

Hệ thống tập tin

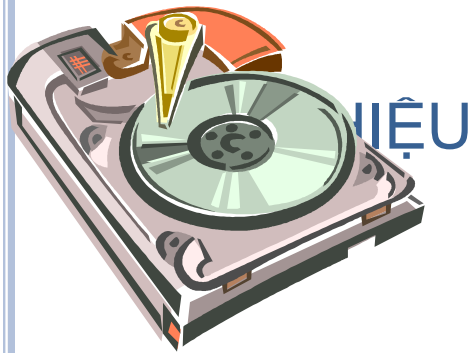
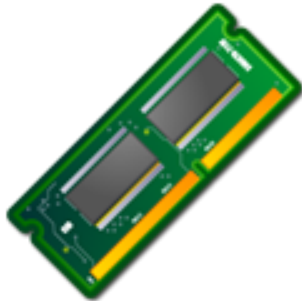
NỘI DUNG

- Giới thiệu
- Tập tin – Thư mục
- Đĩa từ
- Cài đặt hệ thống tập tin
- Minh họa một số hệ thống tập tin



Lưu trữ dữ liệu lớn

- Dữ liệu cần phải lưu lại sau khi kết thúc process
- Nhiều process có thể truy cập dữ liệu cùng lúc



Tìm kiếm thông tin???

Làm sao biết được block nào còn trống???

Quyền hạn???

HỆ THỐNG TẬP TIN

- cung cấp cơ chế
  - lưu trữ
  - truy cập dữ liệu và chương trình trên đĩa
- Đơn vị lưu trữ: tập tin (file)
  - Thư mục là 1 dạng tập tin đặc biệt
- Một số hệ thống tập tin hiện nay:
  - FAT: FAT12, FAT16, FAT32
  - NTFS
  - Ext2, ext
  - Vfat
  - ...

NỘI DUNG

- Giới thiệu
- Tập tin – Thư mục
- Đĩa từ
- Cài đặt hệ thống tập tin
- Minh họa một số hệ thống tập tin

TẬP TIN

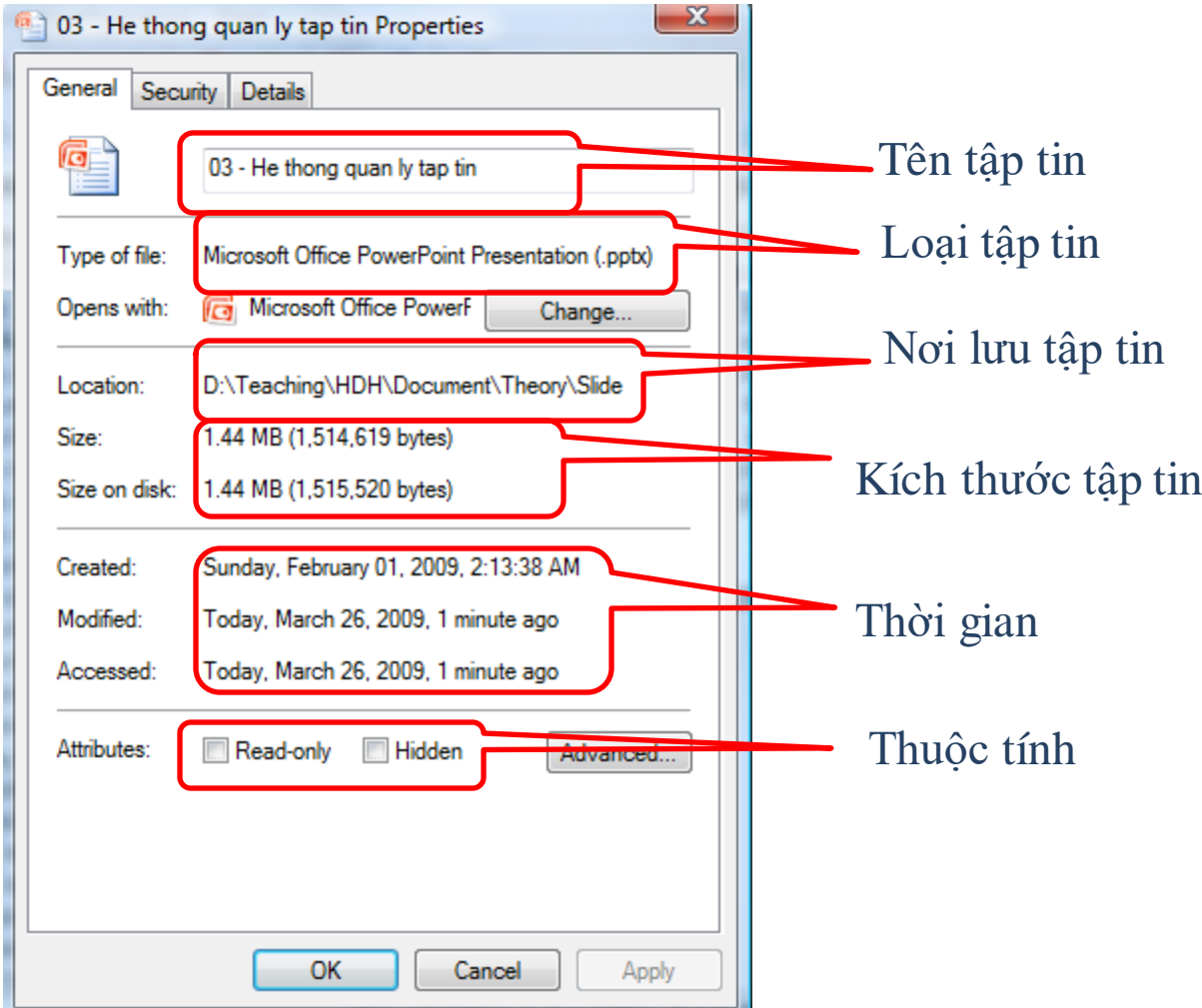
- Tập tin (file)
  - Tập các thông tin liên quan nhau
  - Được HĐH ánh xạ trên ổ đĩa vật lý
  - Gồm chuỗi các bit, byte, record, ...
  - Xác định bằng tên tập tin
    - Example.c

TẬP TIN – PHÂN LOẠI

Loại file	Ví dụ	Ý nghĩa
Thực thi	file.exe	File chứa mã lệnh dùng để load lên bộ nhớ và thực thi
Backup	File.bak	Backup file
Nguồn	File.c	File chứa mã nguồn gồm các dòng code, hàm,...
Đối tượng	File.o	File được tổ chức thành các khối được trình liên kết hiệu
Batch	File.sh File.bat	File chứa tập các lệnh
Thư viện	File.dll File.lib	File chứa thư viện các hàm để dùng cho các chương trình
Hình ảnh	File.jpg file.bmp	File hình ảnh được mã hóa bằng các chuẩn JPEG, RLE,...
Multimedia	File.mp3 File.wma File.rm	File âm thanh, video, ..
File text	File.txt	
Nén	File.zip	
Tài liệu	File.pdf File.doc	
.....		

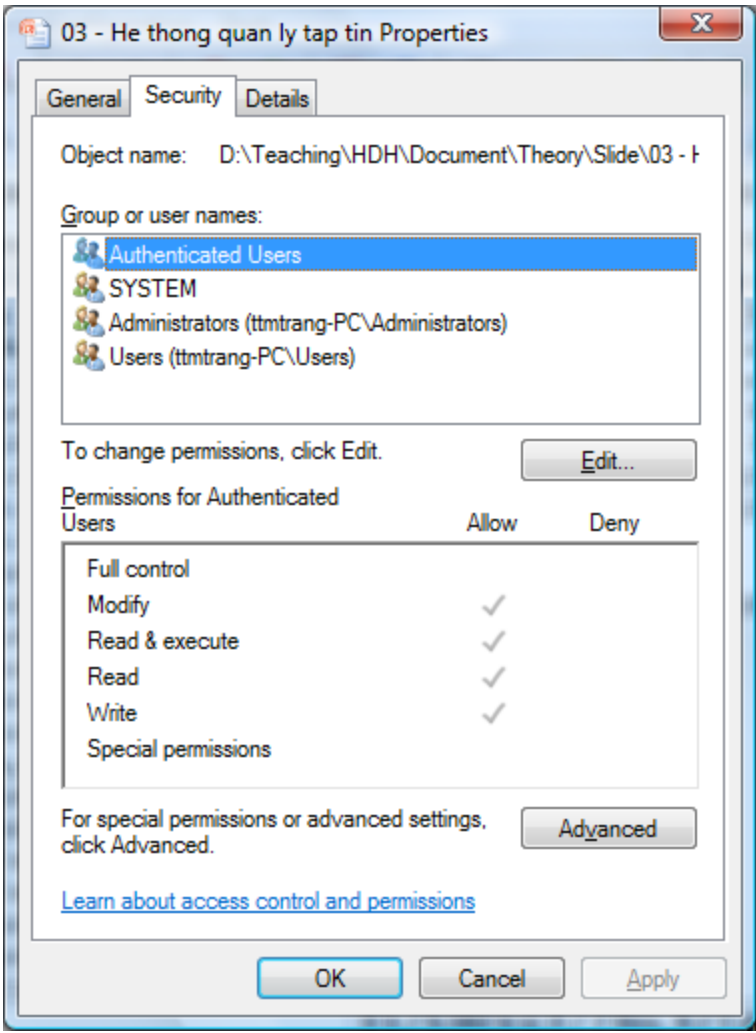
Một số loại file thông dụng

TẬP TIN – THUỘC TÍNH - 1



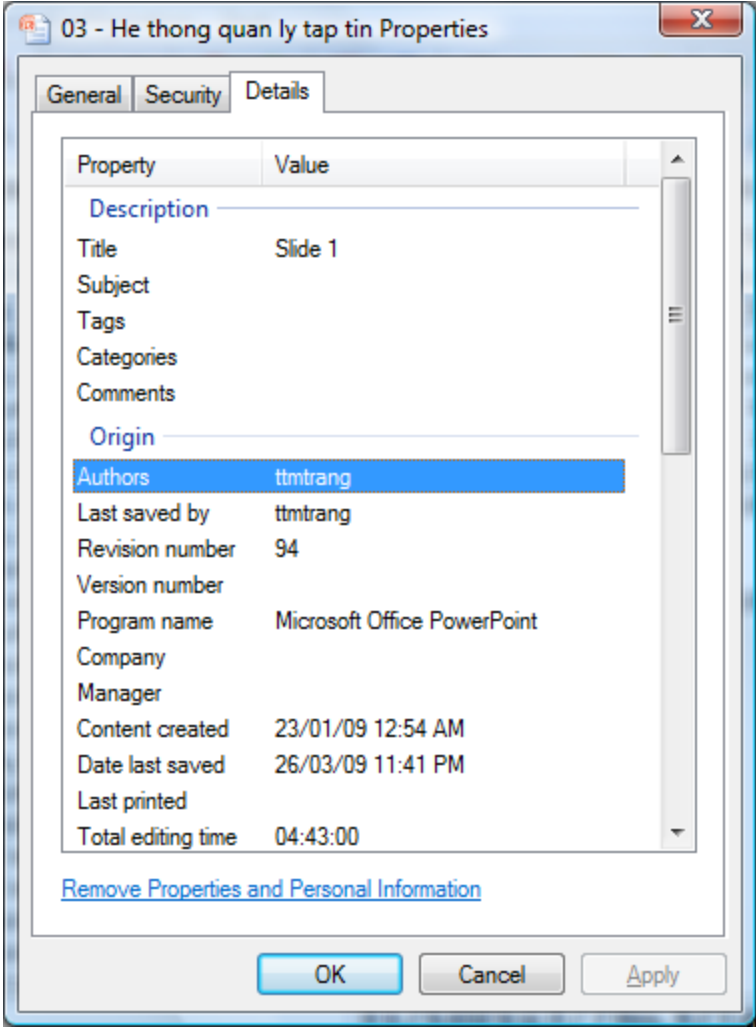
TẬP TIN – THUỘC TÍNH - 2

- Quyền hạn sử dụng

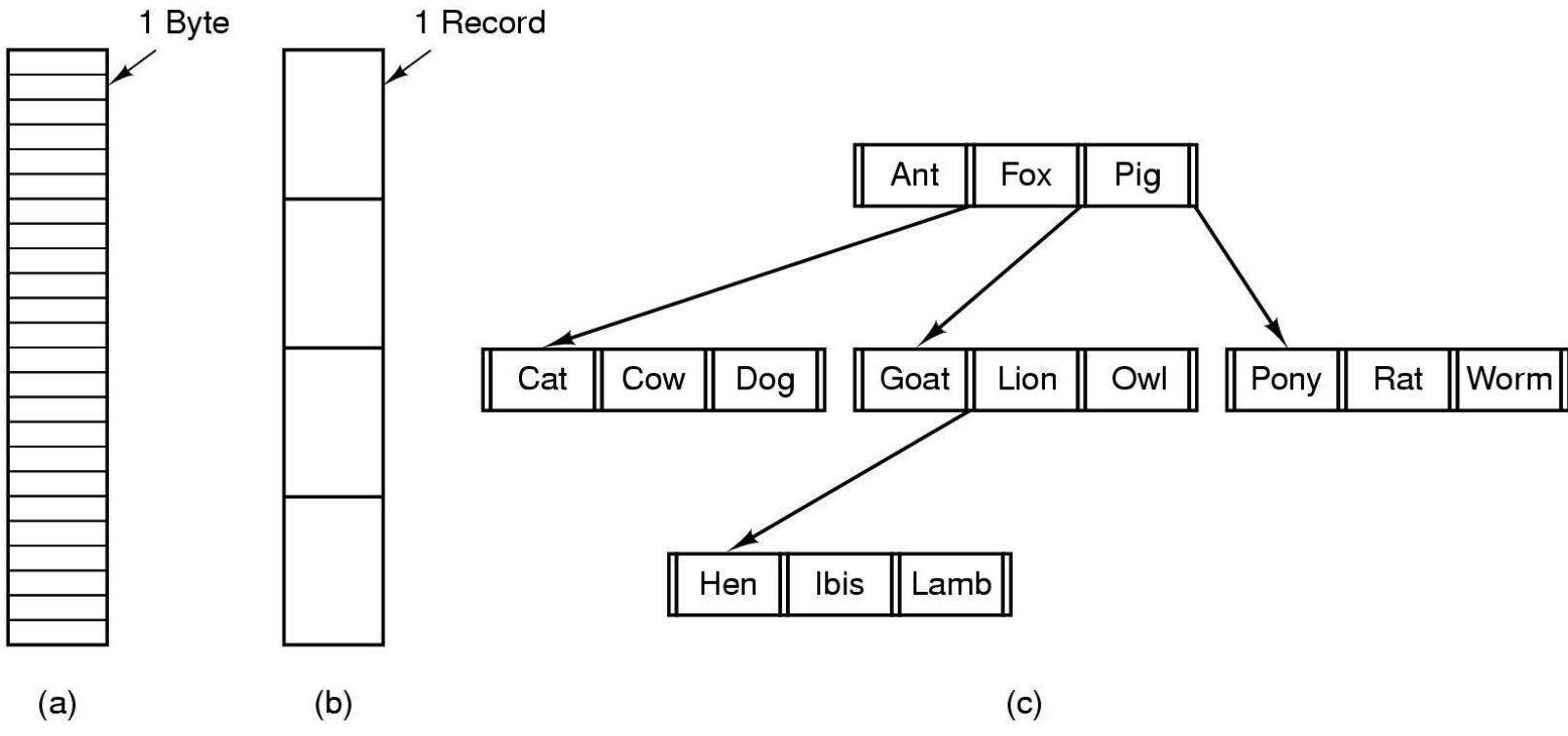


TẬP TIN – THUỘC TÍNH - 3

- Một số thông tin khác

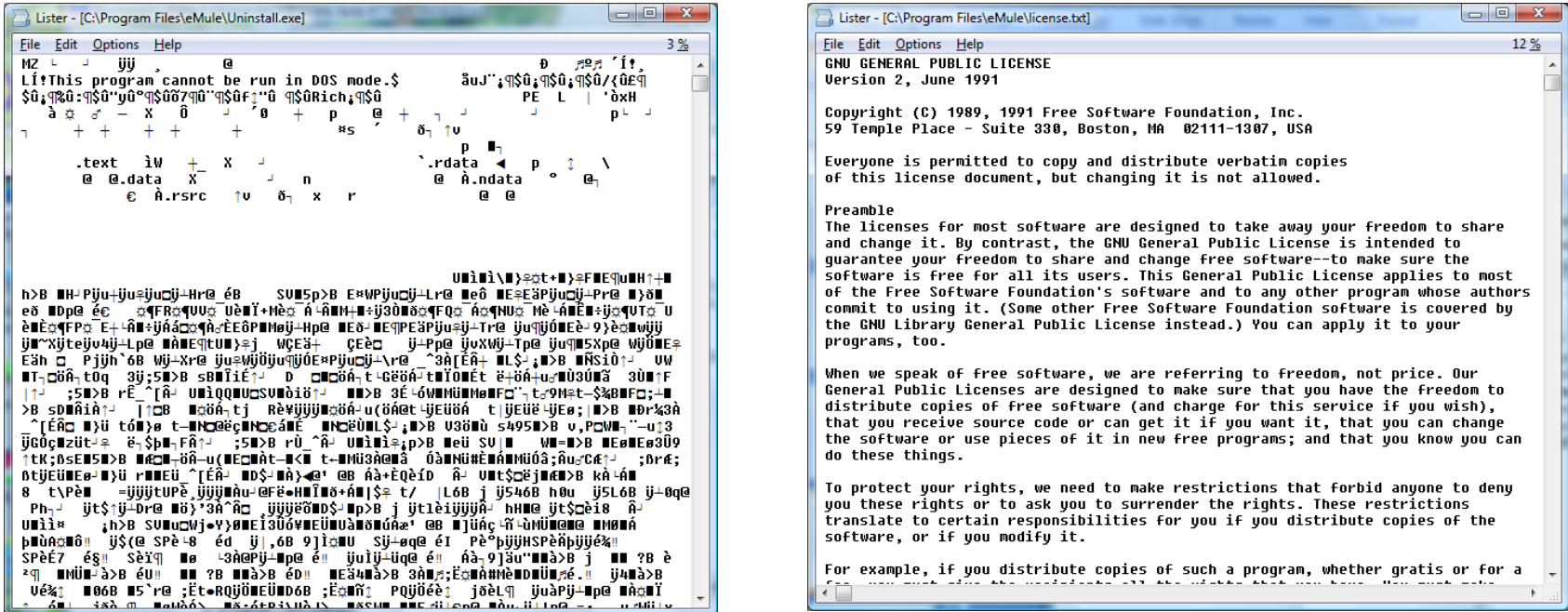


TẬP TIN – CẤU TRÚC - 1



(a) Chuỗi các bit, byte – file mã hóa  
(b) Tập các record – file danh sách sinh viên  
(c) Dạng cây - BTree

TẬP TIN – CẤU TRÚC - 2



(a) File nhị phân – (b) File text

TẬP TIN – THAO TÁC

1. Tạo – create
2. Ghi dữ liệu – write
3. Đọc dữ liệu – read
4. Xóa – delete
5. Mở - open
6. Đóng – close
7. Ghi thêm dữ liệu – append
8. Di chuyển đến 1 khối dữ liệu bất kỳ - seek
9. Đọc thuộc tính – get attr
10. Gán thuộc tính – set attr
11. Đổi tên – rename
12. Sao chép – copy
13. Tìm kiếm - search
14. Liệt kê – list, dir



TẬP TIN – PHƯƠNG PHÁP TRUY CẬP

- Giả thiết: có 1 tập tin lưu danh sách sinh viên
- Đặt vấn đề: cần đọc thông tin của sinh viên thứ N

Kích thước mỗi record	Giải quyết	Phương pháp
khác nhau	Phải đọc từ đầu	Truy cập tuần tự
Giống nhau	1. Tính vị trí logic lưu SV thứ N là p 2. Di chuyển đến vị trí p và đọc	Truy cập ngẫu nhiên
Khác nhau (Có 1 bảng lưu vị trí lưu mỗi SV)	1. Tra bảng 2. Di chuyển đến vị trí p và đọc	Truy cập index

NỘI DUNG

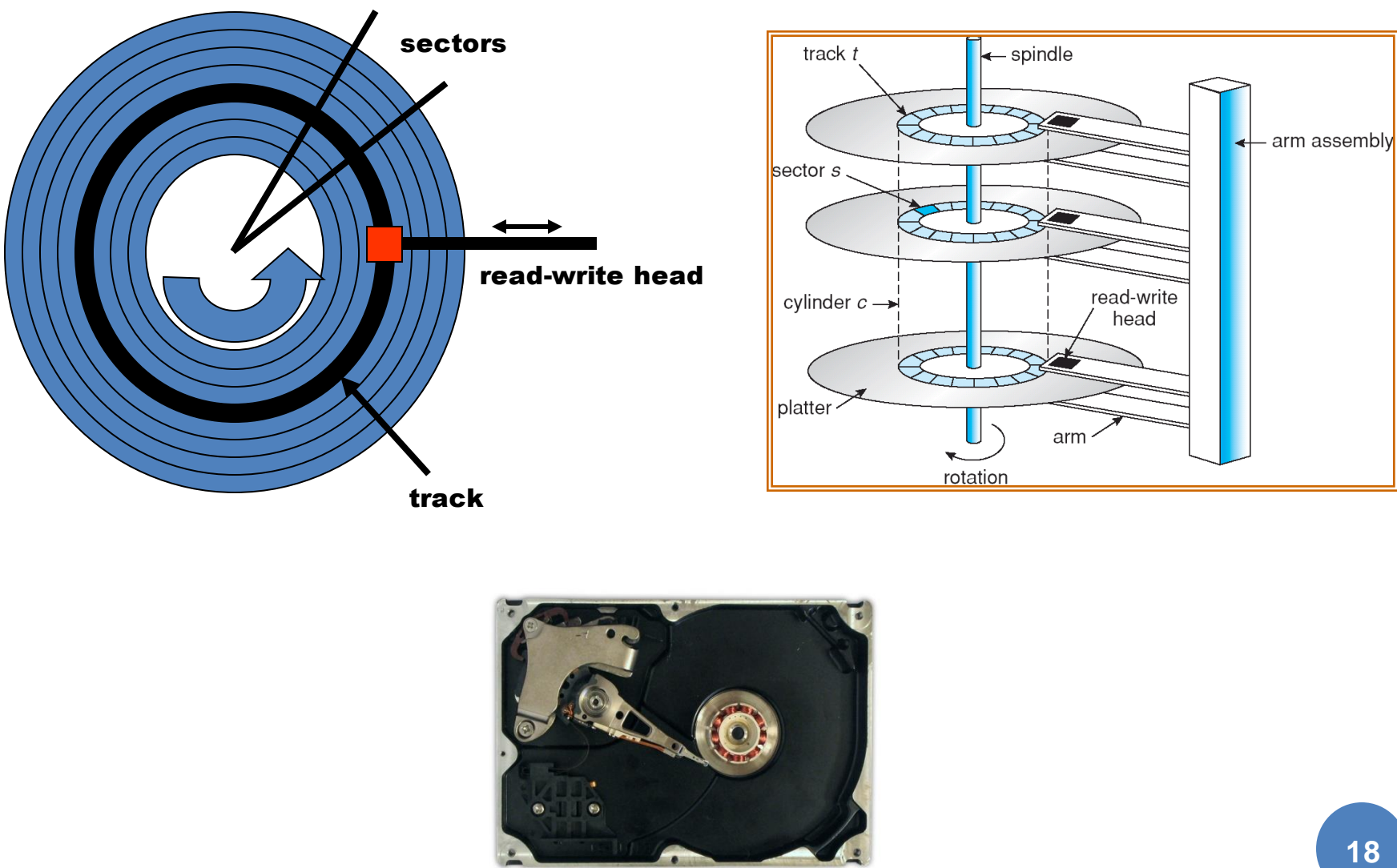
- Giới thiệu
- Tập tin – Thư mục
- Đĩa từ
- Cài đặt hệ thống tập tin
- Minh họa một số hệ thống tập tin



ĐĨA TỪ

- Tổ chức đĩa từ
- Thuật toán đọc đĩa
- Phân loại

ĐĨA TỪ - CẤU TRÚC - 1



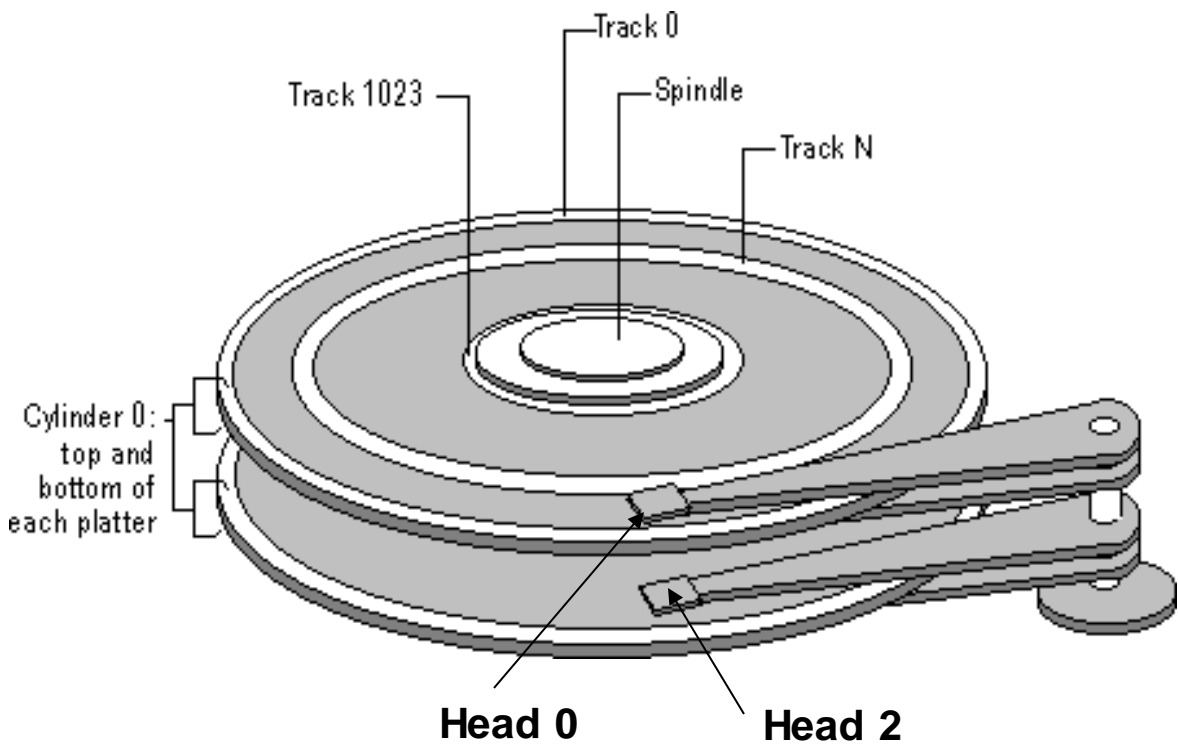
ĐĨA TỪ - CẤU TRÚC - 2

- Cấu trúc vật lý của đĩa từ:
  - Hình tròn, gồm nhiều mặt gọi là *head*.
  - Mỗi mặt có nhiều đường tròn đồng tâm gọi là *track*.
  - Trên các đường tròn (*track*) được chia thành các cung tròn gọi là *sector*.
  - Tập các *track* đồng tâm gọi là *cylinder*
  - Mỗi cung tròn chứa 4096 điểm từ (~ 4096 bit = 512 bytes).
  - Mỗi mặt có 1 đầu đọc để đọc ghi dữ liệu
  - Mỗi lần đọc/ghi ít nhất 1 cung tròn (512B).

ĐĨA TỪ - CẤU TRÚC - 3

- Vị trí của mỗi sector trong đĩa được thể hiện bằng 3 tham số : {sector, track, head}.
- Head được đánh số từ trên xuống bắt đầu từ 0.
- Track được đánh số từ ngoài vào bắt đầu từ 0.
- Sector được đánh số bắt đầu từ 1 theo chiều ngược với chiều quay của đĩa.
- Mỗi lần đọc ghi N sector

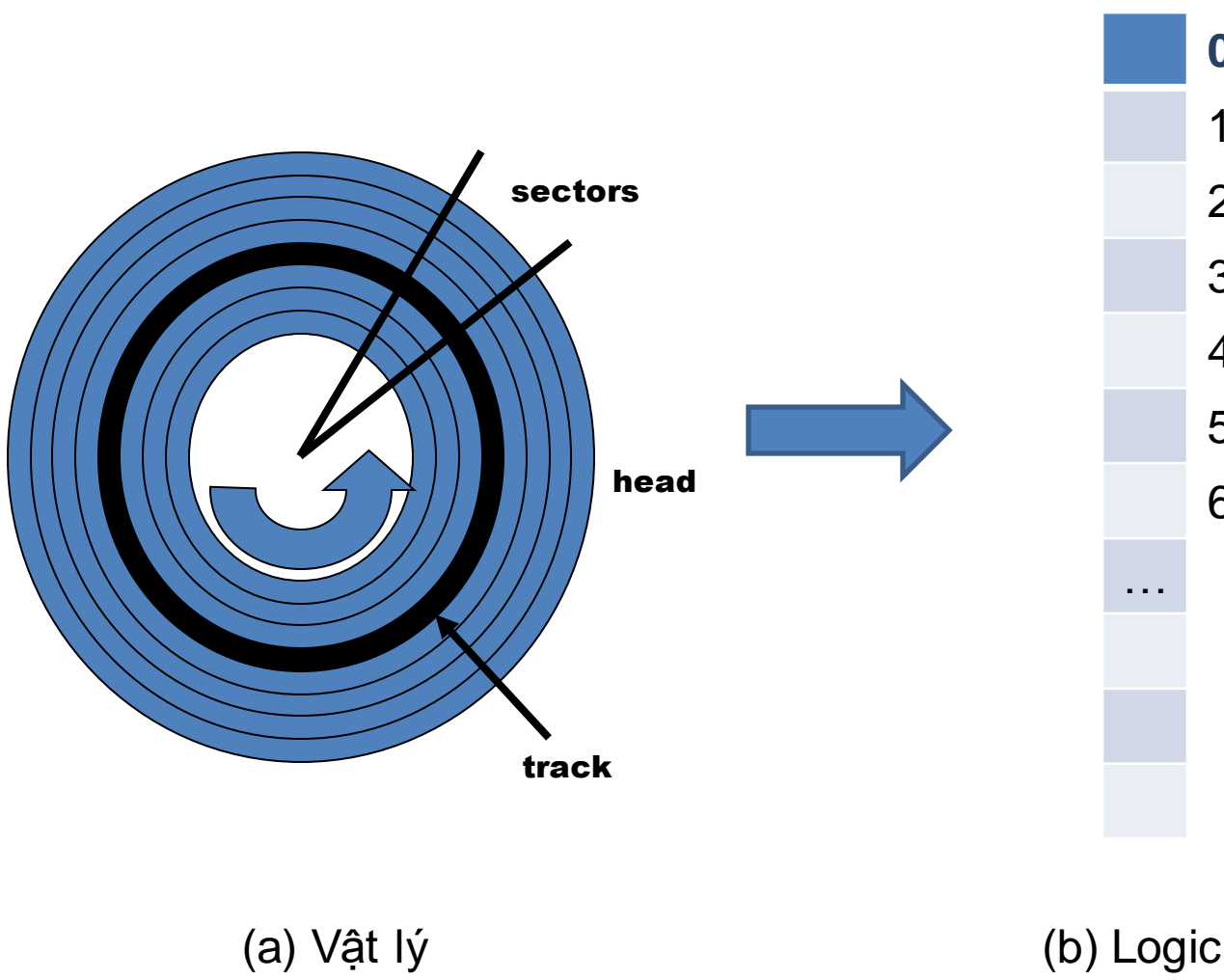
ĐĨA TỪ - CẤU TRÚC - 4



ĐĨA TỪ - DUNG LƯỢNG ĐĨA

- Kích thước đĩa phụ thuộc vào các yếu tố sau:
  - Số mặt từ, head
  - Số track trên mỗi mặt từ
  - Số sector trên mỗi track
  - Kích thước (byte) trên mỗi track.

ĐĨA TỪ - TỔ CHỨC ĐĨA - 1



ĐĨA TỪ - TỔ CHỨC ĐĨA - 2

- Các thông số trên đĩa mềm 1.44MB:
  - 2 head, 80 track/head, 18 sector/track.
  - Dung lượng đĩa = 2 head/disk \* 80 track/head \* 18 sector/track = 2880 sector/disk = 0.5 KB/sector \* 2880 sector/disk = 1440 KB/disk (~ 1.4MB)
  - Sector logic: 0 đến 2879 và tương ứng với các sector vật lý như sau:
    - Sector 0..17 tương ứng với sector vật lý (1,0,0)..(18,0,0)
    - Sector 18..35 tương ứng với sector vật lý (1,0,1)..(18,0,1)
    - ...
    - Sector 2879 tương ứng với sector vật lý (18,79,1).

Đĩa từ - Tổ chức đĩa - 3

- Đổi từ sector vật lý sang sector logic

$$I = t \cdot side \cdot st + h \cdot st + s - 1$$

$st$   
 $th$   
 $side$   
  
 $I$   
  
 $h$   
 $t$   
 $s$

: số sectors / track

: số tracks / side (head)

: số lượng side

: sector logic

: giá trị head

: giá trị track

: giá trị sector

25

Đĩa từ - Tổ chức đĩa - 4

- Đổi từ sector logic sang sector vật lý

$$s = (I \bmod st) + 1$$
$$t = I \div (st \cdot side)$$
$$h = (I \div st) \bmod side$$

$st$   
 $th$   
 $side$   
 $I$   
 $h$   
 $t$   
 $s$

: số sectors / track

: số tracks / side (head)

: số lượng head

: sector logic

: giá trị head

: giá trị track

: giá trị sector

26

Đĩa mềm 1.44 MB

- Có 2 head /disk, 80 track /head, 18 sector /track
- Dung lượng đĩa:  
2 head/disk \* 80 track/head \* 18 sector/track = 2880 sector/disk  
= 0.5 KB/sector \* 2880 sector/disk = 1440 KB/disk (~ 1.44 MB)
- Sector logic có chỉ số từ 0 đến 2879 và tương ứng với sector vật lý như sau:

Sector Logic	Sector vật lý (Sector, Track, Head)
0	(1, 0, 0)
1	(2, 0, 0)
...	...
17	(18, 0, 0)
18	(1, 0, 1)
19	(2, 0, 1)
...	...
35	(18, 0, 1)
36	(1, 1, 0)
37	(2, 1, 0)
...	...

27

Bài tập

1. Một đĩa cứng có 16 head, mỗi mặt có 684 track, và mỗi track có 18 sector thì sẽ có kích thước là bao nhiêu Megabyte ?
2. Cho biết sector vật lý (head 0, track 21, sector 6) tương ứng với sector logic nào trên đĩa mềm 1.44MB
3. Cho các chỉ số sector logic sau hãy cho biết chỉ số sector vật lý tương ứng:
  - a. 347
  - b. 348
  - c. 689
  - d. 690

$$I = t \cdot side \cdot st + h \cdot st + s - 1$$
$$s = (I \bmod st) + 1$$
$$t = I \div (st \cdot side)$$
$$h = (I \div st) \bmod side$$

28

Đĩa từ - Thuật toán đọc đĩa

- First-Come-First-Serve (FCFS)
- Shortest Seek Time First (SSTF)
- SCAN, C-SCAN
- Look, C-Look

29

FIRST COME FIRST SERVE - FCFS

- Phục vụ theo thứ tự yêu cầu
- Đơn giản nhưng không đáp ứng tốt dịch vụ

Các khối cần đọc (đầu đọc hiện tại tại vị trí 11):

24 8 21 7 2 14 12

12

14

2

7

21

8

24

1

5

10

15

20

25

1

5

10

15

20

25

30

SHORTEST SEEK TIME FIRST - SSTF

- Chọn nhu cầu gần với vị trí hiện hành nhất.

➔ Có nhiều yêu cầu chờ ..chờ...và chờ...

12

14

2

7

21

8

24

1

5

10

15

20

25

1

5

10

15

20

25

31

SCAN

- Di chuyển đầu đọc về 1 phía của đĩa đến block xa nhất sau đó di chuyển về phía kia.
- Còn gọi là thuật toán thang máy.

Các khối cần đọc (đầu đọc hiện tại tại vị trí 11):

12 14 2 7 21 8 24

12

14

2

7

21

8

24

1

5

10

15

20

25

1

5

10

15

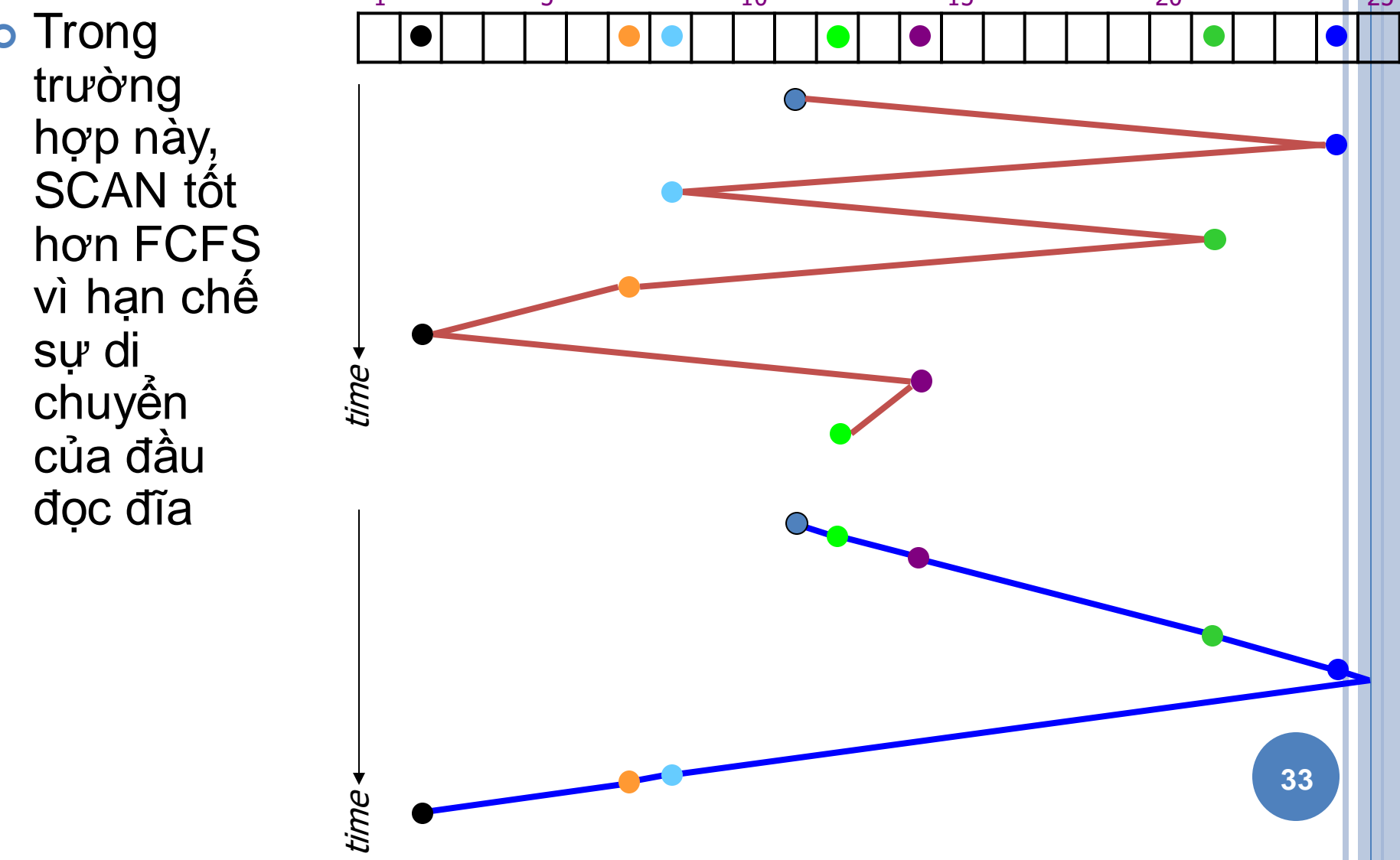
20

25

32



SCAN vs. FCFS

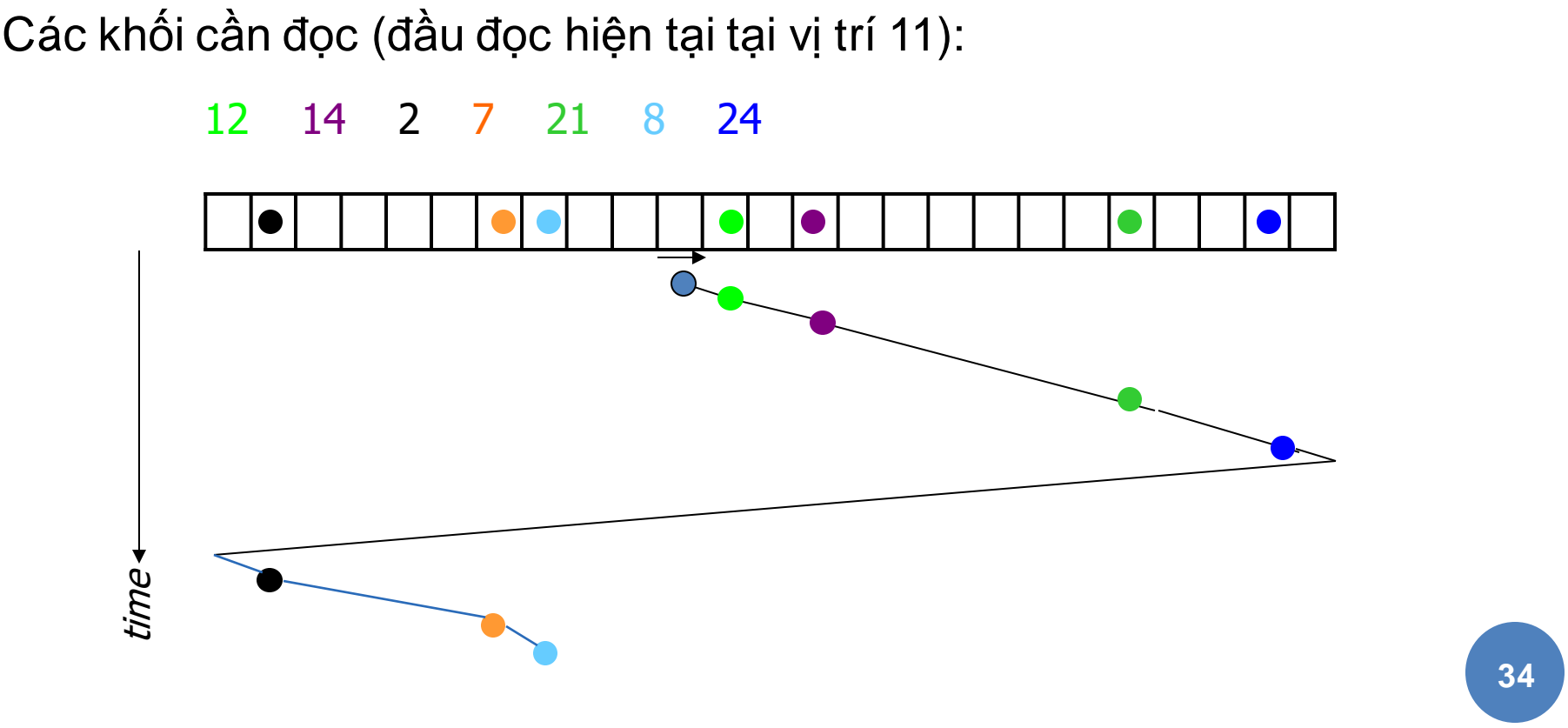


LOOK – C-LOOK

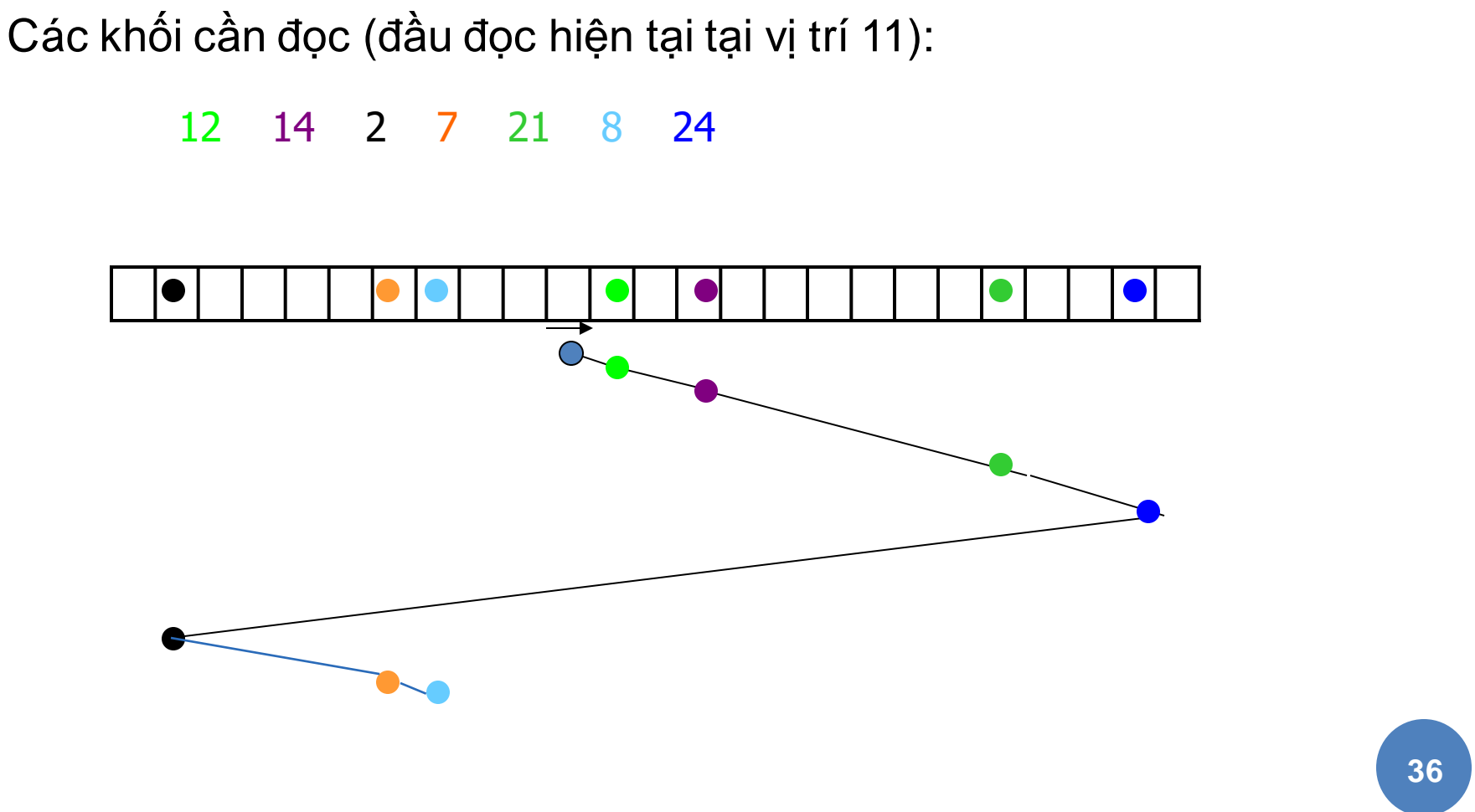
- Nhận xét:
  - Hai thuật toán lập lịch SCAN và C-SCAN luôn luôn di chuyển đầu đọc của đĩa từ đầu này sang đầu kia và di chuyển đến khối cuối cùng ở mỗi hướng.
- Nguyên tắc:
  - Giống SCAN và C-SCAN nhưng chỉ di chuyển đầu đọc đến khối xa nhất chứ không đến cuối.

C-SCAN

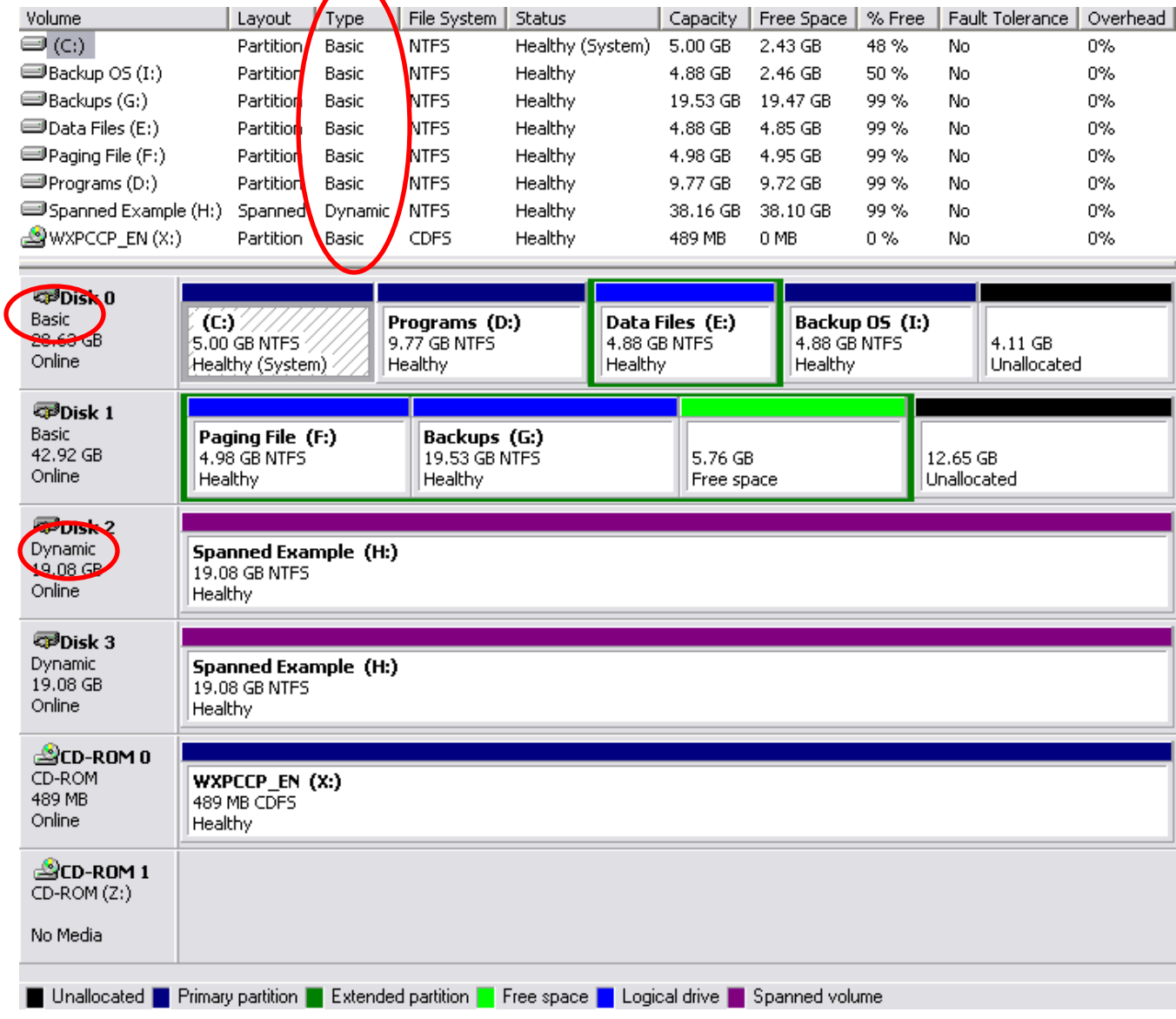
- Nguyên tắc:
  - Tương tự thuật toán SCAN.
  - Chỉ khác khi di chuyển đến 1 đầu của đĩa thì trở về vị trí bắt đầu của đĩa.



LOOK – C-LOOK

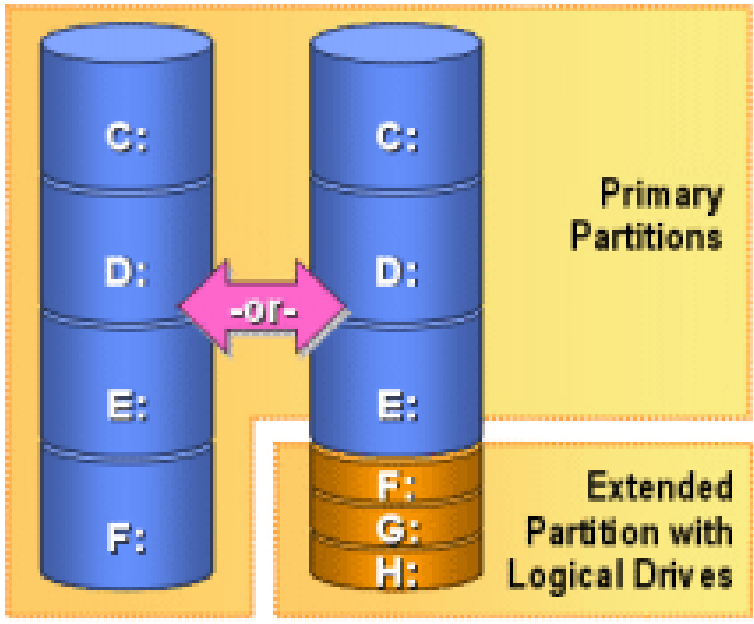


ĐĨA TỪ - PHÂN LOẠI



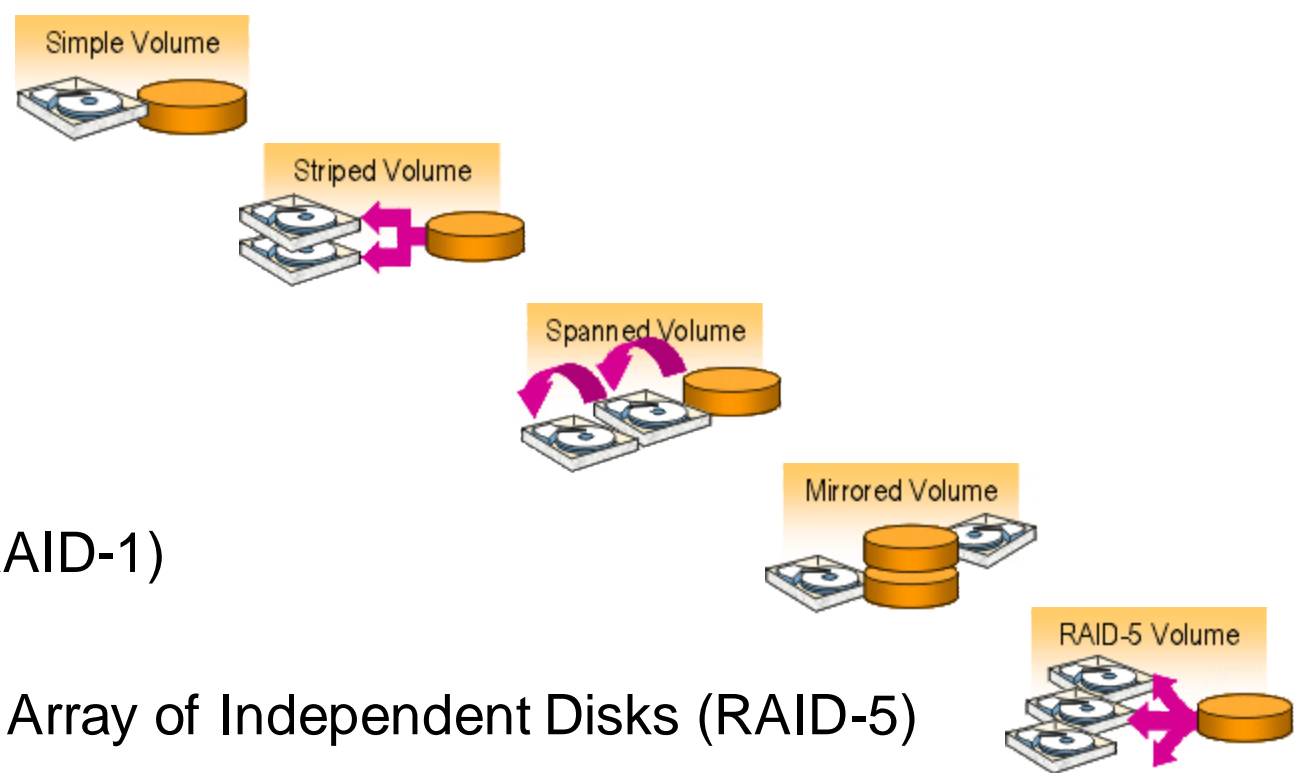
BASIC TYPE

- Tối đa 4 phân vùng (partition)
- Loại partition
  - Primary
    - Mỗi phân vùng: ấn định 1 ký tự
  - Extended
    - Có thể tạo nhiều logical drive, ấn định 1 ký tự cho mỗi logical drive

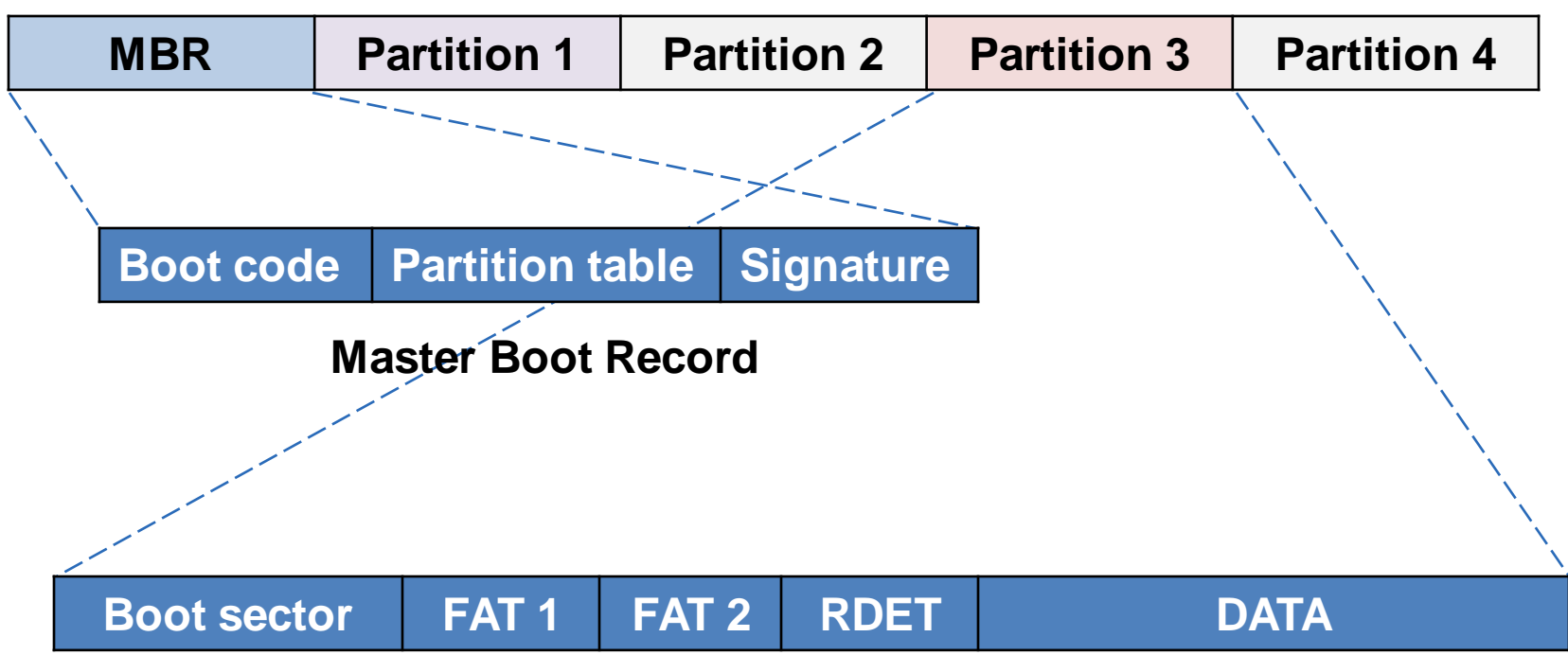


DYNAMIC TYPE

- Chia thành nhiều volume
  - Không bị giới hạn số lượng
- Loại Volume
  - Simple
  - Striped
  - Spanned
  - Mirrored (RAID-1)
  - Redundant Array of Independent Disks (RAID-5)



ĐĨA TỪ - CẤU TRÚC



MASTER BOOT RECORD

BẢNG MÔ TẢ CÁC PHÂN VÙNG - 1

Address			Description	Size in bytes
Hex	Oct	Dec		
0000	0000	0	Code Area	440 (max. 446)
01B8	0670	440	Optional Disk signature	4
01BC	0674	444	Usually Nulls; 0x0000	2
01BE	0676	446	Table of primary partitions (Four 16-byte entries, IBM Partition Table scheme)	64
01FE	0776	510	55h	MBR signature; 0xAA55 <sup>[1]</sup>
01FF	0777	511	AAh	
MBR, total size: 446 + 64 + 2 =				512

Nguồn: wikipedia

41

Offset	Field length (bytes)	Description
0x00	1	status <sup>[7]</sup> (0x80 = bootable, 0x00 = non-bootable, other = invalid <sup>[8]</sup> )
0x01	3	<b>CHS address</b> of first <b>block</b> in partition. <sup>[9]</sup> The format is described in the next 3 bytes.
0x01	1	head <sup>[10]</sup>
0x02	1	sector is in bits 5–0 <sup>[11]</sup> ; bits 9–8 of cylinder are in bits 7–6
0x03	1	bits 7–0 of cylinder <sup>[12]</sup>
0x04	1	<b>partition type</b> <sup>[13]</sup>
0x05	3	<b>CHS address</b> of last <b>block</b> in partition. <sup>[14]</sup> The format is described in the next 3 bytes.
0x05	1	head
0x06	1	sector is in bits 5–0; bits 9–8 of cylinder are in bits 7–6
0x07	1	bits 7–0 of cylinder
0x08	4	<b>LBA</b> of first sector in the partition
0x0C	4	number of <b>blocks</b> in partition, in little-endian format

Nguồn: wikipedia

42

BẢNG MÔ TẢ CÁC PHÂN VÙNG - 2

- Type:
  - 0x07 : Phân vùng chứa “Windows”
  - 0x83 : Phân vùng chứa “Linux”
  - 0x00 : Phân vùng không sử dụng.

Tham khảo thêm:  
[http://www.win.tue.nl/~aeb/partitions/partition\\_types-1.html](http://www.win.tue.nl/~aeb/partitions/partition_types-1.html)

43

MASTER BOOT RECORD – Ví dụ - 1

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000000000	90	E9	7D	01	FA	33	C0	8E	D0	8E	C0	8E	D8	BC	00	7C	é}.ú3ÁiDIAi0%.
0000000010	8B	F4	FB	BF	00	06	B9	00	01	F3	A5	BB	20	06	FF	E3	ôú...'.ô%»..ý3
0000000020	90	90	BE	7D	07	81	3C	AA	55	75	11	E8	58	00	73	0C	%}. <#Uu.èX.s.
0000000030	E8	65	00	72	07	E8	B1	00	72	3B	EB	2C	BE	7D	07	C7	èe.r.è.t.r;è.%.ç
0000000040	04	00	00	BA	80	00	BE	BE	07	B9	04	00	F6	04	80	75	...2  .34%.1...ô.1u
0000000050	07	83	C6	10	E2	F6	EB	1D	8A	74	01	8B	4C	02	BB	00	.. E.àòè.. t.. L.».
0000000060	7C	B8	01	02	CD	13	72	0D	81	3E	FE	7D	55	AA	75	05	.. .f.r. > }U#u.
0000000070	EA	00	7C	00	00	BE	6A	07	AC	0A	C0	74	FE	BB	07	00	è.. .%.j.-.Àt »..
0000000080	B4	0E	CD	10	EB	F2	BB	00	7E	C6	07	13	C6	47	01	00	..f.i.èò»..~E..EG..
0000000090	B2	80	B8	00	E0	CD	13	C3	BF	00	7E	BA	F0	01	B3	A0	2  .ài.Àú.~#è..3
00000000A0	E8	84	00	72	0C	B1	01	E8	48	00	72	05	E8	19	00	73	è r.t.èH.r.è..s
00000000B0	16	F6	C3	10	75	05	80	CB	10	EB	E5	81	FA	70	01	74	ô.À.u. E.è.à ú.p.t
00000000C0	05	BA	70	01	EB	D8	F9	C3	81	BD	FE	01	55	AA	75	17	2p.è0u.Ài% p.U#
00000000D0	8B	75	02	81	FE	BE	01	77	0E	03	F7	81	3C	AA	55	75	tu. p%.v..+ <#Uu
00000000E0	06	F6	44	02	01	75	01	F9	C3	BF	00	7C	B1	0A	E8	01	..GD..u.ú.Àú.. t.è.
00000000F0	00	C3	52	57	83	C2	02	B0	01	EE	42	8A	C1	EE	42	32	..ÄRVi.Ä..iBiÄiB2
0000000100	C0	EE	42	EE	42	8A	C3	EE	42	B0	20	EE	E8	33	00	EC	ÄiBiBiÄiB* iè3.i
0000000110	24	FD	3C	58	75	0D	83	EA	07	B9	00	01	FA	F3	6D	FB	ý<Xu. è..1..úómú
0000000120	F8	EB	01	F9	5F	5A	C3	52	83	C2	07	EC	A8	80	75	0F	èè.ù.ZÄRiÄ.i'1u.
0000000130	4A	8A	C3	EE	42	EC	24	D0	3C	50	75	03	F8	EB	01	F9	JiÄiBi\$D<Pu.èè.ù
0000000140	5A	C3	51	8B	0E	6C	04	83	C1	12	81	C2	FF	01	EC	8A	ZÄQI..l.Ä..iÄy.i
0000000150	E0	80	E4	D8	80	FC	58	74	06	3B	0E	6C	04	75	EF	81	à a0 úXt...l..ui
0000000160	EA	FF	01	B9	00	20	E2	FE	59	C3	0D	0A	45	72	72	6F	èy..'.äpYÄ..Erro
0000000170	72	20	4C	6F	61	64	69	6E	67	20	4F	53	00	55	AA	00	r Loading OS.U#
0000000180	00	E9	80	FE	00	00	00	00	68	00	00	00	00	00	00	00	..é p...h.....
0000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000001B0	00	00	00	00	00	00	00	00	00	0A	01	0A	00	00	80	01	.....
00000001C0	01	00	0C	FE	FF	FF	3F	00	00	00	C1	52	A8	04	00	00	...pýý?...ÄR'...
00000001D0	C1	FF	0F	FE	FF	FF	00	53	A8	04	C1	91	A8	04	00	00	Äy.pýý.S'..Ä'...
00000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	.....U#

44

MASTER BOOT RECORD – Ví dụ - 2

Parti tion	Starting			Ending			Status	Type	First sector	Sector#
	H	T	S	H	T	S				
1	1	0	1	254	1023	63	0x80	0x0C	63	78140097
2	0	1023	1	254	1023	63	0x00	0x0F	78140160	78156225

45

QUÁ TRÌNH BOOT HỆ THỐNG

- POST (Power-On-Self-Test)
- Tải MBR để đọc thông tin bảng phân vùng.
- Tìm phân vùng “active”.
- Chuyển quyền điều khiển về cho đoạn mã chương trình nằm trong Boot Record của phân vùng “active”
- Tải HDH tại phân vùng “active”.



Power on  
Reboot

Bootstrap

OS

Khởi tạo hệ thống

CPU, device controller, main memory, load đoạn code khởi động hdh

46

NỘI DUNG

- Giới thiệu
- Tập tin – Thư mục
- Đĩa từ
- Cài đặt hệ thống tập tin
- Minh họa một số hệ thống tập tin

47

CÀI ĐẶT HỆ THỐNG TẬP TIN

- Hệ thống tập tin chứa thông tin gì?
- Thành phần
- Phương pháp cấp pháp vùng nhớ
- Quản lý không gian đĩa trống

48



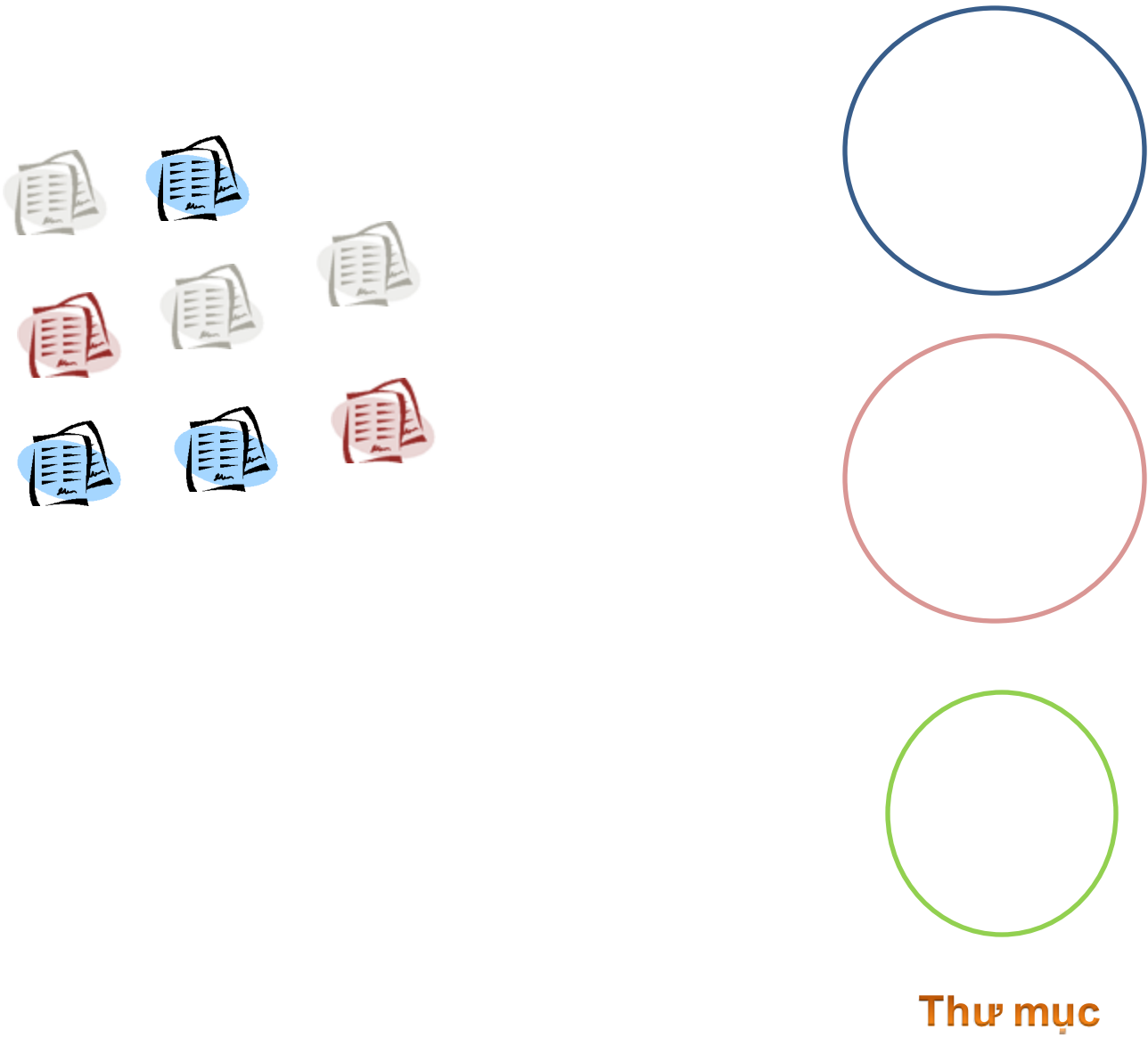
CÀI ĐẶT HỆ THỐNG TẬP TIN

- Hệ thống tập tin chứa:
  - Cách boot hệ điều hành
  - Tổng số block
  - Block trống
  - Cấu trúc cây thư mục (thư mục, tập tin)

CÀI ĐẶT HỆ THỐNG TẬP TIN

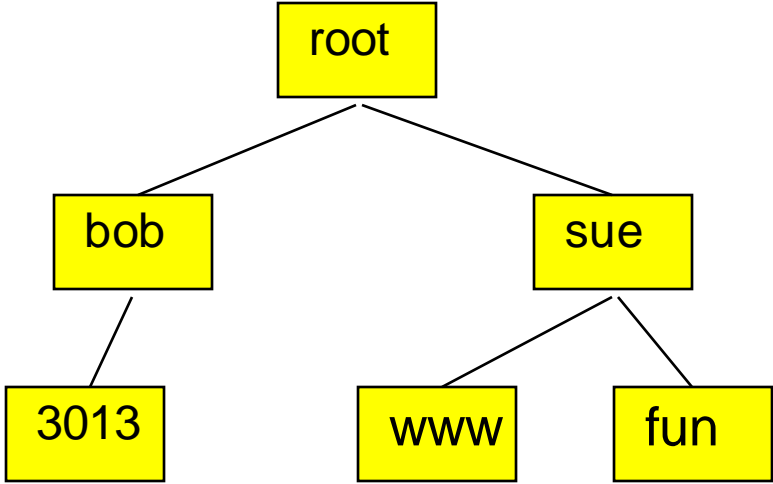
- Các thành phần trong hệ thống tập tin
  - Boot control block
    - Thông tin để boot hữh từ volume này
    - UFS: boot block, NTFS: partition boot sector, FAT: boot sector
  - Volume control block
    - Thông tin chi tiết volume
    - UFS: superblock, NTFS: master file table, FAT: boot sector
  - File control block
    - Tổ chức: tổ chức các tập tin ntn?
    - File: thông tin chi tiết của 1 tập tin
    - UFS: inode, NTFS: master file table, FAT: FAT&RDET

CÀI ĐẶT HỆ THỐNG TẬP TIN



THƯ MỤC - 1

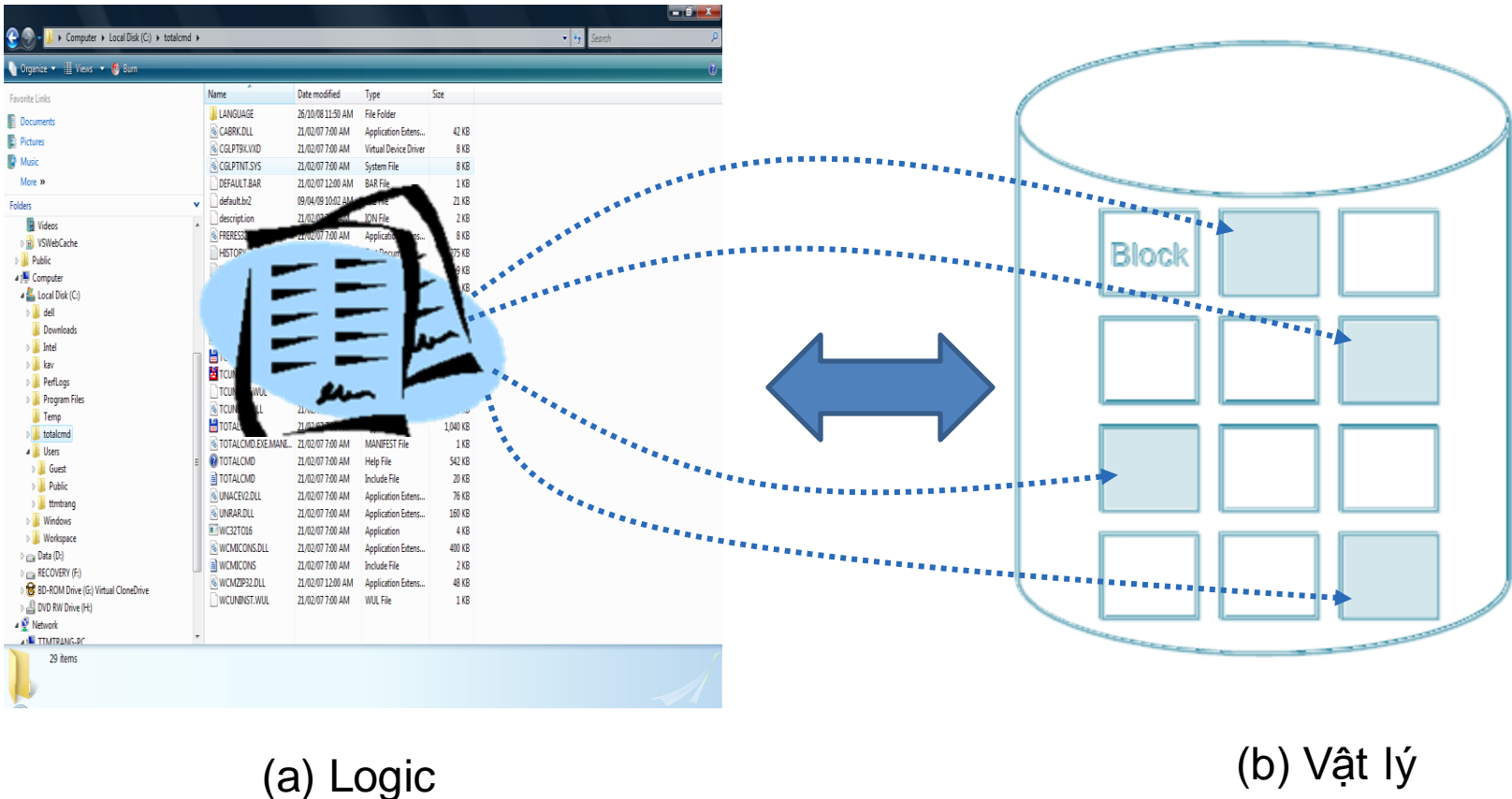
- Là một *tập tin* đặc biệt
- Giúp cho việc quản lý các tập tin dễ dàng hơn.
  - Gom nhóm các tập tin vào trong các thư mục theo ý nghĩa và mục đích sử dụng của người dùng.
  - Giúp định vị các tập tin 1 cách nhanh chóng.



THƯ MỤC - ĐƯỜNG DẪN (PATH)

- Dùng để xác định vị trí lưu tập tin khi hệ thống được tổ chức thành cây thư mục:
  - Đường dẫn tuyệt đối:
    - Ví dụ: “C:\Downloads\software\baigiang.doc”
  - Đường dẫn tương đối:
    - Ví dụ: “software\baigiang.doc” nếu thư mục hiện hành là “C:\Downloads\”
- Các thư mục đặc biệt:
  - Thư mục hiện hành (.)
  - Thư mục cha (..)

CÀI ĐẶT HỆ THỐNG TẬP TIN



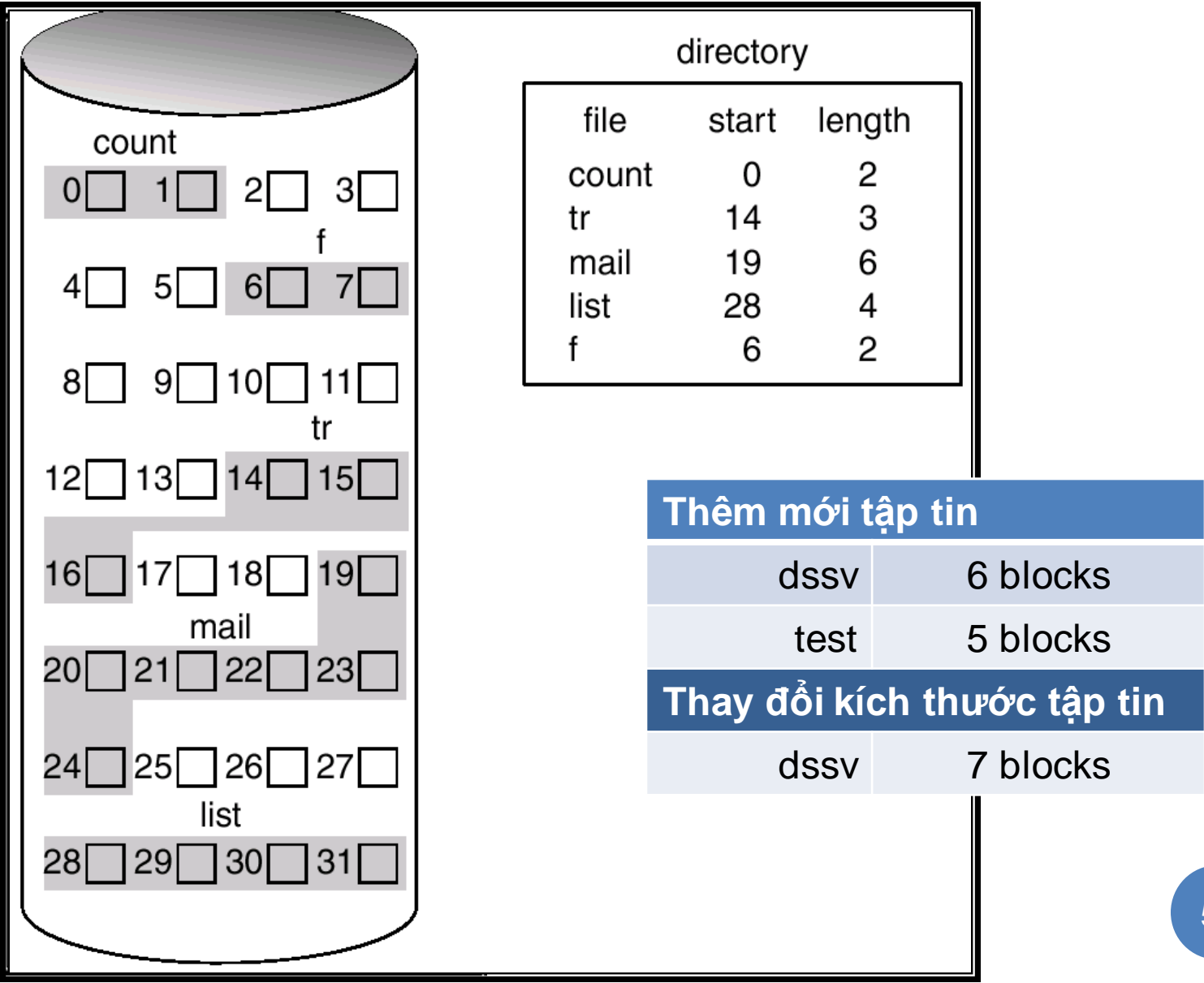
CÀI ĐẶT HỆ THỐNG TẬP TIN

- Làm sao map giữa cây thư mục với các block trên thiết bị lưu trữ
  - Mỗi tập tin lưu:
    - Lưu ở block nào?
    - Khi tạo mới, sử dụng block nào?
- ⇒ cấp phát???

CẤP PHÁT VÙNG NHỚ CHỨA TẬP TIN

- Phương pháp cấp phát:
  - Là cách thức cấp phát vùng nhớ (block) cho tập tin
- Phương pháp:
  - Cấp phát liên tục
  - Cấp phát bằng danh sách liên kết
  - Cấp phát bằng chỉ mục

CẤP PHÁT LIÊN TỤC - 1



57

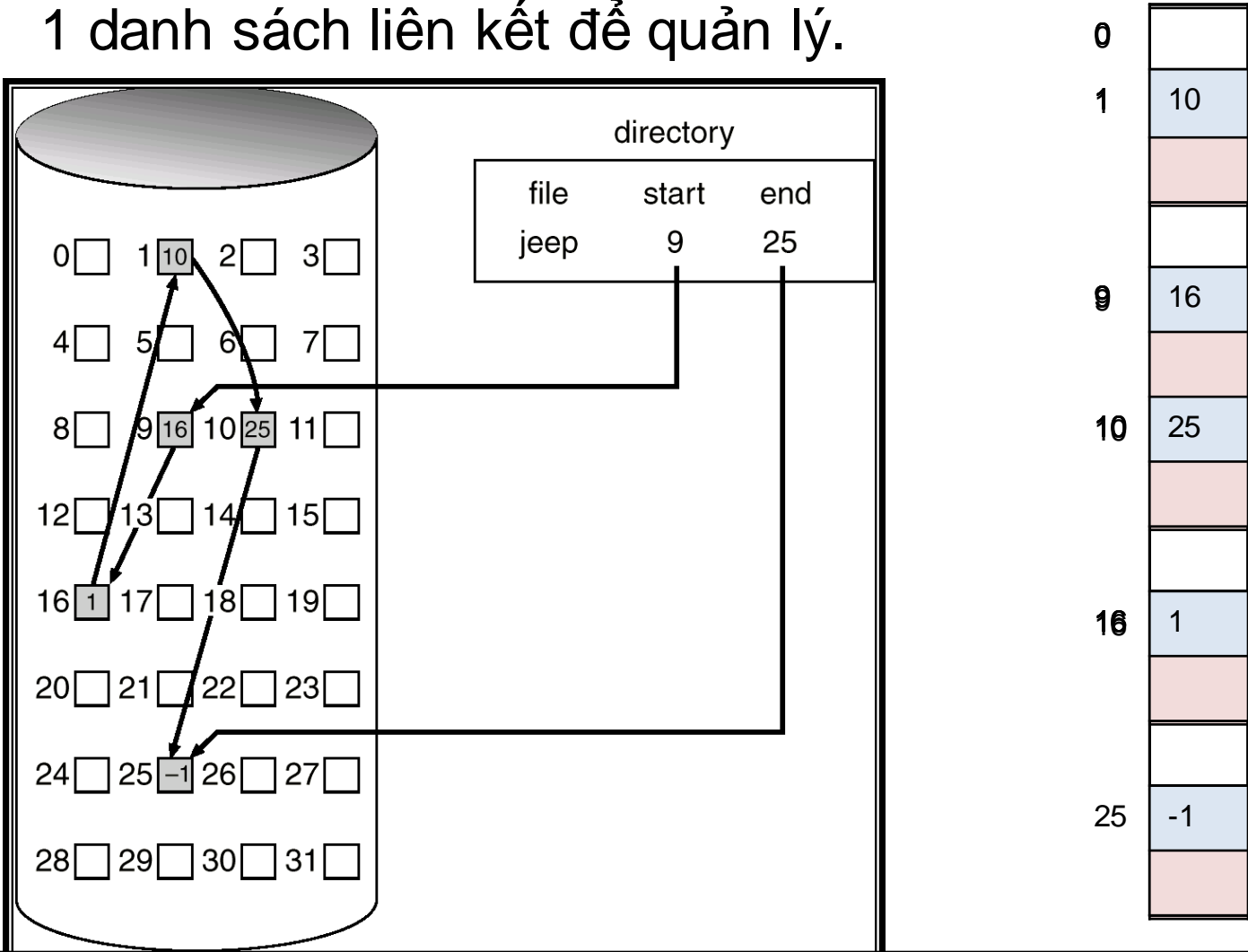
CẤP PHÁT LIÊN TỤC - 2

- Cấp phát 1 số block liên tục trên đĩa để lưu trữ nội dung tập tin
- Nhận xét:
  - Đơn giản: chỉ cần quản lý số hiệu khối bắt đầu và tổng số block chiếm bởi tập tin.
  - Truy cập nội dung tập tin nhanh chóng vì các block nằm kề nhau.
  - Gây lãng phí bộ nhớ.
  - Khó khăn khi tập tin mở rộng kích thước.

58

CẤP PHÁT BẰNG DANH SÁCH LIÊN KẾT - 1

- Nội dung tập tin được lưu trữ ở những block không cần liên tục. Các block này được xâu chuỗi tạo thành 1 danh sách liên kết để quản lý.



59

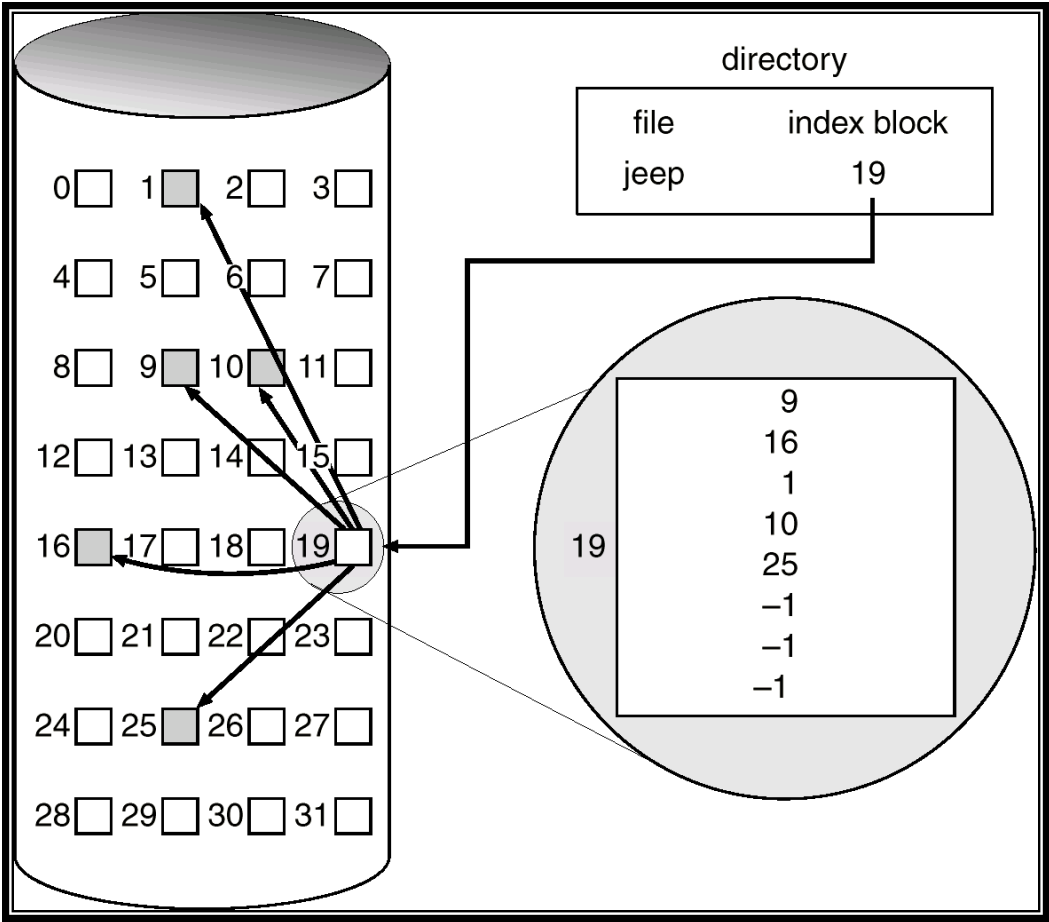
CẤP PHÁT BẰNG DANH SÁCH LIÊN KẾT - 2

- Nhận xét:
  - Đơn giản: Chỉ cần quản lý block bắt đầu.
  - Tận dụng hiệu quả không gian đĩa.
  - Truy cập tập tin lâu hơn vì đầu đọc phải di chuyển nhiều giữa các khối không liên tiếp.
  - Không thể truy cập ngẫu nhiên
  - Khối dữ liệu bị thu hẹp lại vì mỗi khối phải dùng 1 phần để lưu phần liên kết đến khối kế tiếp.

60

CẤP PHÁT BẰNG CHỈ MỤC (INDEX) - 1

- Mỗi tập tin:
  - Index block: Lưu địa chỉ các block của 1 tập tin bằng 1 mảng



61

CẤP PHÁT BẰNG CHỈ MỤC (INDEX) - 2

- Nhận xét:
  - Truy cập ngẫu nhiên
  - Tốn không gian lưu bảng chỉ mục

Theo danh sách liên kết	Theo chỉ mục
Truy cập tuần tự	Truy cập ngẫu nhiên
Tốn n*entry lưu địa chỉ block	Tốn 1 block
Không bị giới hạn kích thước	Giới hạn

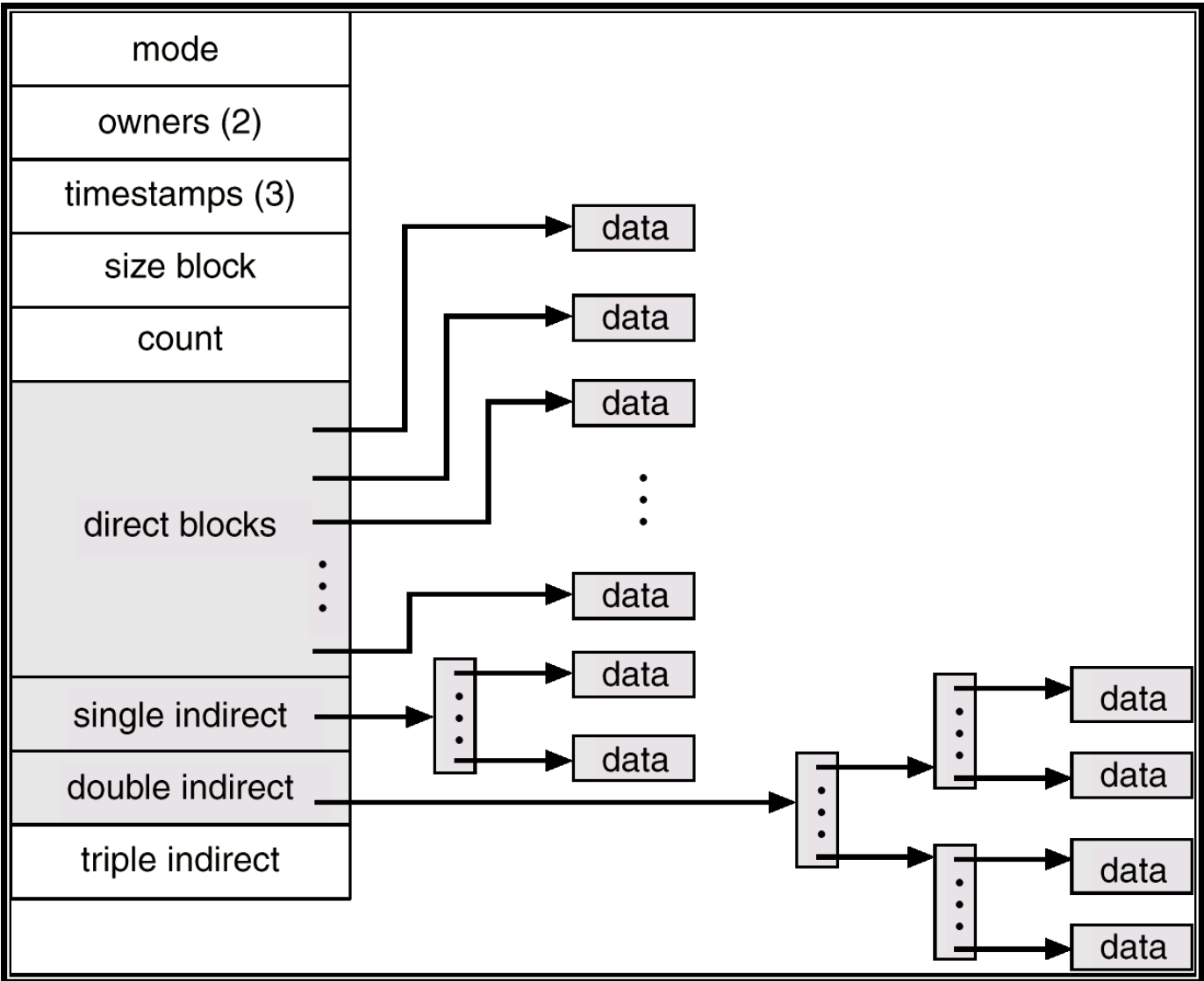
62

CẤP PHÁT BẰNG CHỈ MỤC (INDEX) - 3

- Phương pháp mở rộng:
  - Chỉ mục kết hợp với danh sách liên kết
    - Liên kết nhiều index block để lưu file lớn
    - VD: dùng entry cuối lưu địa chỉ của block index tiếp theo
  - Chỉ mục đa cấp
    - Index block cấp 1 lưu danh sách các index block 2, ....
    - VD: với 2 cấp, mỗi block có 1024 entry → quản lý ??? block
  - Chỉ mục kết hợp
    - Sử dụng N entry
      - N-3 entry đầu lưu địa chỉ của các data block
      - Entry kế tiếp lưu index block cấp 1
      - Entry kế tiếp lưu index block cấp 2
      - Entry kế tiếp lưu index block cấp 3
    - VD: I-node

63

CẤP PHÁT BẰNG I-NODE



64



QUẢN LÝ KHÔNG GIAN ĐĨA TRỐNG

- Ghi nhận danh sách các block trống
- Phương pháp:
  - Bit vector
  - Danh sách liên kết (linked list)
  - Nhóm (grouping)
  - Đếm (counting)
  - Bảng đồ không gian (space maps)

BIT VECTOR

- Mỗi block được đại diện bằng 1 bit
  - 0: chứa dữ liệu
  - 1: trống
- Nhận xét:
  - Đơn giản
  - Tính toán nhanh
  - Khi cần kiểm tra ➔ load bit vector lên bộ nhớ

LINKED – GROUPING - COUNTING

- Linked
  - Các block trống liên kết với nhau: block trống thứ N lưu địa chỉ của block trống thứ N+1
  - Chỉ cần lưu địa chỉ block trống đầu tiên
- Grouping
  - Tương tự linked
  - Lưu địa chỉ của N block trống tiếp theo
- Counting:
  - Với mỗi N block trống liên tiếp: lưu địa chỉ của block trống đầu tiên và số lượng block trống

NỘI DUNG

- Giới thiệu
- Tập tin – Thư mục
- Địa từ
- Cài đặt hệ thống tập tin
- Minh họa một số hệ thống tập tin

MỘT SỐ HỆ THỐNG TẬP TIN

- FAT
- NTFS
- I-node

HỆ THỐNG TẬP TIN FAT - 1

- FAT: File Allocation Table
- Các phiên bản của FAT: FAT12, FAT16, FAT32
  - 12,16,32: Số bit dùng để đánh STT các khối

Boot sector	FAT1	FAT2 (backup)	Root directory	Other directories and files					
0000	...			0000	0001	0002	0003	0004	0005
0001	0000			empty	empty	File1	File1	File1	File2
0002	0003								
0003	0004								
0004	FFFF								
0005	0006								
0006	0008								
0007	FFFF								
0008	FFFF								
0009	0000								
	...								
				0006	0007	0008	0009	0010	0011
				File2	File3	File2	empty	empty	empty
				0012	0013	0014	0015	0016	0017
				empty	empty	empty	empty	empty	empty

HỆ THỐNG TẬP TIN FAT - 2

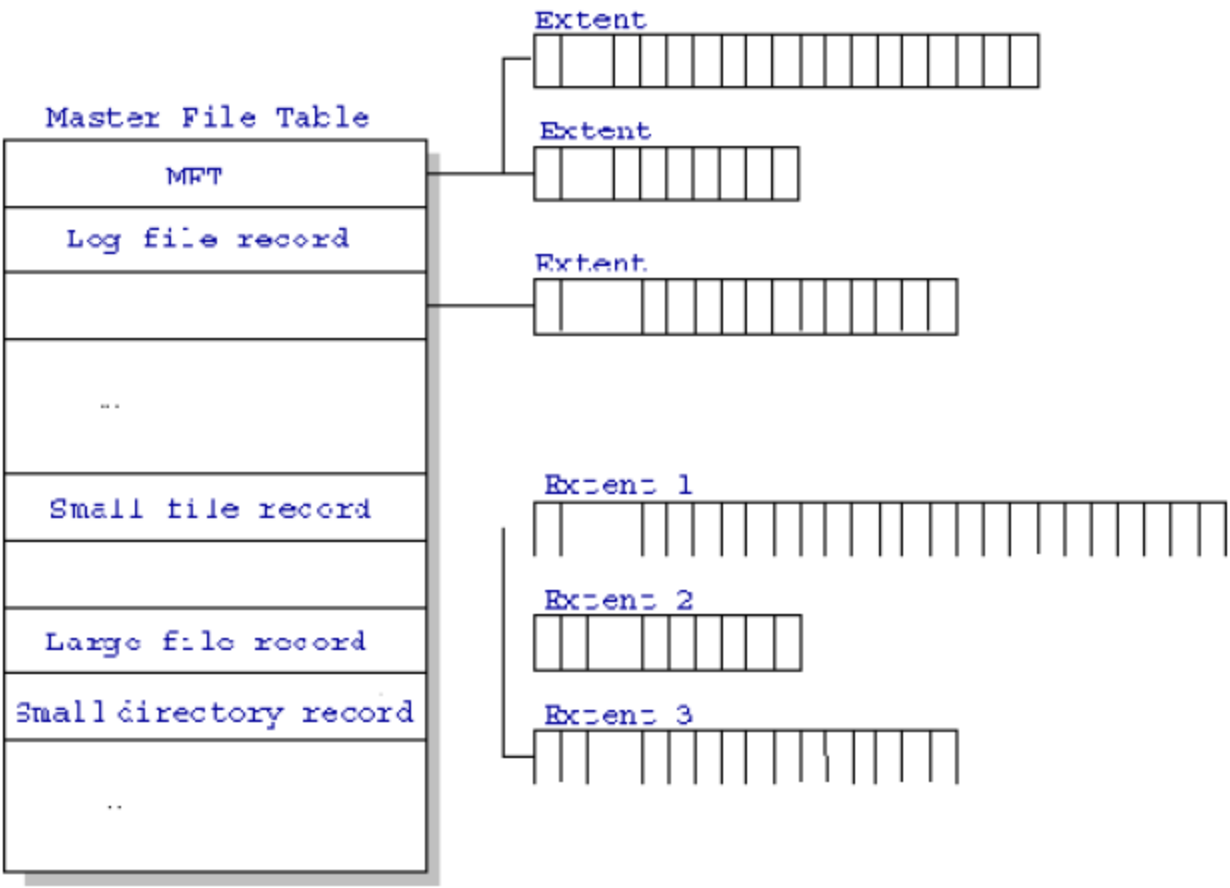
Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

HỆ THỐNG TẬP TIN NTFS - 1

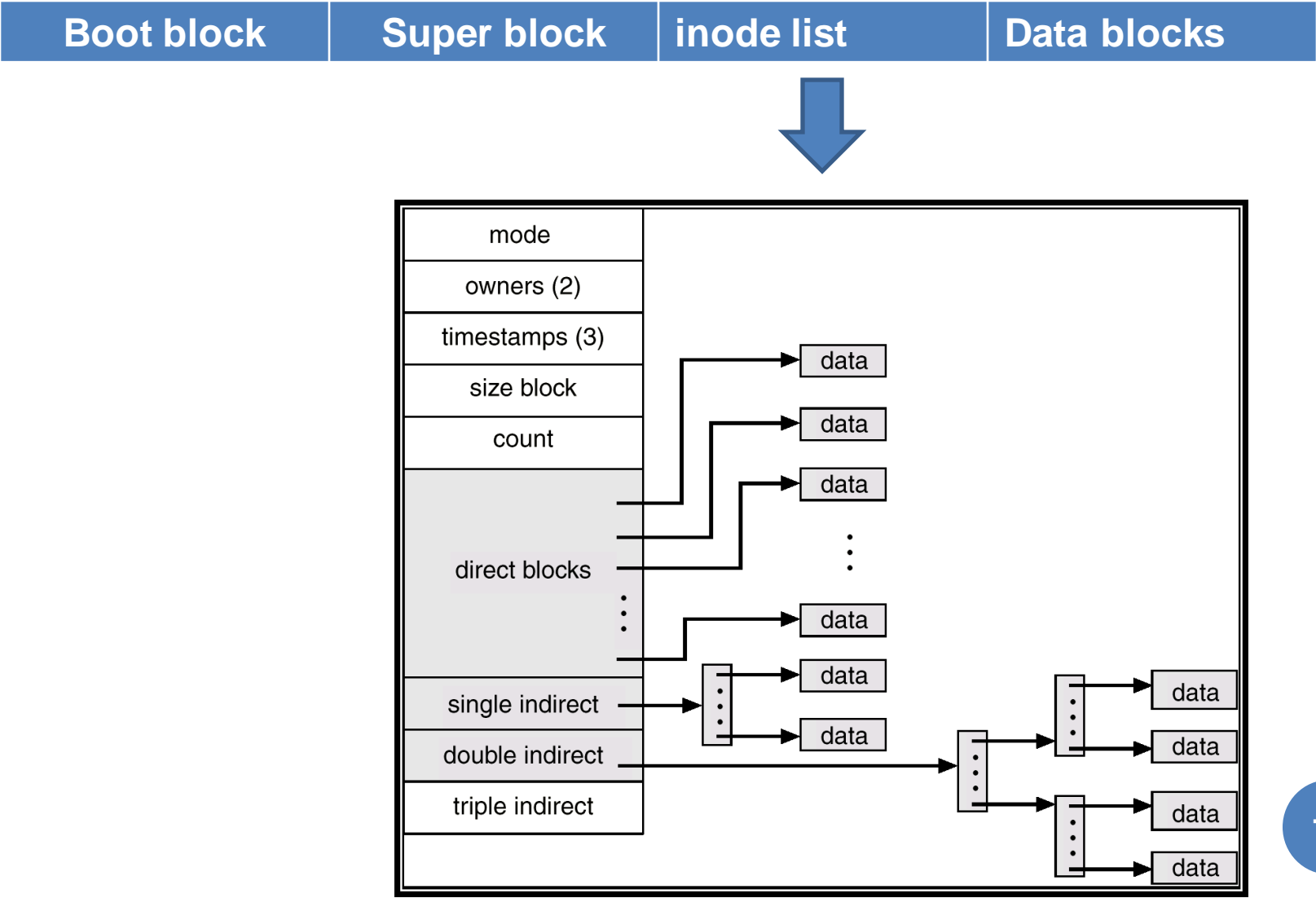
- NTFS: New Technology File System
- Sử dụng MFT (Master File Table).
  - MFT là 1 Metadata file bao gồm 1 danh sách các trường chứa thông tin về mỗi tập tin lưu trữ trên đĩa.
  - Thông tin trong MFT có thể giúp thiết lập các thuộc tính bảo vệ, phục hồi, tìm kiếm, thiết lập quota... cho từng tập tin, thư mục trên đĩa.

Partition boot sector	Master File Table	System files	File area
-----------------------	-------------------	--------------	-----------

HỆ THỐNG TẬP TIN NTFS - 2



HỆ THỐNG TẬP TIN TRÊN UNIX/LINUX: I-NODE - 1



HỆ THỐNG TẬP TIN TRÊN UNIX/LINUX: I-NODE - 2

