



Quản lý hệ thống file

Hệ thống tập tin

- Tập các thuật toán và cấu trúc dữ liệu thích hợp nhằm chuyển các thao tác luận lý trên file thành dữ liệu ghi trên thiết bị lưu trữ.

Mục tiêu của hệ thống file

- Lưu trữ và xử lý dữ liệu
- Đảm bảo đồng nhất dữ liệu, giảm sai.
- Tối ưu hóa hiệu suất hệ thống
- Hỗ trợ nhiều loại thiết bị IO khác nhau
- Cung cấp giao diện người dùng chuẩn
- Hỗ trợ nhiều người dùng

Yêu cầu của người dùng

- Truy xuất file dùng tên gợi nhớ
- Khả năng tạo, xóa, thay đổi file
- Quản lý việc truy xuất đến file hệ thống và file của người dùng khác
- Khả năng tự quản lý file riêng mình
- Sao lưu và phục hồi dữ liệu

Đặt tên file

- Phần tên
- Phần mở rộng

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

Thuộc tính file

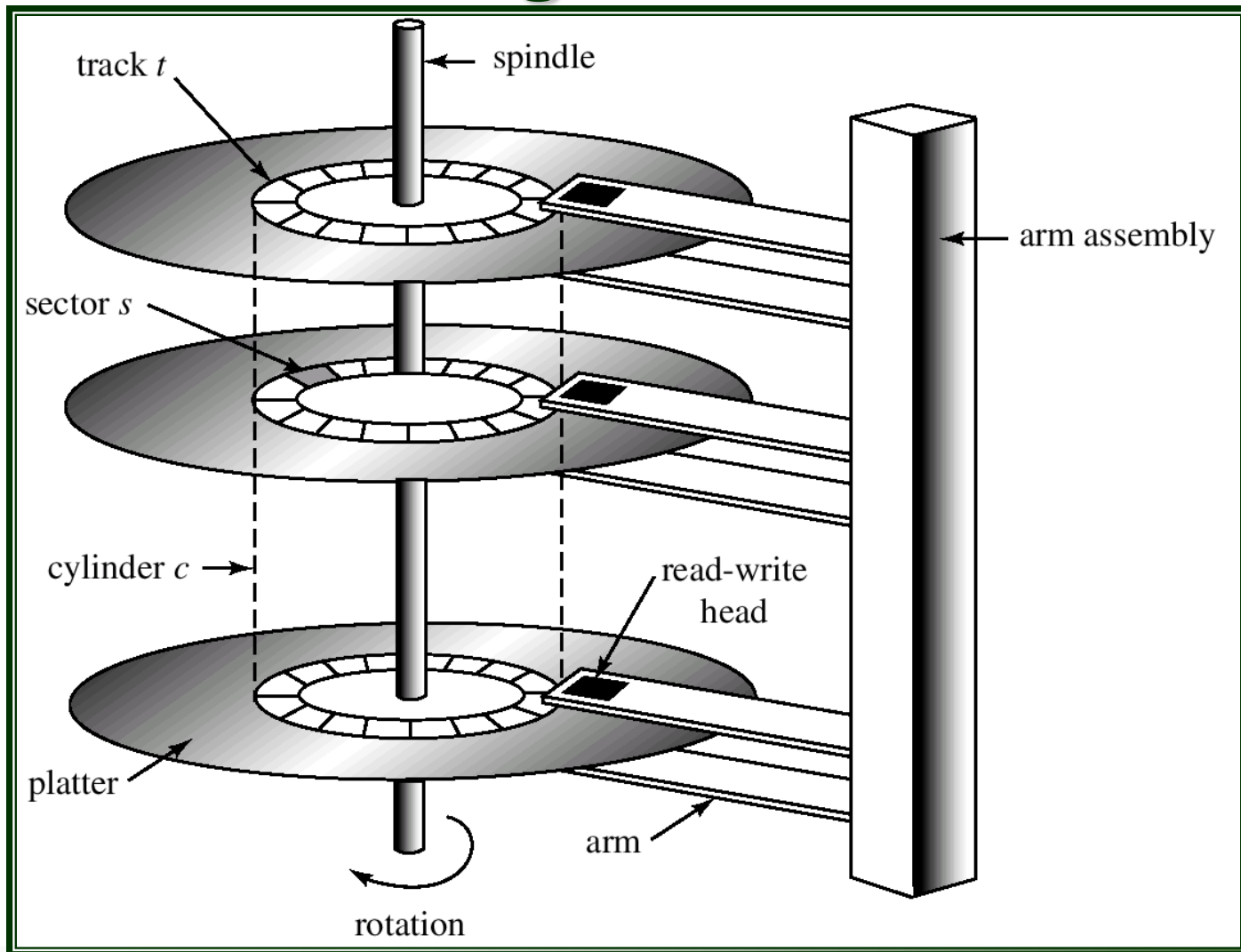
- Read, write, execute, archive, hidden, system...
- Creation, last access, last modification

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

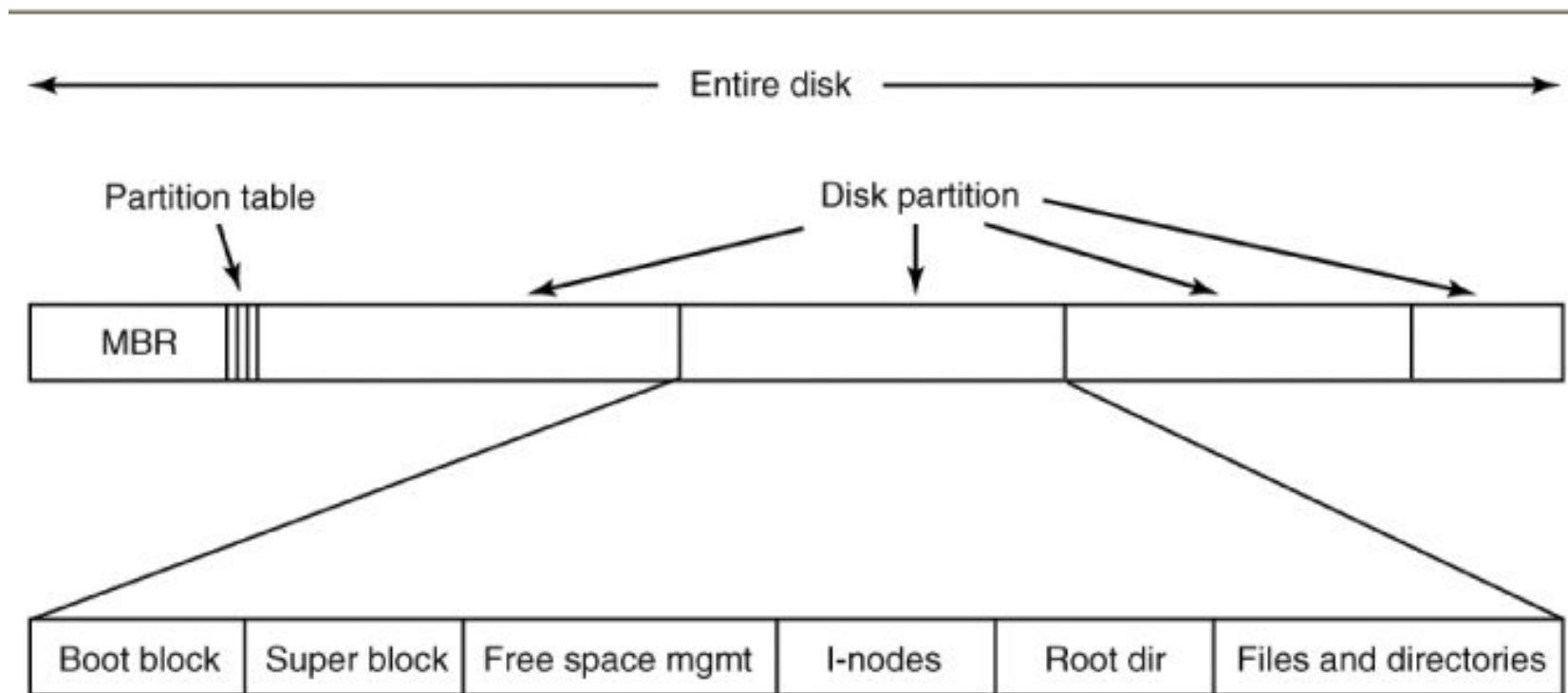
Các thao tác trên file

1. Create
2. Delete
3. Open
4. Close
5. Read
6. Write
7. Append
8. Seek
9. Get attributes
10. Set Attributes
11. Rename

Cấu trúc đĩa cứng



Quản lý phân vùng đĩa cứng

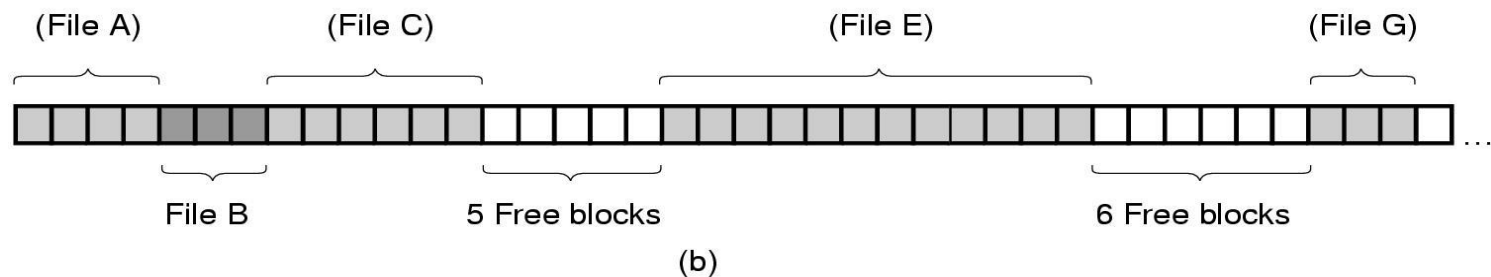
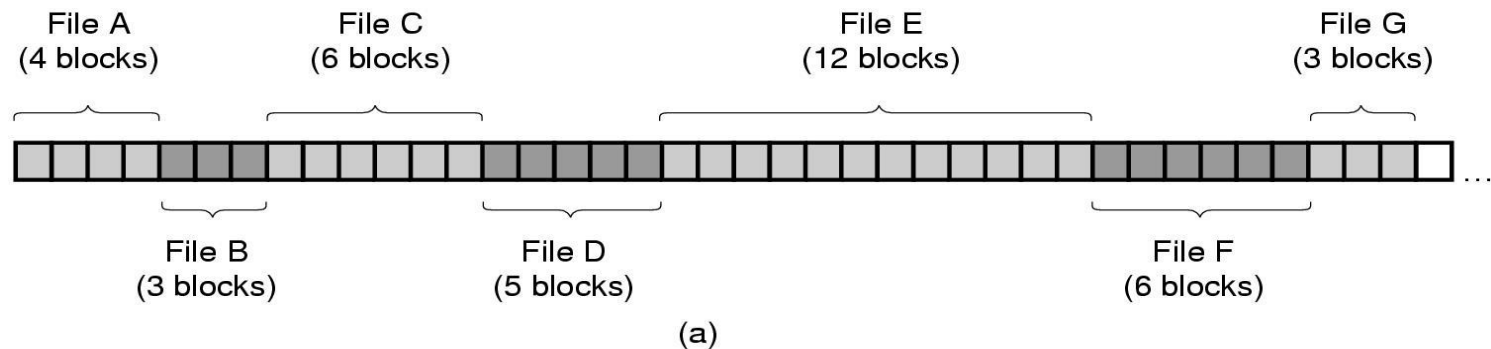


Thực hiện hệ thống file

- Cấp phát liên tục
- Cấp phát theo danh sách liên kết
- Cấp phát theo danh sách liên kết dùng chỉ số index
- i-nodes

Cấp phát liên tục

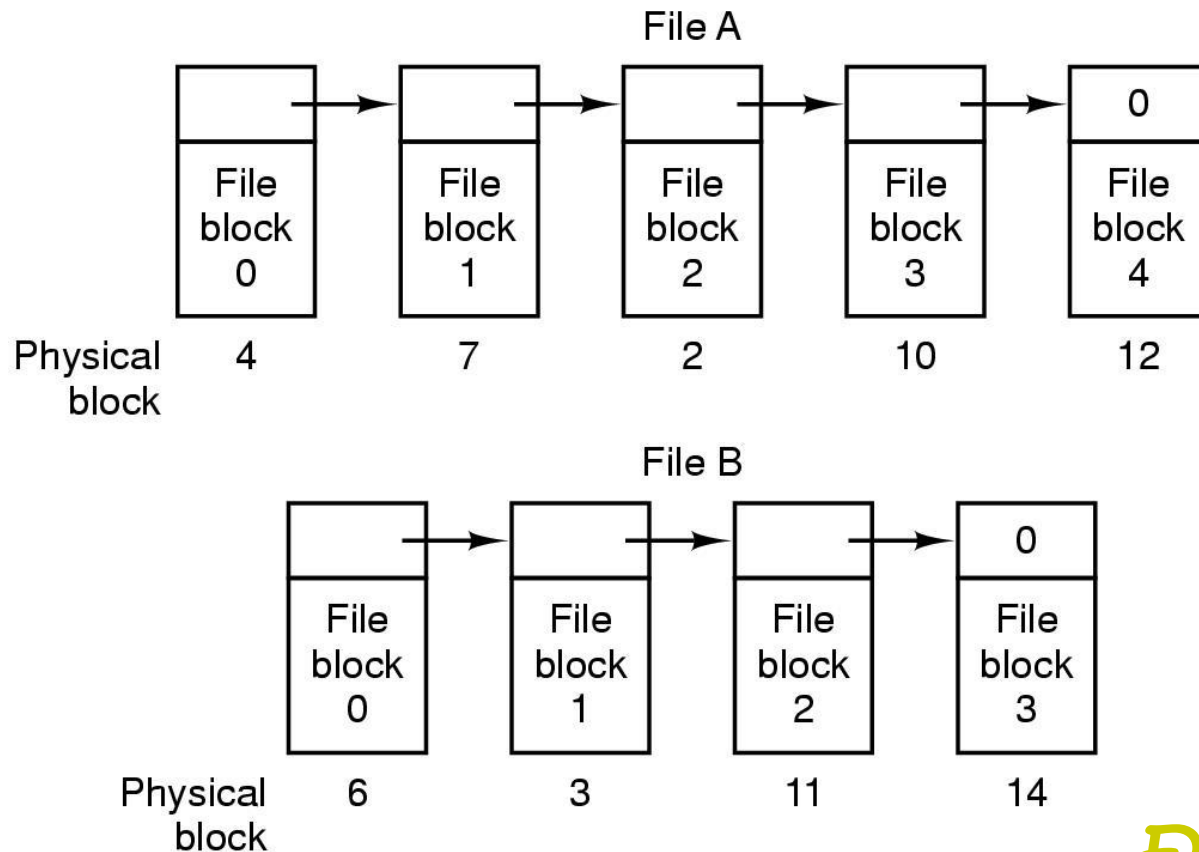
Đặc điểm?



(a) Cấp phát liên tục cho 7 file trên đĩa

(b) Trạng thái mặt đĩa sau khi file D và F bị xóa

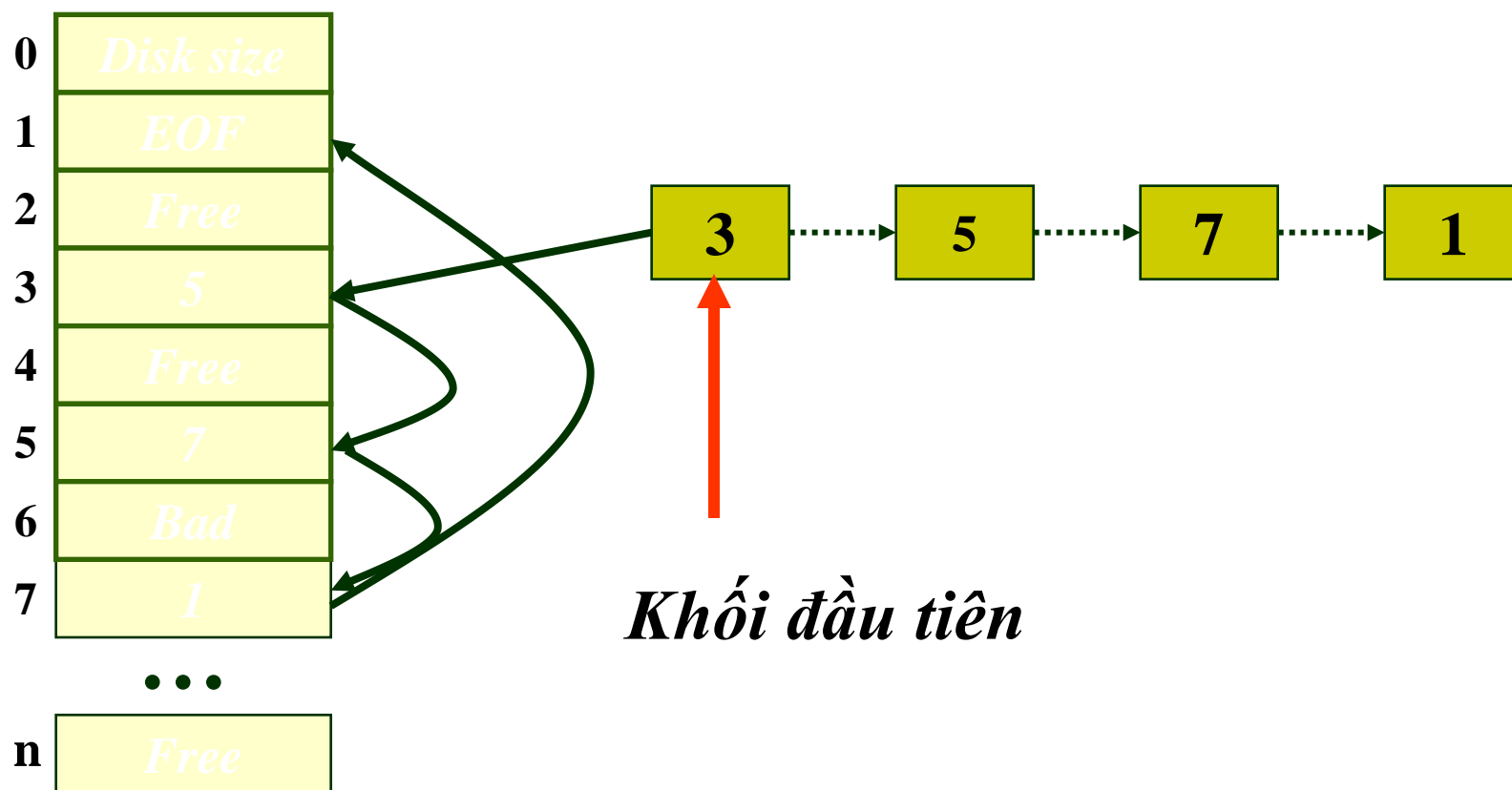
Cấp phát dùng danh sách liên kết



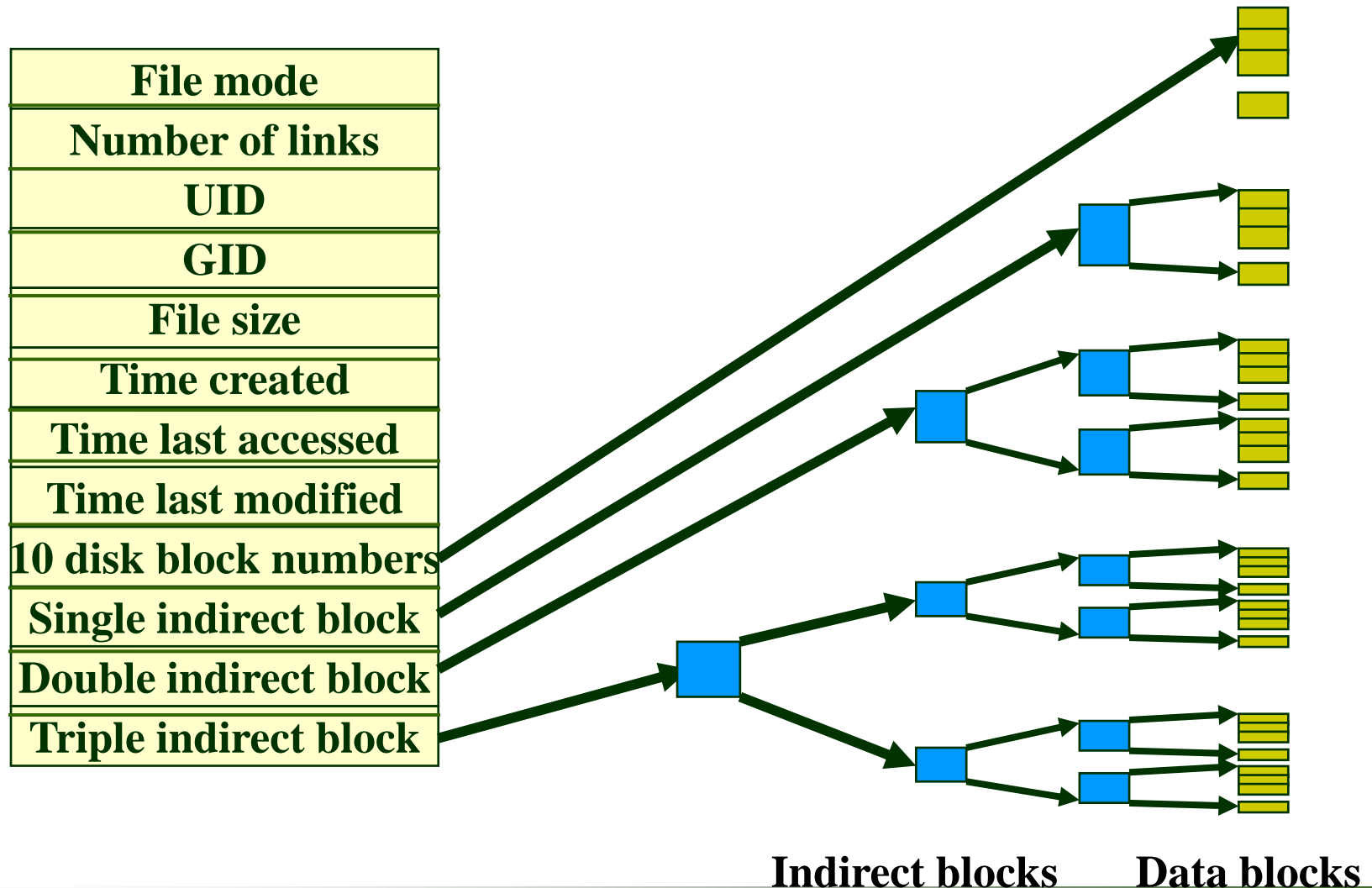
Đặc điểm?

Danh sách liên kết dùng index

FAT (File allocation table)



i-nodes (UNIX)



Cấu trúc thư mục của Windows

8 bytes	3	1	10	2	2	2	4
File name	Ext	A	Reserved	T	D	P	Size

Thuộc tính file

Giờ tạo

Ngày tạo

Khởi đĩa đầu tiên

Cấu trúc thư mục trên Unix

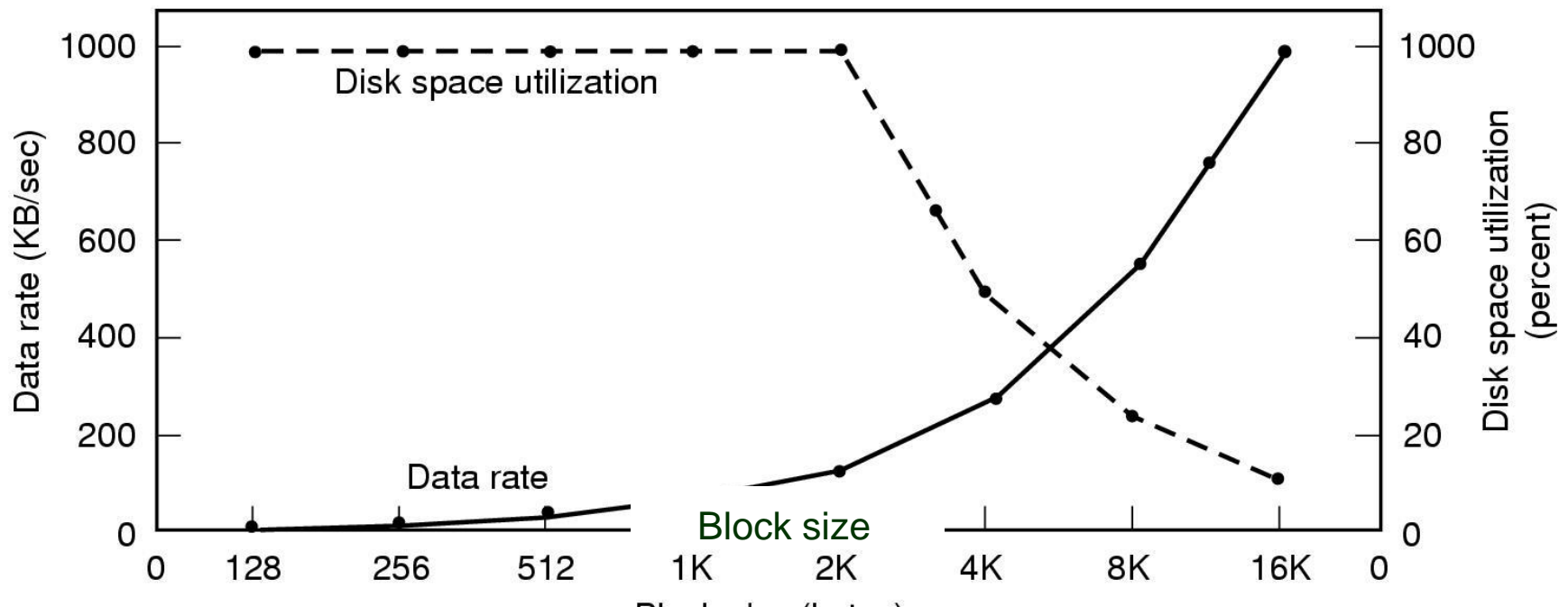
2 bytes

14 bytes

I-node #

File name

Hiệu suất của file system



Quản lý khối trống trên đĩa

- Dùng danh sách liên kết
- Dùng ánh xạ bit (Bit maps)
- Dùng chỉ số index

An abstract graphic design featuring a series of green squares of varying shades (light, medium, and dark) arranged in a stepped, staircase-like pattern on the left side. A solid dark green horizontal bar extends from the right edge of the squares across the top of the image. The word "FAT" is written in white, bold, sans-serif capital letters on the dark green bar.

FAT

FAT

FAT(File Allocation Table) là hệ thống quản lý tập tin đơn giản, được sử dụng chủ yếu trong các hệ điều hành DOS và Windows. Tuy nhiên, FAT(12 và 16) được xem như là các hệ thống quản lý tập tin nhỏ gọn và cơ bản nhất mà bất cứ một hệ điều hành nào cũng phải hỗ trợ.

FAT là tên của một hệ thống tập tin, cũng đồng thời là cơ chế cấp phát khối trên đĩa (block allocation).

Qua quá trình phát triển, đến nay có 3 phiên bản FAT tương thích nhau: là FAT12, FAT16 và FAT32. Chữ số đi sau cho biết số bit trong một entry của bảng FAT.

Tổ chức đĩa theo FAT

Quá trình format sẽ tạo ra trên partition 4 phần riêng biệt như sau:

Boot record: là sector đầu tiên của partition (sector 0 của đĩa mềm), chứa đoạn mã khởi động hệ điều hành và bảng tham số đĩa.

FAT: chứa các entry của bảng FAT, gồm 2 bảng FAT giống nhau, mỗi bảng FAT chiếm một sector hoặc nhiều sector.

Root directory: chứa các entry của thư mục gốc, cũng gồm nhiều sector.

Data area: Vùng chứa dữ liệu, gồm các khối (cluster – gồm 1 hay nhiều sector) để cấp phát cho các tập tin.

Bảng tham số đĩa

Bảng tham số đĩa (BPB_Bios Parameter Block) nằm trong sector đầu tiên của partition (sector 0 của đĩa mềm).

Offset	Kích thước	Ý nghĩa
3	8 byte	Nhận dạng hệ thống
11	1 word	Số byte/sector
13	1 byte	Số sector/cluster
14	1 word	Số sector của boot record
16	1 byte	Số bảng sao của FAT
17	1 word	Số entry trong thư mục gốc
19	1 word	Tổng số sector trên đĩa
21	1 byte	Nhận dạng đĩa
22	1 word	Số sector của mỗi bảng FAT

Cách ghi dữ liệu lên đĩa

Nếu dữ liệu là 1 byte: cách ghi bình thường

Nếu dữ liệu là 1 word: gồm 2 byte, byte thấp ghi trước, byte cao ghi sau

Nếu dữ liệu là 4 byte: chia thành 2 word, word thấp ghi trước, word cao ghi sau.

Ví dụ: Giá trị của trường thứ 2 trong BPB (số byte/sector) là 512 được ghi như sau:

512D = 200H được chia thành 2 byte: 02 và 00

20 là byte cao, 00 là byte thấp

tại offset 11 ghi 00

tại offset 12 ghi 02

Bảng FAT

FAT12: mỗi entry chiếm 12 bit, là một số nguyên không dấu chỉ đến cluster kế tiếp của tập tin.

Bảng FAT bắt đầu ngay sau Boot record.

Để đọc giá trị một entry trong bảng FAT, cần phải đọc 3 byte liên tiếp (2 FAT entries), theo nguyên tắc byte thấp ghi trước byte cao ghi sau.

Ví dụ: 3 byte theo thứ tự trên đĩa là AB CD EF được chuyển thành 2 FAT entries như sau:

EFC DAB -> DAB EFC (thấp ghi trước, cao ghi sau)

ngược lại, nếu muốn ghi 2 entries có số hiệu 123 456 thì ghi các byte theo thứ tự: 23 61 45

FAT16: mỗi entry chiếm 16 bit (1 word), thao tác đọc đơn giản hơn

Các giá trị đặc biệt của FAT entry

000 :	cluster trống
001:	không dùng
002 đến FEF:	chỉ số cluster đã dùng
FF7:	cluster hỏng (bad block)
FF8 đến FFF:	cluster cuối cùng của tập tin

Chú ý:

- Hai entries đầu tiên của bảng FAT (tương ứng 3 byte đầu tiên của sector đầu tiên trên bảng FAT) không dùng để định vị cluster.
- Vị trí dữ liệu bắt đầu ngay sau sector cuối cùng của bảng Root directory.

Root directory

Chứa các thông tin cần thiết để quản lý các tập tin và thư mục con trong thư mục gốc của đĩa. Root directory nằm ngay sau bảng FAT, được tổ chức thành các entry 32 byte có nội dung như sau:

Offset	Kích thước	Ý nghĩa
0	8 byte	tên tập tin (ASCII)
8	3 byte	tên tập tin (mở rộng)
11	1 byte	thuộc tính của tập tin
12	10 byte	không dùng
22	1 word	thời gian tạo file
24	1 word	ngày tạo file
26	1 word	cluster bắt đầu
28	4 byte	Kích thước file

Tên tập tin

Tên tập tin tuân theo quy tắc đặt tên của DOS. Byte đầu tiên của tên có ý nghĩa đặc biệt như sau:

00: entry chưa sử dụng, dùng để kết thúc thao tác liệt kê

E5: entry đã bị xóa. Khi một tập tin bị xóa, byte đầu tiên trong trường tên file được đổi thành E5, các thông tin khác vẫn giữ nguyên.

2E: (mã ASCII của dấu .) nếu chỉ có 1 byte 2E, entry được đánh dấu là 1 thư mục con. Nếu có 2 byte liên tiếp là 2E, đây là entry của thư mục cấp trên của thư mục hiện hành

Thuộc tính của tập tin

Bit	Hex	Ý nghĩa
00000001	1	Read only
00000010	2	Hidden
00000100	4	System
00001000	8	Volume label
00010000	10	Subdirectory
00100000	20	Archive
01000000	40	Reserved
10000000	80	Reserved

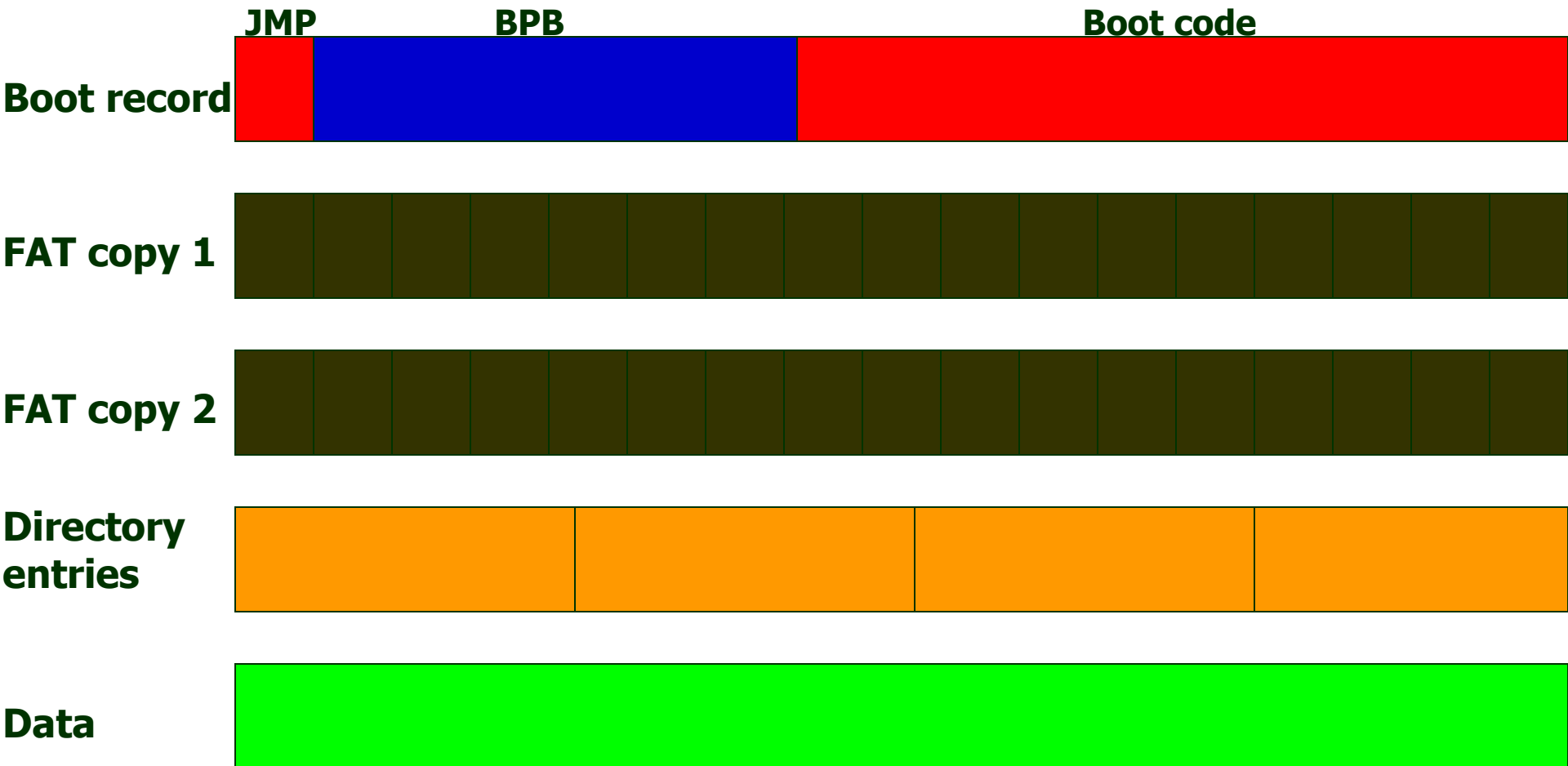
Lưu trữ thời gian

Ngày và giờ được lưu trữ trong directory entry dưới dạng một số nguyên không dấu dài 2 byte, được xác định như sau:

$$\text{Time} = \text{Hour} * 2048 + \text{minute} * 32 + \text{second} / 2$$

$$\text{Date} = (\text{year} - 1980) * 512 + \text{month} * 32 + \text{day}$$

Cấu trúc đĩa mềm theo FAT



Thao tác trực tiếp trên đĩa

Có nhiều cách để đọc và ghi trực tiếp các sector trên đĩa:

- Dùng ngắt 13h của BIOS
- Dùng ngắt 25h và 26h của DOS
- Dùng hàm `absread()` và `abswrite()` của C, thực chất là gọi ngắt 25h và 26h của DOS thông qua thư viện C.

Cách dùng thư viện C:

```
#include<dos.h>
```

```
#include<process.h>
```

```
absread(a,b,c,d);
```

a: ổ đĩa cần truy xuất (0:A, 1:B, 2:C, ...)

b: số sector cần đọc/ghi

c: offset của khối cần đọc/ghi

d: mảng chứa dữ liệu đọc/ghi