



## 03 – Hệ thống tập tin FAT

# MỤC TIÊU

- Mô tả chi tiết một hệ thống tập tin cụ thể: FAT

cuu duong than cong. com

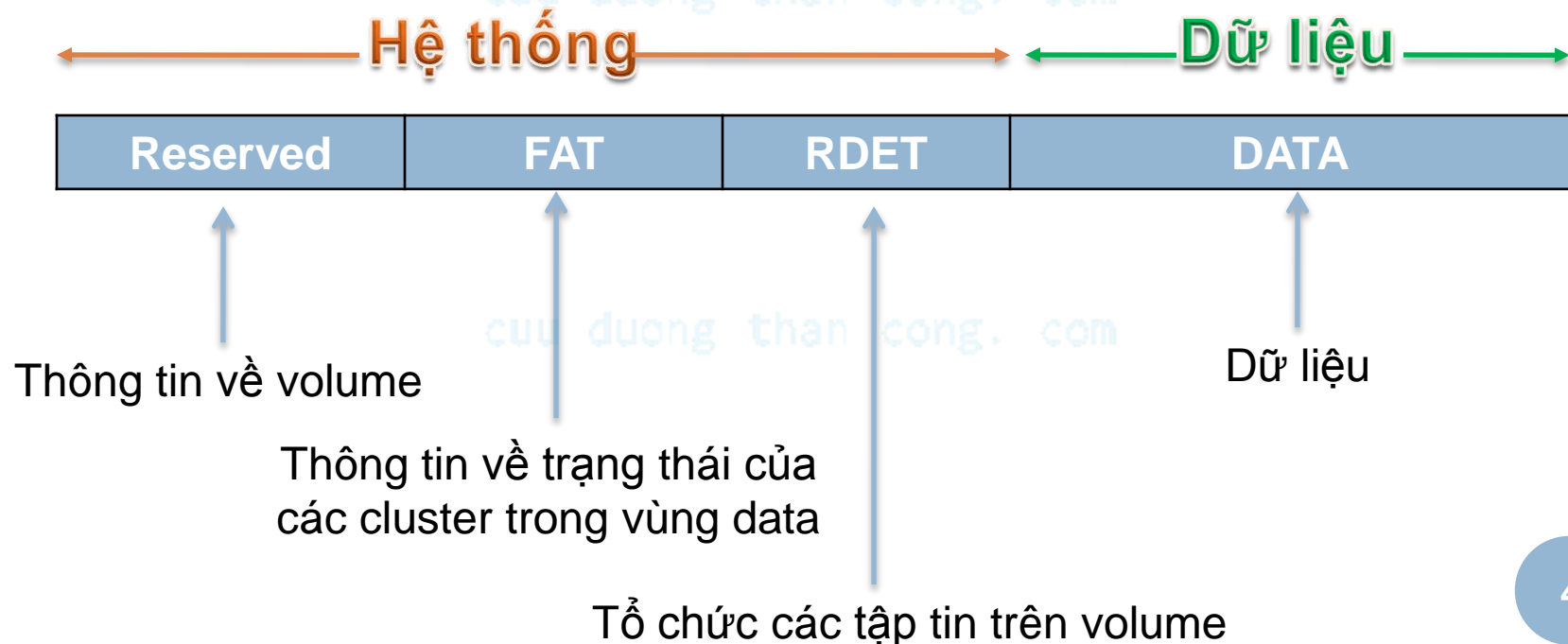
cuu duong than cong. com

# NỘI DUNG

- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

# HỆ THỐNG TẬP TIN FAT - 1

- FAT: File Allocation Table
- Xuất hiện vào cuối những năm 70s và đầu những năm 80s
- Các phiên bản của FAT: FAT12, FAT16, FAT32
  - 12,16, 32: Số bit dùng để đánh STT các khối dữ liệu (cluster)



# NỘI DUNG

- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

# VÙNG DÀNH RIÊNG

- Hay còn gọi là vùng boot sector
- Gồm các sector đầu tiên trong phân vùng
- BIOS Parameter Block (BPB)
  - Sector đầu tiên
  - Tên gọi khác: boot sector, reserved sector, sector thứ 0
  - mô tả về Volume
  - Đoạn chương trình boot hệ điều hành

cuu duong than cong. com

# BOOTSECTOR – CẤU TRÚC

Name	Offset Hex	Size (bytes)	Description	Ký hiệu
BS_jmpBoot	0	3	Lệnh nhảy đến đoạn boot code.	
BS_OEMName	3	8	Version/tên HĐH	
BPB_BytsPerSec	B	2	Số bytes/sector Ví dụ: 512, 1024, 2048 hoặc 4096	
BPB_SecPerClus	D	1	Số sectors/cluster	$S_C$
BPB_RsvdSecCnt	E	2	Số sector để dành (khác 0) (Số sector trước bảng FAT)	$S_B$
BPB_NumFATs	10	1	Số bảng FAT	$N_F$
BPB_RootEntCnt	11	2	FAT12, FAT16: số entry trong bảng RDET FAT32: có giá trị là 0	$N_{RDET}$
BPB_TotSec16	13	2	FAT12, FAT16: tổng số sector của Volume FAT32: có giá trị là 0	$S_V$
BPB_Media	15	1	Loại Volume	
BPB_FATSz16	16	2	FAT12, FAT16: số sector trong 1 bảng FAT FAT32: có giá trị là 0 (BPB_FATSz32)	$S_F$
BPB_SecPerTrk	18	2	Số sectors/track	
BPB_NumHeads	1A	2	Số heads	
BPB_HiddSec	1C	4	Số sector ẩn trước Volume	
BPB_TotSec32	20	4	Số sector trong Volume. Nếu bằng 0, BPB_TotSec16 phải khác 0	$N_V$

**Cấu trúc 36 bytes đầu tiên trong Bootsector**

# BOOTSECTOR – CẤU TRÚC

Name	Offset hexa	Size (bytes)	Description
BS_DrvNum	24	1	Ký hiệu vật lý đĩa (0x00: floppy disks, 0x80: hard disks). <b>NOTE: This field is actually operating system specific.</b>
BS_Reserved1	25	1	Dành riêng
BS_BootSig	26	1	Ký hiệu nhận diện HĐH (0x29).
BS_VolID	27	4	Volume serial number.
BS_VolLab	2B	11	Volume label.
BS_FilSysType	36	8	Chuỗi nhận diện loại FAT: “FAT12 ”, “FAT16 ”, “FAT ”
	3E	448	Boot code
	1FE	2	Dấu hiệu kết thúc bootsector (0x55AA)

**FAT12, FAT16: Cấu trúc 476 bytes còn lại trong Bootsector**



# BOOTSECTOR – CẤU TRÚC

Name	Offset hexa	Size (bytes)	Description
<b>BPB_FATs32</b>	<b>24</b>	<b>4</b>	<b>số sector trong 1 bảng FAT</b> <b>BPB_FATs16 must be 0.</b>
BPB_ExtFlags	28	2	0-3: chỉ số bảng FAT active Bits 4-6: dành riêng 7: 0 – cập nhật lên tất cả các bảng FAT 1 – chỉ cập nhật lên bảng FAT active 8-15: dành riêng
BPB_FSVer	2A	2	Version FAT32 (byte thấp mirror)
<b>BPB_RootClus</b>	<b>2C</b>	<b>4</b>	<b>Chỉ số cluster đầu tiên của RDET (thông thường: 2)</b>
BPB_FSInfo	30	2	Chỉ số sector chứa FSINFO – thông tin sector trống. (thông thường: 1)
BPB_BkBootSec	32	2	Chỉ số sector chứa bản sao của bootsector (thông thường: 6)
BPB_Reserved	34	12	Dành riêng
<b>BS_DrvNum</b>	<b>40</b>	<b>1</b>	<b>Ký hiệu vật lý đĩa (0x00: floppy disks, 0x80: hard disks).</b>
BS_Reserved1	41	1	Dành riêng
BS_BootSig	42	1	Ký hiệu nhận diện HĐH (0x29).
BS_VolID	43	4	Volume serial number..
BS_VolLab	47	11	Volume label.
BS_FilSysType	52	8	Chuỗi nhận diện loại FAT: ” <b>FAT32</b> ”.
	5A	420	Boot code
	1FE	2	Dấu hiệu kết thúc bootsector (0x55AA)

**FAT32: Cấu trúc 476 bytes còn lại trong Bootsector**

# BOOTSECTOR

## ○ Loại Volume

- Các giá trị có thể có: 0xF0, 0xF8, 0xF9, 0xFA, 0xFB, 0xFC, 0xFD, 0xFE, and 0xFF.
  - 0xF8: “fixed” (non-removable) media.
  - 0xF0: removable media
- Ghi chú: giá trị byte này đặt trong byte đầu tiên của bảng FAT

cuu duong than cong. com

cuu duong than cong. com

# NỘI DUNG

- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

# RDET

- RDET = Root Directory Entry Table
- Nằm trên vùng hệ thống (FAT12, FAT16) hoặc trên vùng dữ liệu
- Tổ chức cây thư mục trên phân vùng
- Chia thành nhiều entry, mỗi entry
  - Chiếm 32 bytes
    - Byte đầu tiên: trạng thái của entry
  - Chứa thông tin về 1 tập tin/thư mục
  - Phân loại:
    - Entry chính: chứa thông tin của tập tin
    - Entry phụ: chứa tên tập tin (dành cho LFNs)

0	1	2	3	4	....	
0...1F	20...3F	40...5F	60....7F	80...9F	....	....

# RDET

- Trạng thái của entry

Giá trị	Ý nghĩa
0x00	Entry trống
0x05	Initial character is actually 0xE5
0x2E	'Dot' entry; hoặc '.' or '..'
0xE5	Entry đã bị xoá.

# RDET – CẤU TRÚC ENTRY CHÍNH

OFFSET	ĐỘ DÀI (byte)	NỘI DUNG
0h (0)	8	Tên chính của tập tin
8h (8)	3	Tên mở rộng
Bh (11)	1	Thuộc tính (0-0-A-D-V-S-H-R) Nếu có giá trị là 0x0F thì entry này sử dụng cho LFNs
Ch (12)	10	Không dùng
16h (22)	2	Giờ cập nhật tập tin
18h (24)	2	Ngày cập nhật tập tin
1Ah (26)	2	Cluster bắt đầu
1Ch (28)	4	Kích thước tập tin

# RDET – CẤU TRÚC ENTRY CHÍNH

- Thuộc tính của tập tin

8n	Mask	Mô tả
0	0x01	Read Only
1	0x02	Hidden
2	0x04	System
3	0x08	Volume Label
4	0x10	Subdirectory
5	0x20	Archive
6	0x40	Device (internal use only, never found on disk)
7	0x80	Unused

# RDET – CẤU TRÚC ENTRY PHỤ

Byte Offset	Length	Description
0	1	Số thứ tự của entry
1	10	Các ký tự của tên file (5 ký tự <u>UTF-16</u> )
B	1	Attributes (luôn luôn có giá trị là 0x0F)
C	1	Reserved (luôn luôn có giá trị là 0x00)
D	1	Checksum của tên file MS-DOS
E	12	Các ký tự của tên file (6 ký tự <u>UTF-16</u> )
1A	2	Cluster đầu tiên (luôn luôn có giá trị là 0x0000)
1C	4	Các ký tự của tên file (2 ký tự <u>UTF-16</u> )



# RDET – VÍ DỤ

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00002600	54	45	53	54	20	20	20	20	20	20	08	00	00	00	00	00	TEST .....
00002610	00	00	00	00	00	00	4C	42	92	3A	00	00	00	00	00	00	.....LB':.....
00002620	42	53	20	20	20	20	20	20	49	4D	47	20	00	A6	2B	43	BS IMG . +C
00002630	92	3A	92	3A	00	00	C8	06	5C	39	02	00	00	02	00	00	':':...Ê.\9.....
00002640	E5	65	00	44	00	61	00	69	00	2E	00	0F	00	7D	64	00	âe.D.a.i.....}d.
00002650	6F	00	63	00	00	00	FF	FF	FF	FF	00	00	FF	FF	FF	FF	o.c...ÿÿÿÿ..ÿÿÿÿ
00002660	E5	50	00	68	00	75	00	4C	00	75	00	0F	00	7D	63	00	âP.h.u.L.u...}c.
00002670	5F	00	54	00	65	00	6E	00	46	00	00	00	69	00	6C	00	_.T.e.n.F...i.l.
00002680	E5	48	55	4C	55	43	7E	31	44	4F	43	20	00	4C	2E	43	âHULUC~1DOC .L.C
00002690	92	3A	92	3A	00	00	7A	7E	71	34	03	00	00	88	00	00	':':...z~q4...!..
000026A0	E5	48	00	44	00	54	00	48	00	42	00	0F	00	32	61	00	âH.D.T.H.B...2a.
000026B0	69	00	36	00	2E	00	64	00	6F	00	00	00	63	00	00	00	i.6...d.o...c...
000026C0	E5	44	54	48	42	41	49	36	44	4F	43	20	00	0D	2F	43	âDTHBAI6DOC ../C
000026D0	92	3A	92	3A	00	00	7C	89	71	34	47	00	00	2C	02	00	':':... !q4G.....
000026E0	54	48	5F	48	44	48	20	20	20	20	20	10	00	24	39	43	TH_HDH ...\$9C
000026F0	92	3A	92	3A	00	00	3A	43	92	3A	5D	01	00	00	00	00	':':...:C':}.....
00002700	E5	67	00	5F	00	54	00	48	00	2E	00	0F	00	1E	64	00	âg._.T.H.....d.
00002710	6F	00	63	00	00	00	FF	FF	FF	FF	00	00	FF	FF	FF	FF	o.c...ÿÿÿÿ..ÿÿÿÿ
00002720	E5	43	00	44	00	5F	00	48	00	44	00	0F	00	1E	48	00	âC.D._.H.D....H.
00002730	5F	00	44	00	65	00	63	00	75	00	00	00	6F	00	6E	00	_.D.e.c.u...o.n.
00002740	E5	44	5F	48	44	48	7E	31	44	4F	43	20	00	56	51	43	âD_HDH~1DOC .VQC
00002750	92	3A	92	3A	00	00	4B	64	54	3A	CB	01	00	D4	00	00	':':...KdT:Ê...ô..
00002760	42	69	00	70	00	00	00	FF	FF	FF	FF	0F	00	B2	FF	FF	Bi.p...ÿÿÿÿ..²ÿÿ
00002770	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ
00002780	01	57	00	69	00	6E	00	68	00	65	00	0F	00	B2	78	00	.W.i.n.h.e...²x.
00002790	5F	00	66	00	75	00	6C	00	6C	00	00	00	2E	00	7A	00	_.f.u.l.l.....z.
000027A0	57	49	4E	48	45	58	7E	31	5A	49	50	20	00	72	6F	43	WINHEX~1ZIP .roC
000027B0	92	3A	92	3A	00	00	83	24	5C	3A	CB	01	9E	B9	12	00	':':...!\$:\:Ê..!¹..
000027C0	41	48	00	44	00	54	00	48	00	42	00	0F	00	25	61	00	AH.D.T.H.B...%a.
000027D0	69	00	30	00	38	00	2E	00	64	00	00	00	6F	00	63	00	i.0.8...d...o.c.
000027E0	48	44	54	48	42	41	7E	31	44	4F	43	20	00	24	7D	43	HDTHBA~1DOC .}\$C
000027F0	92	3A	92	3A	00	00	CA	06	5C	39	4E	00	00	36	01	00	':':...Ê.\9N..6..

# SDET

- SDET = Sub Directory Entry Table
- Nằm trên vùng data
- Tổ chức thành các entry như RDET
  - Luôn luôn có 2 entry “.” và “..”
    - Entry “.”: thông tin thư mục nó
    - Entry “..”: thông tin thư mục cha
- Chứa thông tin các tập tin thư mục trong 1 thư mục

cuu duong than cong. com

# NỘI DUNG

- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

cuu duong than cong. com

# VÙNG FAT - 1

- Có 1 hay nhiều bảng FAT
- Mỗi bảng FAT:
  - Chứa thông tin về trạng thái của các cluster trong vùng dữ liệu
  - Định vị các tập tin: theo danh sách liên kết
  - FATxx: mỗi entry có xx bit và đại diện cho 1 cluster
    - Trống (FREE) : 0x000 - 0x0000 – 0x00000000
    - Hư (BAD): 0xFF7 – 0xFFF7 – 0xFFFFFFFF7
    - Đang dùng
      - Cluster kết thúc tập tin (EOF):
        - 0xFF8/0xFFF - 0xFFF8/0xFFFF – 0xFFFFFFFF8 - 0xFFFFFFFF
      - Cluster tương ứng chứa nội dung tập tin: chứa giá trị của cluster tiếp theo chứa nội dung tập tin (2 – 0xFE7)

# FAT

0000	...
0001	0000
0002	0003
0003	0004
0004	FFFF
0005	0006
0006	0008
0007	FFFF
0008	FFFF
0009	FFF7
	...

0000	0001	0002	0003	0004	0005
empty	empty	File1	File1	File1	File2
0006	0007	0008	0009	0010	0011
File2	File3	File2	bad	empty	empty
0012	0013	0014	0015	0016	0017
empty	empty	empty	empty	empty	empty

cuu duong than cong. com

# FAT

## ○ Nhận xét:

- FAT12: tối đa 4077 (2 - 0xFE7) clusters
- FAT16: tối đa 65517 (2 - 0xFFEF) clusters
- FAT32: nhiều hơn FFEE clusters

cuu duong than cong. com

cuu duong than cong. com

# FAT – TRUY XUẤT

## ○ Entry thứ k

- Chiếm  $xx$  bit =  $N$  bytes
- Bắt đầu tại vị trí  $I = k * N$
- Lưu ý: FAT12:
  - Đặt:
    - C\_B1, T\_B1: 4 bit cao và 4 bit thấp byte thứ 1
    - C\_B2, T\_B2: 4 bit cao và 4 bit thấp byte thứ  $I+1$
  - K chẵn:
    - Byte thứ 1: trọn byte
    - Byte thứ  $i+1$ : 4 bit thấp
    - Đọc:  $T\_B2 - C\_B1 - T\_B1$
  - K lẻ
    - Byte thứ 1: 4 bit cao
    - Byte thứ  $i+1$ : trọn byte
    - Đọc:  $C\_B2 - T\_B2 - C\_B1$

# FAT – VÍ DỤ FAT12

Byte	0	1	2	3	4	5	6	7	8	9
Giá trị	F0	FF	FF	03	40	00	FF	6F	00	07
Byte	10	11	12	13	14	15	...	...	...	...
Giá trị	80	00	09	A0	00	0B	C0	...	...	...

cuu duong than cong. com

Cluster **4** → I = 6

Byte i: **FF**

Byte i+1: **6F**

Giá trị cluster 4: FFF

Cluster **7** → I = 10

Byte i: **80**

Byte i+1: **00**

Giá trị cluster 7: 008

cuu duong than cong. com

Entry	0	1	2	3	4	5	6	7	8	.....
Giá trị	FF0	FFF	003	004	FFF	006	007	008	009	



# NỘI DUNG

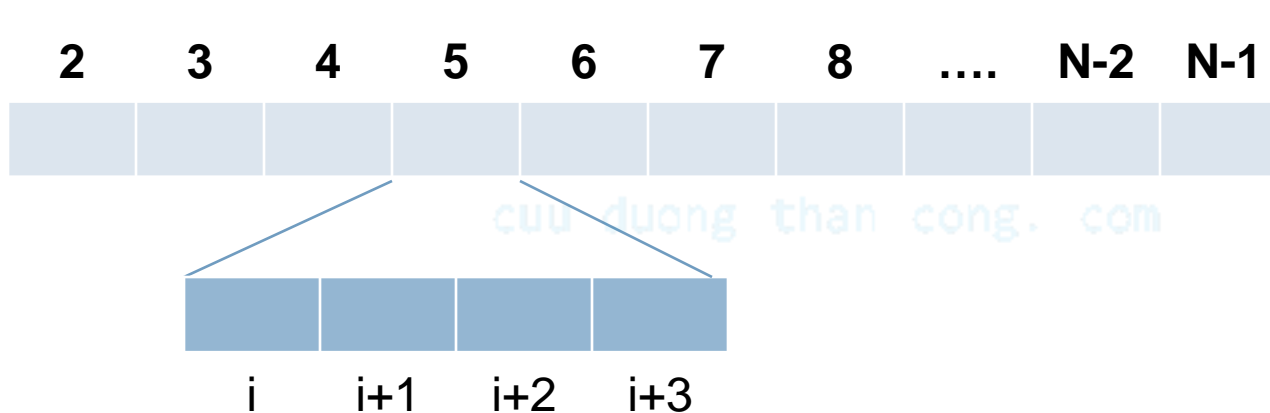
- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

# VÙNG DỮ LIỆU

- Tổ chức thành các khối dữ liệu, cluster
  - Có  $2^n$  sector
  - Đánh số từ 2 (Do cluster trống đánh là 0)
  - Cluster k trong vùng Data sẽ bắt đầu tại sector logic thứ i

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

$$S_{RDET} = N_{RDET} * 32 / 512$$



$$S_C = 4$$

$$S_B = 1$$

$$N_F = 2$$

$$S_F = 9$$

$$N_{RDET} = 224$$

$$K = 5 \rightarrow i = 45$$

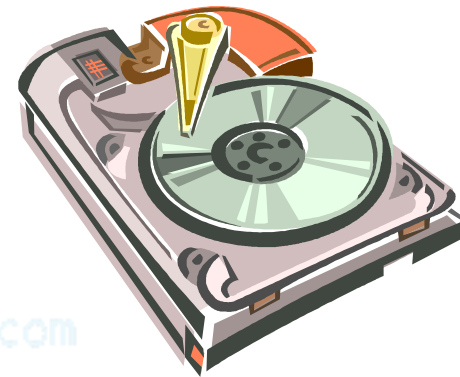
# NỘI DUNG

- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

# ĐẶT VẤN ĐỀ



FAT???



cuu duong than cong. com

# TÌM KIẾM TẬP TIN - SEARCH

- Xuất phát: RDET
- Duyệt qua từng entry trong RDET
- Duyệt qua các SDET (nếu có)

cuu duong than cong. com

cuu duong than cong. com

# ĐỌC NỘI DUNG TẬP TIN - TYPE

- Dựa trên RDET/SDET → tìm entry chính tập tin
- Dựa trên entry chính:
  - Sector bắt đầu + Kích thước
  - FAT: tìm danh sách các cluster chứa nội dung tập tin
  - DATA: đọc nội dung tập tin tương ứng từng cluster
    - Lưu ý: cluster cuối cùng chỉ đọc đúng số byte còn lại (vì có thể nó không chiếm trọn toàn bộ nội dung cluster cuối cùng)

# XEM NỘI DUNG MỘT THƯ MỤC - DIR

- Dựa trên RDET/SDET → tìm entry chính thư mục
- Dựa trên entry chính:
  - Sector bắt đầu + Kích thước
  - FAT: tìm danh sách các cluster chứa bảng SDET của thư mục
  - DATA: đọc bảng SDET của thư mục

# TẠO TẬP TIN – COPY CON

- Tính số entry cần lưu tên tập tin, N entry
- Tính số cluster để lưu nội dung tập tin, M cluster
- Xác định vị trí cần thêm (RDET hay SDET)
  - Kiểm tra có đủ vùng nhớ để lưu N entry?
- FAT: kiểm tra đủ cluster trống lưu tập tin? → danh sách các cluster lưu nội dung tập tin
- Thêm N entry vào RDET/SDET tương ứng
- FAT: cập nhật trạng thái các cluster lưu nội dung tập tin
- DATA: ghi nội dung tập tin vào các cluster tương ứng



# TẠO THƯ MỤC - MD

- Tính số entry cần lưu tên thư mục, N entry
- Số cluster để lưu bảng SDET, 1 cluster
- Xác định vị trí cần thêm (RDET hay SDET)
  - Kiểm tra có đủ vùng nhớ để lưu N entry?
- FAT: kiểm tra đủ cluster trống lưu SDET? ➔ cluster lưu SDET
- Thêm N entry vào RDET/SDET tương ứng
  - Trường thuộc tính của entry chính: bật bit Directory
- FAT: cập nhật trạng thái các cluster lưu SDET (EOF)
- DATA: ghi bảng SDET
  - Tạo 2 entry “.” và “..” vào 2 entry đầu tiên trong SDET

# ĐỔI TÊN TẬP TIN - REN

- Dựa trên RDET → tìm các entry liên quan tập tin
- RDET/SDET: Cập nhật thông tin tên trong các entry
  - Thêm entry phụ:
    - Tìm vị trí mới lưu thông tin
    - Hủy thông tin tại vị trí cũ (set trạng thái là E5)
  - Hủy bớt entry phụ
    - Hủy bớt các entry phụ dư

# DI CHUYỂN TẬP TIN - MOVE

- Dựa trên RDET/SDET → tìm entry chính tập tin
- RDET/SDET → xác định vị trí mới tập tin
- Kiểm tra đủ vùng nhớ trong vị trí mới để lưu hay không?
- Copy các entry liên quan tập tin đến vị trí mới
- Hủy các entry ở vị trí cũ

cuu duong than cong. com

# SAO CHÉP TẬP TIN - COPY

- Dựa trên RDET/SDET → tìm entry chính tập tin
  - Sector đầu tiên + kích thước
- RDET/SDET → xác định vị trí mới tập tin
- Kiểm tra đủ vùng nhớ trong vị trí mới để lưu hay không?
- FAT: kiểm tra đủ vùng nhớ để lưu tập tin không?
- Copy các entry ở vị trí cũ sang vị trí mới
- cập nhật FAT
- Copy từng cluster trong vùng data

# XÓA TẬP TIN - DELETE

- Dựa trên RDET/SDET → tìm các entry tập tin
  - Sector đầu tiên + kích thước
  - Hủy các entry này
- FAT: tìm các cluster chứa nội dung tập tin
  - Set trạng thái của các cluster về FREE

cuu duong than cong. com

# FORMAT

## ○ Quick format

- Bootsector: Giữ lại các thông số cũ của volume
- RDET: Xóa các entry trong RDET: gán về 0
- FAT: Cập nhật trạng thái các cluster là FREE

## ○ Full format

- Thiết lập các thông số hệ thống cho volume
- RDET: Xóa các entry trong RDET: gán về 0
- FAT: Cập nhật trạng thái các cluster là FREE
- **DATA: Xóa dữ liệu của tất cả các cluster**

cuu duong than cong. com

# NỘI DUNG

- Giới thiệu
- Bootsector
- RDET
- FAT
- Vùng dữ liệu
- Minh họa các thao tác trên tập tin
- Các vấn đề trên FAT

cuu duong than cong. com

# CÁC VẤN ĐỀ TRÊN FAT

- Lost cluster
- Particular link
- Circular link
- Cross link
- Fragment (phân mảnh)

cuu duong than cong. com

cuu duong than cong. com



# CÁC VẤN ĐỀ TRÊN FAT

- Giả sử: trong đĩa A ( $Sc = 1$ ) có 3 tập tin như sau:
  - File1.txt (1052 byte): nội dung tập tin lưu ở cluster 2, 3, 4
  - File2.txt (890 byte): nội dung tập tin lưu ở cluster 6, 7
  - File3.txt (500 byte): nội dung tập tin lưu ở cluster 9

0	cuu duong than	1 cong. com
2	003	004 3
4	FFF	0 5
6	007	FFF 7
8	0	FFF 9
...	...	...

**BẢNG FAT**

# LOST CLUSTER

- Là hiện tượng giá trị của 1 cluster trong FAT chỉ định cluster đó không trống, không hư nhưng không thuộc tập tin nào

0			1
2	003	004	3
4	FFF	020	5
6	007	FFF	7
8	0	FFF	9
...	...	....	...

**BẢNG FAT**

# PARTICULAR LINK

- Là hiện tượng giá trị của cluster kết thúc trong FAT có giá trị là 0 (thay vì EOF)

0			1
2	003	004	3
4	0	0	5
6	007	FFF	7
8	0	FFF	9
...	...	...	...

**BẢNG FAT**

# CIRCULAR LINK

- Là hiện tượng giá trị 1 cluster **cuối cùng** của tập tin chứa địa chỉ của 1 cluster nào đó đang chứa nội dung tập tin

0			1
2	003	004	3
4	FFF	0	5
6	007	006	7
8	0	FFF	9
...	...	...	...

**BẢNG FAT**

# CROSS LINK

- Là hiện tượng giá trị của 1 cluster trong FAT chứa địa chỉ của 1 cluster chứa nội dung của 1 tập tin khác

0			1
2	003	007	3
4	FFF	0	5
6	007	FFF	7
8	0	FFF	9
...	...	...	...

**BẢNG FAT**

# TỔNG KẾT

FAT			
<u>Developer</u>	<u>Microsoft</u>		
Full Name	File Allocation Table		
	(12-bit version)	(16-bit version)	(32-bit version)
Introduced	1980 ( <u>Seattle QDOS</u> )	November 1987, ( <u>Compaq</u> DOS 3.31)	August 1996 ( <u>Windows 95</u> OSR2)
<u>Partition identifier</u>	0x01 ( <u>MBR</u> )	0x04, 0x06, 0x0E ( <u>MBR</u> )	0x0B, 0x0C ( <u>MBR</u> ) EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 ( <u>GPT</u> )
Structures			
Directory contents	Table		
File allocation	<u>Linked List</u>		
Bad blocks	Cluster tagging		
Limits			
Max file size	4 <u>GB</u> minus 1 byte (or volume size if smaller)		
Max <u>cluster</u> count	4,077 (2 <sup>12</sup> -19)	65,517 (2 <sup>16</sup> -19)	268,435,437 (2 <sup>28</sup> -19)
Max filename size	<u>8.3 filename</u> , or 255 UTF-16 characters when using <u>LFN</u>		
Max volume size	32 <u>MB</u>	2 <u>GB</u> 4 <u>GB</u> with 64k clusters (not widely supported)	2 <u>TB</u> 8 <u>TB</u> (with 32KB clusters)
Features			
<u>Dates recorded</u>	Creation, modified, access (accuracy to day only) (Creation time and access date are only available when LFN support is enabled)		
Date range	<u>January 1, 1980</u> - <u>December 31, 2107</u>		
<u>Forks</u>	<u>Not natively</u>		
Attributes	Read-only, hidden, system, volume label, subdirectory, <u>archive</u>		
<u>Permissions</u>	No		
Transparent compression	Per-volume, <u>Stacker</u> , <u>DoubleSpace</u> , <u>DriveSpace</u>	No	
Transparent encryption	Per-volume only with <u>DR-DOS</u>		No