
Sct number of Boot backup sct :	6
Logical drive number of Volume:	80h
	29h
Serial Number of Volume :	BC92-DFD9
	NO NAME
FAT Type :	FAT32
Boot signature :	55 AA
_	

I	Notes		
-			
_			
_			
-			
-			
-			
-			 

### Chương 4: Quản lý hệ thống file

4. He thong FAT

### File System Information Sector

- Thường là Sector thứ 2 của Volume
  - Ngay sau Boot sector (Sector số hiệu 1)

Cấu trúc

<u>Câu t</u>	ruc		
Stt	Ofs	Size	Ý nghĩa
1	0	1DW	Chữ ký thứ nhất của FSInfo sector. Giá trị các
			byte theo thứ tự: 52h 52h 61h 41h
2	4	480B	Không rõ, thường chứa giá trị 00
3	484	1DW	Chữ ký của File System Information Sector. Giá
			trị các byte theo thứ tự: 72h 72h 41h 61h
4	488	1DW	Số cluster tự do1 nếu không xác định
5	492	1DW	Số hiệu của cluster vừa mới được cung cấp
6	496	12B	Để dành
7	508	2B	Không xác định, thường bằng 0
8	510	2B	Chữ ký Bootsector. Có giá trị 55 AA

Notes

79 / 108

Chươ	ng 4:	Quản	lý l	nệ thốr	ng file
	Hệ th	ống F	ΑТ		

File system information sector của một volume dùng FAT32

52	52	61	41	00	00	00	00	00	00	00	00	00	00	99	00	00	00	00	00
00	00	00	00	00	00	00	ØØ	ØØ	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FΙ	FILE SYSTEM INFO:																		
Fi	rs	t	si	gn	atı	ur	е							=	4	16:	15:	252	2h
		S						8	ion	at	ши	e			6	14	17	272	2h
														-			70		
Νu	ımb	er	· o	f	Fr	ee	C	lus	ŧŧε	rs	:			:				616	10
Νu	ımb		· o	f	Fr	ee	C	lus	ŧŧε	rs	:		n	:					10
Νυ #(	ւտն :1ս	er ist	er	f	Fr ec	ee en	C	lus	ŧŧε	rs	:		n	:				610 246	90 59
Nu #C Bo	imb :lu :ot	er st	er ig	f na	Fr ec tu	ee en: re	C: t 1	lus y f	ite 11]	rs loc	at	io		:			5	610 240 5 6	00 59 9A
Nu #0 Bo	imb :lu :ot	er st	er ig	f na	Freedom ectual tual	ee en re	C: t 1;	lus y f	ite 11]	rs Loc	: :at	io	00	: :	99	00	5!	610 246 5 6	00 59 9A
<b>N</b> u #0 Bo	:1u :1u :ot :00	er st s	er ig	f na 00	Freedra ectus tus 00	ee en re	C: t 1:	lus y f 00	ite 11]	rs Loc	at	io	00 00		00	00 00	5!	616 246 5 6	90 9 9 9 9 9 9 9 9
<b>N</b> u #0 <b>B</b> o 00 00	: 1 u : 1 u : 0 t : 00 : 00	er st s	er ig	f na 00 00	Fr ec tu: 00 00	ee ent re 00 00	C: t 1:	lus y f 00 00	11]	90 00 00	a.t	10 00 00	00 00 00	90 90 90	00 00	00 00 00	5! 00 00	616 246 5 6	30 59 9 A 9 A 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
<b>N</b> u #0 <b>B</b> o 00 00	1 u 1 u 1 o t 00 00	er s t 00	er ig	f na 00 00	Fr ec tu: 00 00 00	ee en re 00 00	C: t 1;	1 u s y f 00 00	11]	90 00 00	at	10 00 00 00	00 00 00	90 90 90	00 00 00	00 00 00	5 90 90 90	616 246 5 6 00 00	99 99 99 99 99
<b>N</b> u <b>B</b> u 00 00 00	10 10 00 00 00	er s t 00 00 00	er 19	f na 00 00 00	Fr ec tu: 00 00 00	ee en re 00 00 00	t 1	1 u s y f 00 00 00	11]	90 90 90 90	at	10 00 00 00	00 00 00 00	90 90 90 90	99 99 99	00 00 00 00	5. 90 90 90 90	616 246 5 6 00 00 00	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
<b>N</b> u <b>B</b> o 00 00 00 00	1 u 1 u 00 u 00 00 00 00	er s t 00 00 00 00	er 19	f na 00 00 00 00	60 00 00 00 00 00	ee en re 00 00 00 00	C: t1	1 u s y f 00 00 00 00	11]	00 00 00 00 00	a <b>t</b>	10 00 00 00 00	00 00 00 00 00	90 90 90 90 90	99 99 99 99	00 00 00 00	5 99 99 99 99 99	616 246 5 6 00 00 00 00	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
<b>N</b> u <b>B</b> u 00 00 00	10 10 00 00 00	er s t 00 00 00	er 19	f na 00 00 00	Fr ec tu: 00 00 00	ee en re 00 00 00	t 1	1 u s y f 00 00 00	11]	90 90 90 90	at	10 00 00 00	00 00 00 00	90 90 90 90	99 99 99	00 00 00 00	5. 90 90 90 90	616 246 5 6 00 00 00	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

Cấu trúc	bảng th	am số (	đĩa cho h	ệ thống	NTFS 1
4.1 Boot sector					
<ol> <li>Hệ thống FAT</li> </ol>					
Citaong 4. Quantity ne ti	nong me				

Stt	Ofs	Kt	Giá trị mẫu	Ý nghĩa
1	0	3B	EB 52 90	Nhảy đến đầu chương trình mồi
2	3	8B	NTFS	Tên hệ thống file đã format đĩa
3	11	1W	00 02	Bytes per Sector
4	13	1B	80	Sectors per Cluster (4K-Cluster)
5	14	1W	00 00	Reserved sectors. Allways zero
6	16	1B	00	Allways 0 (FAT: Số bảng FAT)
7	17	1W	00 00	Allways 0 ( <b>FAT</b> : Số p/tử của ROOT)
8	19	1W	00 00	Not used by NTFS (FAT:K/thước đĩa)
9	21	1B	F8	Media Type
10	22	1W	00 00	Allway 0 (FAT:Sectors cho FAT)
11	24	1W	3F 00	Sector per Track (63)
12	26	1W	FF 00	Number of Head (255)
13	28	1DW	3F 00 00 00	Hidden sectors (63)
14	32	1DW	00 00 00 00	Not used by NTFS (FAT: \( \sum_{\text{sectors}} \)

Notes			

Notes			

### Chương 4: Quản lý hệ thắng file 4. Hệ thắng FAT 4.1 Boot sector Cầu trúc bảng tham số đĩa cho hệ thống NTFS 2

Stt	Ofs	Kt	Giá trị mẫu	Ý nghĩa
15	36	1DW	80 00 80 00	Not used by NTFS(FAT:Tổng số sec-
				tors cho FAT)
16	40	1LCN	2B 92 00 02	Total sectors (LCN:LONGLONG)
			00 00 00 00	(33591851)
17	48	1LCN	00 00 0C 00	Logical cluster number for MFT
			00 00 00 00	(786432)
18	56	1LCN	22 09 20 00	Logical #cluster for MFT mirroring
			00 00 00 00	(2099490)
19	64	1DW	F6 00 00 00	Clusters per file record segment (246)
20	68	1DW	01 00 00 00	Clusters per index block (1)
21	72	1LCN	A6 CA D7 C6	Volume serial number
			00 D8 6C 24	246C-D800-C6D7-CAA6
22	80	1DW	00 00 00 00	Checksum
23	84	-		Bootstrap loader

82 / 108

83 / 108

Notes	

Chươ	ng 4: Quản	lý hệ t	hống fil	le											
4.	Hệ thống F	AT													
4	.1 Boot sec	ctor													
	Boot	sec	tors	cúa	một	dia	dùng	NI	FS						
EB	52	90	4E	54	46	53	20	20	20	20	99	02	98	99	00
00	00	00	00	00	F8	00	00	3F	00	FF	00	3F	00	00	00
00 00	00 00	00 0C	99 99	80 00	00 00	80 00	99 99	2B 22	92 09	00 20	02 00	99 99	00 00	00 00	00 00
F6	99	00	99 90	00 01	99 88	00 00	99	A6	CA	D7	6C	99 99	D8	6C	24
99	ดด	AN.	99	FA	33	CØ	8E	DØ	BC	99	7C	FB	B8	CØ	07
8E	D8	E8	16	00	B8	00	ØĎ	8E	CØ	33	ĎВ	C6	Ø6	ØE	øø
10	E8	53	ōо	68	00	ØD	68	6A	02	CB	88	16	24	00	<b>B4</b>
08	ĈĎ	13	73	Ø5	B9	ĔĔ	FF	8A	F1	66	ØF	B6	ĈĜ.	40	66
ØF	B6	D1	80	E2	3F	F7	E2	86	CD	CØ	ED	<b>Ø</b> 6	41	66	ØF
<b>B</b> 7	C9	66	F7	E1	66	A3	20	00	C3	<b>B4</b>	41	$\mathbf{B}\mathbf{B}$	AA	55	8A
16	24	00	CD	13	72	ØF	81	$\mathbf{FB}$	55	AA	75	09	F6	C1	<b>Ø1</b>
74	04	FE	Ø6	14	00	C3	66	60	1E	<b>Ø</b> 6	66	A1	10	00	66
03	Ø6	1C	00	66	3B	06	20	00	ØF	82	3A	00	1E	66	6A
00	66	50	<b>Ø</b> 6	53	66	68	10	00	01	00	80	3E	14	99	00
ØF	85	ØC	00	E8	B3	FF	80	3E	14	99	99	ØF	84	61	00
B4 66	42 58	8A 66	16 58	24 1F	00 EB	16 2D	1F 66	8B 33	F4 D2	CD 66	13 ØF	66 B7	58 ØE	5B 18	07 00
66	58 F7	F1	FE	C2	8A	CA	66	8B	DØ	66	C1	EA	10	F7	36
18	ดีด์	86	D6	8A	16	24	00	8A	E8	ČÖ	E4	Ø6	ØÃ	ĈĆ	B8
9 <u>1</u>	02	CD	13	ØF	82	19	00	8C	ČØ	05	20	00	8E	ČØ	66
FF	Ø6	10	ÕÕ	FF	ØE	ØÉ	00	ØF	85	6 F	FF	07	1F	66	61
C3	ÃØ	F8	01	Ē8	09	ดิติ	ÃØ	$\tilde{\mathbf{F}}\tilde{\mathbf{B}}$	01	E8	øŝ	ØØ	$\tilde{\mathbf{F}}\tilde{\mathbf{B}}$	ĔΒ	$\mathbf{FE}$
<b>B4</b>	01	8B	FØ	AC	3C	00	74	09	B4	ØE	BB	07	00	CD	10
$\mathbf{E}\mathbf{B}$	F2	C3	ØD	ØA	41	20	64	69	73	6B	20	72	65	61	64
20	65	72	72	6F	72	20	6F	63	63	75	72	72	65	64	00
ØD	ØA	4E	54	4C	44	52	20	69	73	20	6D	69	73	73	69
6E	67	00	ØD	ØA	4E	54	4C	44	52	20	69	73	20	63	6F
6D	70	72	65	73	73	65	64	99	ØD	ØA	50	72	65	73	73
20	43 72	74 65	72 73	6C 74	2B 61	41 72	6C 74	74 ØD	2B ØA	44 00	65 00	6C 00	20 00	74 00	6F 00
20	72	05	73	74	61	72	74	อก	ØН	99	99	99	99	90	99

Chương 4: Quản lý hệ thống file	
4. Hệ thống FAT	
4.1 Boot sector	
Giải mã bảng tham số của đĩa dùng N	ITFS
EB 52 90 4E 54 46 53 20 20 2	20 20 00 02 08 00 00
00 00 00 00 00 F8 00 00 3F 0	00 FF 00 3F 00 00 00
	72 00 02 00 00 00 00
	39 20 00 00 00 00 00
	CA D7 6C 00 D8 6C 24
00 00 00 00 FA 33 CO 8E DO 1	BC 00 7C FB B8 C0 07
BIOS PARAMETER BLOCK (BPB)	
	: NTFS
Bytes per sector	512
Sectors per cluster	8
Media Desctiptor	F8h
Sectors per Tracks	63 255
Number of Header Number of Hiden Scts in Volume:	
	33591851
	786432
	2099490
Cluster per file Record Seg.	246
	1
	246C-D800-6CD7-CAA6
Checksum	:0
Boot signature	: 55 AA

Notes			

Notes		



- Boot sector
- Bảng FAT (File Allocation Table)
- Thư mục gốc



Notes

Notes

85 / 108

Chương 4: Quản lý hệ thống file 4. Hệ thống FAT 4.2 Bảng FAT

Mục đích

FAT được sử dụng để quản lý các khối nhớ (blocks/clusters) trong vùng dữ liệu của bộ nhớ lưu trữ

- Khối nhớ đang sử dụng
  - $\bullet$  Phân phối cho từng file/thư mục
- Khối nhớ tự do
- Khối nhớ bị hỏng

## Thực hiện như thế nào ?



86 / 10

Chương 4: Quản lý hệ thồng file 4. Hệ thống FAT 4.2 Báng FAT **Phương pháp** 

FAT gồm nhiều phần tử

- Mỗi phần tử có thể 12bit, 16bit, 32bit
- $\bullet$  Mỗi phần tử ứng với 1 khối ( <code>cluster</code> ) trên vùng dữ liệu
  - $\bullet~2$  phần tử đầu (0,1) có ý nghĩa đặc biệt
    - Khuôn dạng đĩa, Bit shutdown, Bit diskerrror
  - Phần tử thứ 2 ứng với cluster đầu của phần Data

					FA	Т								
0		2	3	4	5	6	7	8	9	10				
Spe- valu	cial ues									(				
			\	#2	<u>#</u>	3	#4	7	¥ <b>5</b>	#6	#7	#8	#9	
										DATA				CONTROL CONTROL

NI .			
Notes			

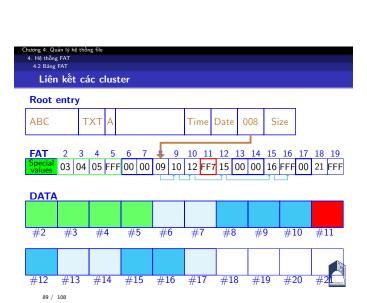
### Chương 4: Quản lý hệ thống file 4. Hệ thống FAT 4.2 Bảng FAT

Cài đặt

Mỗi phần tử của bảng FAT mang một giá trị đặc trưng cho tính chất của cluster tương ứng

FAT[(32)16]12	Ý nghĩa
[(0000)0]000h	Cluster tương ứng tự do
[(0000)0]001h	Giá trị không sử dụng
[(0000)0]002h	Cluster đang được sử dụng. Giá trị đóng vai
$\rightarrow$ [(0FFF)F]FEFh	trò con trỏ, trỏ tới cluster tiếp theo của file
[(0FFF)F]FF0h	Các giá trị để dành, chưa được sử dụng
$\rightarrow$ [(0FFF)F]FF6h	
[(0FFF)F]FF7h	Đánh dấu cluster tương ứng bị hỏng
[(0FFF)F]FF8h→	Cluster đang đc sử dụng và là cluster cuối cùng
$\rightarrow$ [(0FFF)F]FFFh	của file ( <i>EOC:End Of Cluster chain</i> ). Thực tế
	thường dùng giá trị [(0FFF)F]FFFh

88 / 10

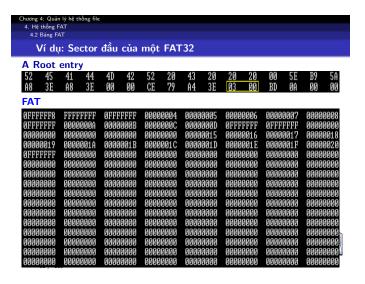


Chương 4: Quản lý hệ thống file	
4. Hệ thống FAT	
4.2 Bảng FAT	
Ví dụ: Đọc một sector của FAT32	

Notes			

Notes			

Notes			



Notes

### 4 Hệ thống FAT

- Boot sector
- Bảng FAT (File Allocation Table)
- Thư mục gốc



Notes

92 / 108

### Cấu trúc thư mục gốc

- Bảng gồm các *bản ghi file* 
  - Mỗi bản ghi có kích thước 32 bytes
    - $\bullet$  Chứa các thông tin liên quan tới một file/thư mục/ nhãn đĩa
- Hệ thống FAT12/FAT16

  - Thư mục gốc nằm ngay sau các bảng FAT Kích thước = Số phần tử tối đa trong thư mục gốc \*  $\frac{32}{512}$
- Hệ thống FAT32
  - Vị trí được xác định dựa vào BPB
    - Trường 18: Số hiệu cluster đầu của ROOT
  - Kích thước không xác định
  - Hỗ trợ tên file dài (LFN: Long File Name)
    - Một file có thể sử dụng nhiều hơn một phần tử



١	Votes			
_				
_				

### Cấu trúc một phần tử

Stt	Ofs	Size	Ý nghĩa
1	0	8B	Tên file
2	8	3B	Phần mở rộng
3	11	1B	Thuộc tính của file
4	12	10B	Không dùng với FAT12/FAT16. Sử dụng với FAT32
4.1	12	1B	Để dành
4.2	13	1B	Thời điểm tạo file, theo đơn vị 10ms
4.3	14	1W	Thời điểm tạo file (giờ - phút - giây)
4.4	16	1W	Ngày tạo file ( <i>tạo bởi ứng dụng hoặc bởi copy sang</i> )
4.5	18	1W	Ngày truy nhập cuối
4.6	20	1W	Số hiệu cluster bắt đầu của file( <i>FAT32: Phần cao</i> )
5	22	1W	Thời gian cập nhật cuối cùng
6	24	1W	Ngày cập nhật cuối (không y/cầu sau ngày tạo file)
7	26	1W	Số hiệu cluster bắt đầu của file (FAT32: Phần thấ
8	28	1DW	Kích thước tính bằng byte
94	/ 108		

94	/	108

# hương 4: Quản lý hệ thống file 4. Hệ thống FAT 4.3 Thư mục gốc

### Cấu trúc một phần tử :Tên file

- Chuỗi ASCII chứa tên file. Các ký tự là chữ in
- Không chấp nhận khoảng trống ở giữa
  - Các câu lệnh copy, del,... không nhận biết tên có dấu trắng
- $\bullet$  Nếu ít hơn 8 ký tự, được chèn các ký tự trống cho đủ 8
- Ký tự đầu có thể mang ý nghĩa đặc biệt
  - 00h: Phần tử đầu tiên của phần chưa dùng đến
  - E5h (ký tư "δ"): File tương ứng với phần tử này đã bị xóa.
    2Eh (ký tư "."): Dây là thư mục con

  - Trường số hiệu cluster bắt đầu chỉ đến chính nó
     Cấu trúc như thư mục con giống như thư mục gốc: gồm các Cau trùc như thu mục còn giống như thu mục gọc: gom các phần tử 32bytes
     ZEh2Eh (ký tự ".."): Đây là thư mục cha của thư mục hiện tại
     Trường số hiệu cluster bắt đầu chỉ đến thư mục cha
     Nếu cha là gốc, #cluster bắt đầu bằng zero (FAT12/16)
  - - Thư mục con nằm trên phần Data, được quản lý như một file
    - ⇒ File của các bản ghi file
       FAT12/16: Thư mục gốc ở vị trí xác định; FAT32: Thư mục gốc cũng nằm trong phần data

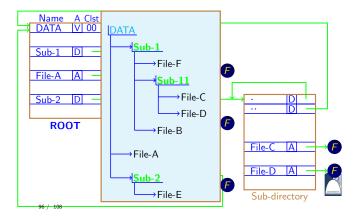


Notes

Notes

95 / 108

# Thư mục con



Notes	

### Notes Cấu trúc một phần tử: Trường thuộc tính 0 0 Α D ٧ S Η R Unused Archive Read only Subdirectory 4 Hidden Volume label System Ví dụ: Byte thuộc tính $\overline{\mathbf{0Fh}}$ : $\boxed{0}$ $\boxed{0}$ $\boxed{0}$ $\boxed{1}$ $\boxed{1}$ $\boxed{1}$ $\boxed{1}$ $\Rightarrow$ Có các thuộc tính Volume label+System+Hidden+Read only **Ghi chú:** Giá trị byte thuộc tính 0x0F không sử dụng trong $\mathit{MS\text{-}DOS} \Rightarrow \mathsf{Dùng} \; \mathsf{d\hat{e}} \; \mathsf{d}\mathsf{á}\mathsf{nh} \; \mathsf{d\hat{a}u} \; \mathsf{la} \; \mathsf{ph}\mathsf{\hat{a}n} \; \mathsf{t\mathring{u}} \; \; \mathit{Long File Name}$ Notes Cấu trúc một phần tử: Trường thời gian H H H H M M M M M M | s | s | s | s - Hours-Minutes--Seconds/2 Có giá trị : 7C56 98 / 108 Notes Cấu trúc một phần tử: Trường ngày tháng -Month--Year - 1980-

# Hệ thống Long File Name (LFN)

	Ofs	Kt	Ý nghĩa
	0	1B	Trường thứ tự.
	1	5W	5 ký tự unicode đầu tiên
	11	1B	Thuộc tính. Đánh dấu là phần tử
Phần tử LFN 3			<i>LFN</i> . Luôn có giá trị 0Fh
Than to LIN 5	12	1B	Để dành (00)
Phần tử LFN 2	13	1B	Checksum: Cho phép kiểm tra tên
I Hall tu LI IV Z			file dài có ứng với tên file 8.3?
Phần tử LFN 1	14	6W	Các ký tự unicode 6,7,8,9,10,11
r IIaii tu Li IV 1	26	1W	Số hiệu cluster. Không dùng (0000)
Phần tử 8.3 (ttt∼n.xxx)	28	1W	Ký tự unicode 12
I Hall tu 0.5 (ttt/=II.xxx)	30	1W	Ký tự unicode 13

100 / 108

### Hệ thống Long File Name: Trường thứ tự

- Cho biết trật tự các phần tử LFN
  - Mỗi phần tử LFN chứa 13 ký tự Unicode
- $\bullet$  Phần tử đầu tiên có giá trị trường thứ tự bằng 1
- Phần tử cuối sẽ dùng bít số 6 để đánh dấu
  - Chỉ dùng tối đa 20 phần tử

  - Sau ký tự cuối cùng là 0x00 0x00.
    Các ký tự không sử dụng có giá trị 0xFF 0xFF
- Bít số 7 (0x80) cho biết phần tử tương ứng đã bị xóa
- Ví dụ file "This is a very long file name.docx"

Entry	Ord	Attr	Data			
LFN 3	0×43	0×0F	ame.docx			
LFN 2	0×02	0×0F	y long file n			
LFN 1	0×01	0×0F	This is a ver			
8.3 Name	THISIS~1.DOC					



101 / 108

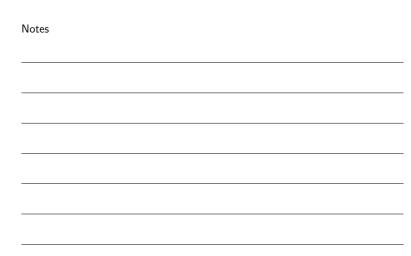
# Ví dụ: Một sector của ROOT

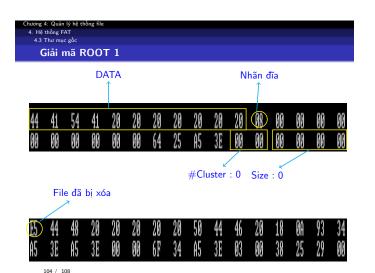
1																
	44 00	41	54	41	20	20	20	20	20	20	20	98	00	00	00	00
	ดด	00	00	00	00	00	64	25	<u>A5</u>	3E	00	00	00	00	00	00
	E5	44	48	20	20	20	20	20	50	44	46	20	18	ØA	93	34
	A5	3E	A5	3E	00	00	6F	34	A5	3E	03	00	38	25	29	00
	41 65	45	00	78	00	65	99	6 D	99	70	00	ØF	00	EF	6C	00
	65	99	73	00	99	00	FF	FF	FF	FF	00	00	FF	FF	FF	FF
	45	58 3E	45	4D	50 00	4C	45 90	53	20	20	20	10	00	C4	9B	34
	A5	JE	A5 00	3E	99 99	00		34 00	A5 00	3E	96 FF	14 0F	00	99	00	00
	42	72 FF	FF	2E	FF	63 FF	00	FF	FF	FF FF	99 00	90 90	00 FF	43	FF	FF
	FF 01	52	77 00	FF 65	77 00	61	FF	64	44 00	42	00 00	ØF	44 00	FF 43	60	FF 00
	6F	90 00	73	00	53	86	00 65	00	63	00	99	90	74	99	FF 69 6F	99
	52	45	41	44	42	49	7E	31	43	20	20	20	00	A6	B2	4B
	92 85	3E	45 A5	3E	92 00	99	76	5c	9A	3E	3F	2E	D1	нь 01	00 00	98 99
	н5 52	3E 45	н5 41	3E 44	4D	42	52	20	43	3E 20	20	2E 20	90 DT	3C	86	5B
	A5	3E	A5	3E	99	9 <u>6</u>	CF	79	43 44	3E	40	2E	BD	9A	øй	90
	но 41	54	00 H3	65	99	6 D	00	70	00	73	90	ØF	00 DD	FØ	00	99
	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	99	FF	FF	FF	DD.
	54	45	4D	50	53	20	20	20	20	20	20	10	00	11	9 A	FF 96
	A5	3E	A5	3E	00	õõ	9B	96	Ã5	ЗE	46	2E	øø	ōō	óΰ	őő
	42	A1	01	δĒ	00	67	00	20	ЙÕ	34	90	ØF	00	12	2E	00
	70	бö	64	00	66	øé	00	õõ	FF	FF	øø	00	FF	FF	FF	FF
	<b>01</b>	42	00	EØ	00	69	00	20	00	67	00	ØF	øø	12	69	00
	Ã3	1Ĕ	6E	00	67	øø	20	õõ	63	øø	00	00	68	ōõ	ВÓ	01
	42	41	49	47	49	4E	7E	31	50	44	46	20	00	ØA	93	34
	ÁŠ	ЗĒ	Á5	ЗĒ	øø	øõ	6F	34	A5	ЗÊ	øз	00	38	25	29	ØØ
	ØØ	00	00	øø	00	00	ØØ	øø	00	øø	øø	00	øø	00	ЙÓ	00
	ØØ	00	ЙÖ	00	00	00	ØØ	øø	ØØ	00	00	00	ØØ	00	ØØ	00
	ØØ	00	øø	00	00	ØØ	00	00	ØØ	00	00	00	ØØ	00	ØØ	00
	øø	00	00	00	00	ØØ	00	øø	ØØ	00	øø	00	ØØ	00	ØЙ	00
	ØØ	00	ดิติ	00	00	00	00	00	00	ดิด	00	00	ØØ	00	00	00
	ØØ	00	00	00	00	00	00	00	00	00	00	00	ØØ	00	ØØ	00 <sup>=</sup>

	LCT
	A
1	V

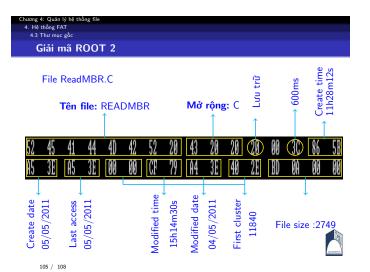
Notes

Notes

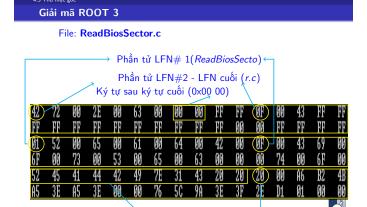








ļ	Votes			
-				
-				
-				
-				



Phần tử 8.3 thường

Notes

Chương 4: Quản lý hệ thống file

### Bài tập

106 / 108

1 Viết chương trình Diskedit

 $\mathsf{READBI}{\sim}1.\mathsf{C}$ 

- Cho phép xem (và sửa chữa) từng sector của một đĩa cứng.
- Các sector được hiện thị dưới cả 2 dạng: Hexa và ASCII
- ② Viết chương trình liệt kê tất cả các phân vùng của ổ đĩa cứng.
  - Nếu phân vùng sử dụng hệ thống file FAT32 hoặc NTFS, đưa ra các thông tin tương ứng
- Viết chương trình đưa ra nội dung của thư mục gốc của đĩa cứng sử dụng FAT32
  - Chỉ sử dụng thủ tục đọc sector trên đĩa
- lacktriangle Nghiên cứu cách tổ chức của các hệ thống file NTFS, EXT3
- $oldsymbol{\circ}$  Xây dựng một hệ thống file trên một đĩa ảo



Notes

107 / 108

### Chương 4: Quản lý hệ thống file

### Kết luận

- 1 Hệ thống file
  - Khái niệm file
  - Cấu trúc thư mục
- 2 Cài đặt hệ thống file
  - Cài đặt thư mục
  - Các phương pháp phân phối vùng lưu trữ
  - Quản lý vùng lưu trữ tự do
- 3 Tổ chức thông tin trên đĩa từ
  - Cấu trúc vật lý của đĩa
  - Cấu trúc logic của đĩa
- 4 Hệ thống FAT
  - Boot sector
  - Bảng FAT (File Allocation Table)
  - Thư mục gốc



Notes			