

Chương 2: Hệ Thống Files

Hệ điều hành

ThS. Đinh Xuân Trường
truongdx@ptit.edu.vn



Posts and Telecommunications
Institute of Technology
Faculty of Information Technology 1



CNTT1
Học viện Công nghệ Bưu chính Viễn thông

August 15, 2022

- Cấp phát không gian cho file

 - Cấp phát các khối liên tiếp

 - Cấp phát sử dụng danh sách kết nối

 - Cấp phát sử dụng danh sách kết nối trên bảng chỉ số

 - Cấp phát sử dụng khối chỉ số

- Quản lý không gian trống trên đĩa

 - Bảng bit

 - Danh sách kết nối

 - Danh sách vùng trống

- Độ tin cậy của hệ thống file

 - Phát hiện và loại trừ các khối hỏng

 - Sao dự phòng

 - Kiểm tra tính toàn vẹn của hệ thống file

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ▶ ↺ 🔍 ↻

Nhiệm vụ quan trọng của HDH trong việc quản lý hệ thống file:

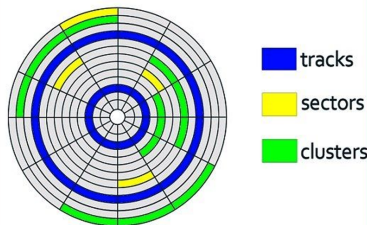
- ▶ Cấp phát không gian trên đĩa và các thiết bị nhớ ngoài để lưu trữ file và thư mục
- ▶ Ghi lại vị trí các khối nhớ đã cấp phát để có thể tiến hành truy cập về sau.

Sector hay Cung là đơn vị nhỏ nhất do chương trình điều khiển đĩa cho phép đọc hoặc ghi (khối vật lý)

1 sector = 512byte

Cluster hay đơn vị cấp phát gồm một số khối vật lý, là đơn vị thông tin nhỏ nhất mà hệ điều hành cấp phát cho file (khối logic) 2KB-32KB

Hard disk drive structure



- ▶ Mỗi file gồm 1 tập các khối, HĐH chịu trách nhiệm cấp phát các khối cho file:
 - Không gian trên đĩa phải được cấp phát cho file
 - Theo dõi không gian trống sẵn sàng cho việc cấp phát
 - Thực hiện đảm bảo tối ưu không gian và tốc độ truy cập trên đĩa
- ▶ Một số vấn đề đặt ra:
 - Không gian tối đa cấp phát cho file 1 lần là bao nhiêu?
 - Không gian cấp phát cho file (partion), kích thước phần như thế nào?

Mục đích cấp phát không gian cho file

- ▶ Tăng hiệu năng truy cập tuần
- ▶ Dễ dàng truy cập ngẫu nhiên đến
- ▶ Dễ dàng quản lý file

Các phương pháp cấp phát không gian cho file

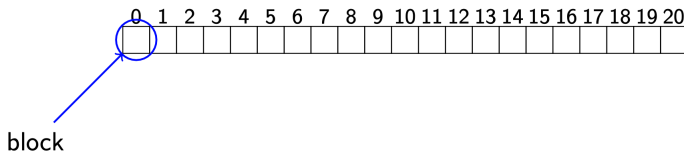
- ▶ Cấp phát khối liên tiếp
- ▶ Sử dụng danh sách kết nối
- ▶ Sử dụng danh sách kết nối trên bảng chỉ số
- ▶ Sử dụng khối chỉ số

Cấp phát không gian cho file

Cấp phát các khối liên tiếp

- ▶ **Nguyên tắc:** file được cấp phát các khối liên tiếp trên đĩa
- ▶ HĐH chọn 1 vùng trống có đủ số lượng khối cho file
- ▶ Bảng cấp phát xác định vị trí file gồm 1 khoản mục cho 1 file, khối bắt đầu, và số khối (độ dài) của file
- ▶ HĐH cấp phát trước và biết kích thước file khi tạo file

File	Pos	Size
file-1	15	4
file-2	4	5
file-3	11	3
Directory		

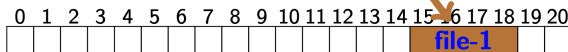


Cấp phát không gian cho file (cont.)

Cấp phát các khối liên tiếp

- ▶ **Nguyên tắc:** file được cấp phát các khối liên tiếp trên đĩa
- ▶ HĐH chọn 1 vùng trống có đủ số lượng khối cho file
- ▶ Bảng cấp phát xác định vị trí file gồm 1 khoản mục cho 1 file, khối bắt đầu, và số khối (độ dài) của file
- ▶ HĐH cấp phát trước và biết kích thước file khi tạo file

File	Pos	Size
file-1	15	4
file-2	4	5
file-3	11	3
Directory		

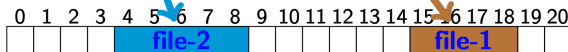


Cấp phát không gian cho file (cont.)

Cấp phát các khối liên tiếp

- ▶ **Nguyên tắc:** file được cấp phát các khối liên tiếp trên đĩa
- ▶ HĐH chọn 1 vùng trống có đủ số lượng khối cho file
- ▶ Bảng cấp phát xác định vị trí file gồm 1 khoản mục cho 1 file, khối bắt đầu, và số khối (độ dài) của file
- ▶ HĐH cấp phát trước và biết kích thước file khi tạo file

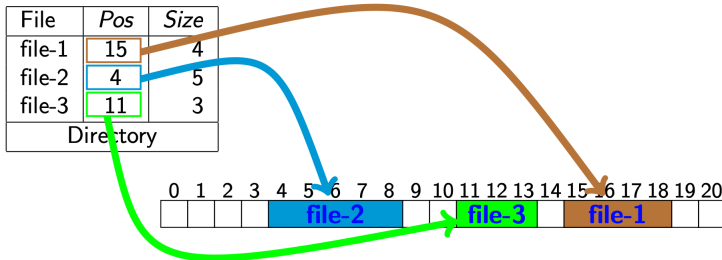
File	Pos	Size
file-1	15	4
file-2	4	5
file-3	11	3
Directory		



Cấp phát không gian cho file (cont.)

Cấp phát các khối liên tiếp

- ▶ **Nguyên tắc:** file được cấp phát các khối liên tiếp trên đĩa
- ▶ HĐH chọn 1 vùng trống có đủ số lượng khối cho file
- ▶ Bảng cấp phát xác định vị trí file gồm 1 khoản mục cho 1 file, khối bắt đầu, và số khối (độ dài) của file
- ▶ HĐH cấp phát trước và biết kích thước file khi tạo file



Cấp phát không gian cho file (cont.)

Cấp phát các khối liên tiếp

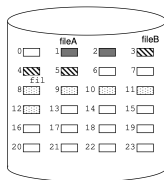
Cấp phát các khối liên tiếp:

Ưu điểm:

- + Cho phép truy cập trực tiếp và tuần tự
- + Đơn giản, tốc độ cao do tiết kiệm thời gian di chuyển đầu từ khi đọc các khối

Nhược điểm:

- Phải biết trước kích thước file khi tạo
- Khó tìm ra khoảng không gian trống đủ lớn trên đĩa để cấp phát cho file
- HĐH cần kiểm tra vùng trống khi cấp phát
- Gây phân mảnh ngoài



Tên file	Thư mục	
	Bắt đầu	Độ dài
fileA	1	2
fileB	3	3
fileB	8	5

* *Phân mảnh ngoài: hiện tượng vùng trống còn lại trên đĩa có kích thước quá nhỏ do vậy không thể cấp phát cho file có kích thước lớn hơn.*

Cấp phát không gian cho file

Cấp phát sử dụng danh sách kết nối

- ▶ **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- ▶ File được cấp thêm khối mới, khối đó thêm vào cuối danh
- ▶ Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo

File	Pos	End
abc	12	3
def	5	11
Directory		

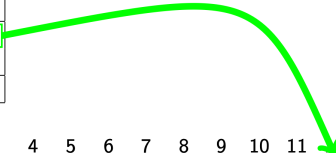
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	3	-1	0	6	8	14	9	11	7	-1	10	0	15	2	0	0

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối

- ▶ **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- ▶ File được cấp thêm khối mới, khối đó thêm vào cuối danh
- ▶ Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo

File	Pos	End
abc	12	3
def	5	11
Directory		



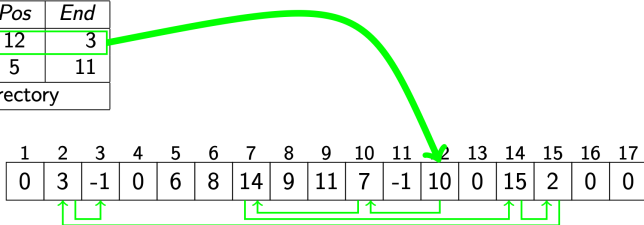
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	3	-1	0	6	8	14	9	11	7	-1	10	0	15	2	0	0

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối

- ▶ **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- ▶ File được cấp thêm khối mới, khối đó thêm vào cuối danh
- ▶ Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo

File	Pos	End
abc	12	3
def	5	11
Directory		



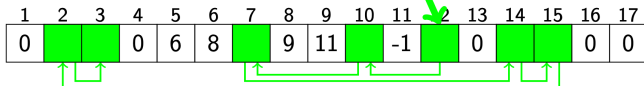
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	3	-1	0	6	8	14	9	11	7	-1	10	0	15	2	0	0

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối

- **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- File được cấp thêm khối mới, khối đó thêm vào cuối danh
- Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo

File	Pos	End
abc	12	3
def	5	11
Directory		

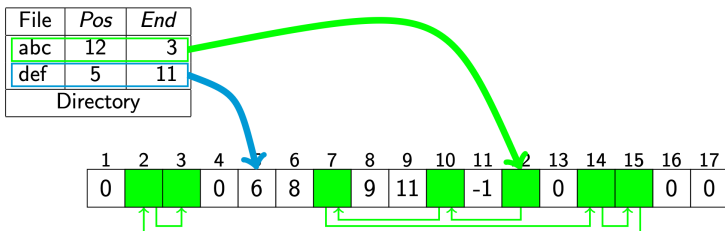


File abc gồm 7 khối: 12, 10, 7, 14, 15, 2, 3

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối

- **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- File được cấp thêm khối mới, khối đó thêm vào cuối danh
- Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo

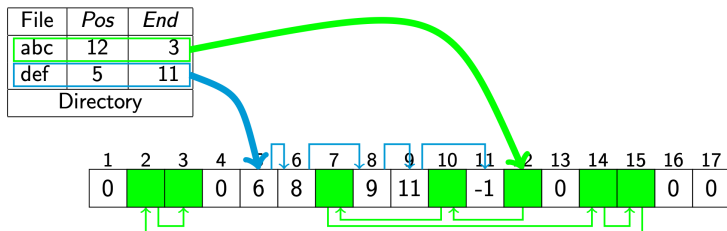


File abc gồm 7 khối: 12, 10, 7, 14, 15, 2, 3

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối

- **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- File được cấp thêm khối mới, khối đó thêm vào cuối danh
- Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định khối tiếp theo

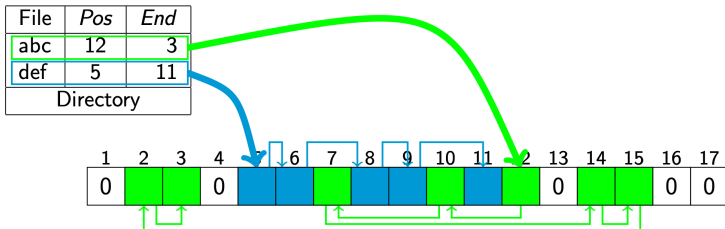


File abc gồm 7 khối: 12, 10, 7, 14, 15, 2, 3

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối

- **Nguyên tắc:** File được phân phối các khối nhớ kết nối với nhau thành danh sách kết nối, đầu khối chứa con trỏ tới khối tiếp theo
- File được cấp thêm khối mới, khối đó thêm vào cuối danh
- Bảng cấp phát chứa con trỏ tới khối đầu tiên của file, HĐH đọc lần lượt từng khối và sử dụng con trỏ để xác định 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

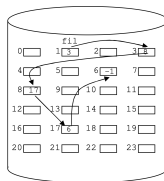
Cấp phát các khối liên tiếp:

Ưu điểm:

- + Không bị phân mảnh do các khối không nhất thiết phải nằm cạnh nhau
- + Không yêu cầu biết trước kích thước file
- + Dễ tìm vị trí cho file, khoản mục đơn giản

Nhược điểm:

- Không hỗ trợ truy cập trực tiếp
- Để đọc một khối ta phải đọc từ khối đầu tiên cho tới khối cần đọc
- Khi con trỏ thay đổi giá trị đổi, xác định khối nào thuộc file nào sẽ không chính xác
- Giảm độ tin cậy, tính toàn vẹn Hệ thống file
- Tốc độ truy cập không cao



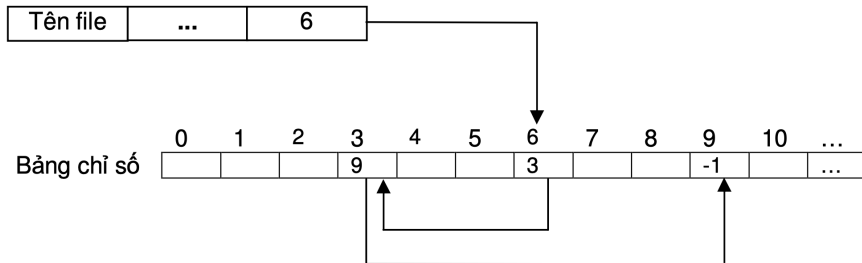
Thư mục	
Tên file	Bắt đầu
fileA	1

Cấp phát không gian cho file

Cấp phát sử dụng danh sách kết nối trên bảng chỉ số

- ▶ Bảng chỉ số: mỗi ô của bảng ứng với 1 khối của đĩa
- ▶ Con trỏ tới khối tiếp của file được chứa trong ô tương ứng của bảng
- ▶ Mỗi đĩa logic có 1 bảng chỉ số được lưu ở vị trí xác định
- ▶ Kích thước mỗi ô trên bảng phụ thuộc vào số lượng khối trên đĩa

thư mục



Cấp phát không gian cho file (cont.)

Cấp phát sử dụng danh sách kết nối trên bảng chỉ số

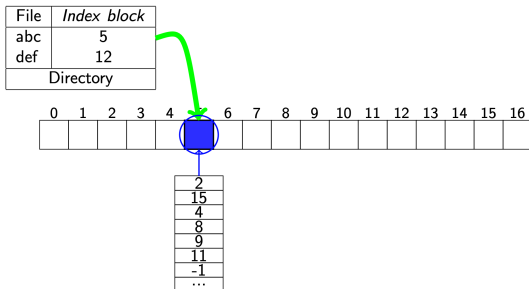


- ▶ Cho phép truy cập file trực tiếp: đi theo chuỗi con trỏ chứa trong bảng chỉ mục
- ▶ Bảng FAT (File Allocation Table) được lưu ở đầu mỗi đĩa logic sau sector khởi động
- ▶ FAT12, FAT16, FAT32: Mỗi ô của bảng có kích thước 12, 16, 32 bit cho phép quản lý tối đa 2^{12} , 2^{16} , 2^{32}

Cấp phát không gian cho file

Cấp phát sử dụng khối chỉ số

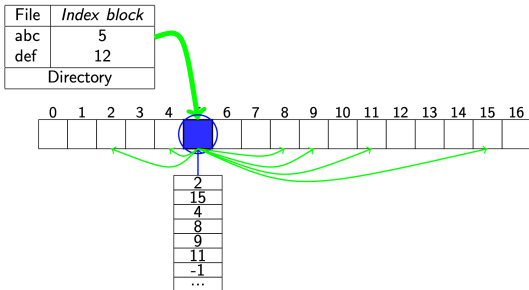
- ▶ **Nguyên tắc:** Mỗi file có một khối chỉ mục chính (index block) chứa danh sách các khối dữ liệu của file
- ▶ Tất cả con trỏ tới các khối thuộc về 1 file được tập trung 1 chỗ, trong một khối gọi là khối chỉ mục (l-node)
- ▶ Mảng chứa thuộc tính của file và vị trí các khối của file trên đĩa
- ▶ Ô thứ i của mảng chứa con trỏ tới khối thứ i của file



Cấp phát không gian cho file (cont.)

Cấp phát sử dụng khối chỉ số

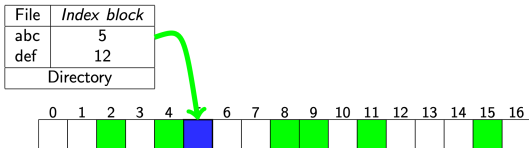
- ▶ **Nguyên tắc:** Mỗi file có một khối chỉ mục chính (index block) chứa danh sách các khối dữ liệu của file
- ▶ Tất cả con trỏ tới các khối thuộc về 1 file được tập trung 1 chỗ, trong một khối gọi là khối chỉ mục (l-node)
- ▶ Mảng chứa thuộc tính của file và vị trí các khối của file trên đĩa
- ▶ Ô thứ i của mảng chứa con trỏ tới khối thứ i của file



Cấp phát không gian cho file (cont.)

Cấp phát sử dụng khối chỉ số

- ▶ **Nguyên tắc:** Mỗi file có một khối chỉ mục chính (index block) chứa danh sách các khối dữ liệu của file
- ▶ Tất cả con trỏ tới các khối thuộc về 1 file được tập trung 1 chỗ, trong một khối gọi là khối chỉ mục (l-node)
- ▶ Mảng chứa thuộc tính của file và vị trí các khối của file trên đĩa
- ▶ Ô thứ i của mảng chứa con trỏ tới khối thứ i của file

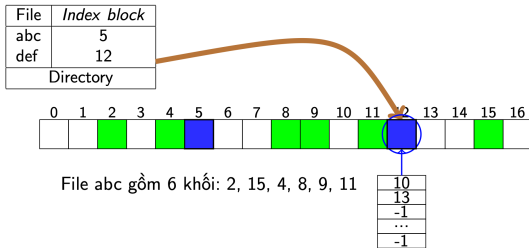


File abc gồm 6 khối: 2, 15, 4, 8, 9, 11

Cấp phát không gian cho file (cont.)

Cấp phát sử dụng khối chỉ số

- ▶ **Nguyên tắc:** Mỗi file có một khối chỉ mục chính (index block) chứa danh sách các khối dữ liệu của file
- ▶ Tất cả con trỏ tới các khối thuộc về 1 file được tập trung 1 chỗ, trong một khối gọi là khối chỉ mục (l-node)
- ▶ Mảng chứa thuộc tính của file và vị trí các khối của file trên đĩa
- ▶ Ô thứ i của mảng chứa con trỏ tới khối thứ i của file

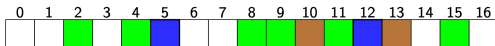


Cấp phát không gian cho file (cont.)

Cấp phát sử dụng khối chỉ số

- ▶ **Nguyên tắc:** Mỗi file có một khối chỉ mục chính (index block) chứa danh sách các khối dữ liệu của file
- ▶ Tất cả con trỏ tới các khối thuộc về 1 file được tập trung 1 chỗ, trong một khối gọi là khối chỉ mục (l-node)
- ▶ Mảng chứa thuộc tính của file và vị trí các khối của file trên đĩa
- ▶ Ô thứ i của mảng chứa con trỏ tới khối thứ i của file

File	Index block
abc	5
def	12
Directory	



File abc gồm 6 khối: 2, 15, 4, 8, 9, 11

File def gồm 2 khối: 10, 13

Cấp phát không gian cho file (cont.)

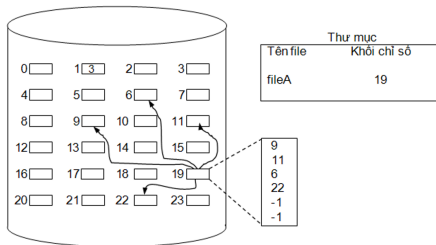
Cấp phát sử dụng khối chỉ số

► Chọn kích thước l-node:

- Nhỏ: tiết kiệm không gian nhưng không đủ con trỏ tới các khối nếu file lớn
- Lớn: với file nhỏ chỉ chiếm 1 vài ô thì lãng phí

► Giải pháp:

- Thay đổi kích thước i-node = sử dụng danh sách kết nối
- Sử dụng l-node có cấu trúc nhiều mức



Cấp phát không gian cho file (cont.)

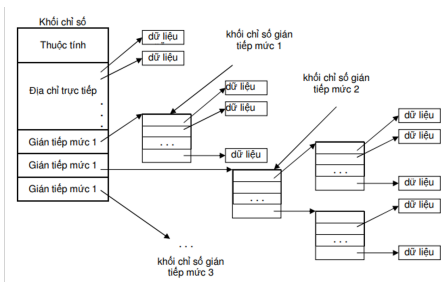
Cấp phát sử dụng khối chỉ số

► Ưu điểm:

- Cho phép truy cập trực tiếp
- Các khối thuộc 1 file không cần nằm liên tiếp nhau

► Nhược điểm:

- Tốc độ truy cập file chậm



- ▶ Không gian trên đĩa được cấp phát theo từng khối không phân chia. Để quản lý hiệu quả không gian trên đĩa, ngoài phương pháp cấp phát khối nhớ cho file, cần chú ý:
 - Lựa chọn kích thước cho khối
 - Quản lý khối trống
- ▶ Kích thước khối:
 - Khối là đơn vị cấp phát nhỏ nhất của hệ điều hành, mỗi file bao gồm một số nguyên các khối nhớ.

- ▶ Kích thước khối lớn:
 - Giảm kích thước bảng chỉ mục, tăng tốc độ đọc file;
 - Bị phân mảnh trong
- ▶ Kích thước khối nhỏ:
 - Mỗi file chiếm nhiều khối nhỏ, nằm rải rác trên đĩa
 - Thời gian đọc file lâu
- ▶ Chọn kích thước khối tùy thuộc:
 - Kích thước đĩa: đĩa lớn, chọn kích thước khối lớn \Rightarrow thời gian truy cập nhanh, đơn giản hóa việc quản lý
 - Kích thước file: hệ thống sử dụng nhiều file lớn, kích thước tăng và ngược
- ▶ Kích thước khối thường là lũy thừa 2 của sector và nằm trong khoảng từ 512B tới 32 KB

- ▶ Để có thể cấp phát khối nhớ trên đĩa cho file, HDH cần biết khối nào hiện đang trống. Các khối trống bao gồm những khối chưa được cấp phát lần nào hoặc cấp phát rồi nhưng đã được giải phóng.
- ▶ Các phương pháp quản lý không gian trống:
 - Bảng bit
 - Danh sách kết nối
 - Danh sách vùng trống

- ▶ Vector bit là mảng 1 chiều
- ▶ Mỗi ô có kích thước 1 bit tương ứng với một khối trên đĩa
- ▶ Khối được cấp phát có bit tương ứng là 0, khối trống: 1 hoặc ngược lại
- ▶ Dễ tìm 1 hoặc nhóm các khối trống liên tiếp
- ▶ Với đĩa có kích thước lớn, đọc toàn bộ vector bit vào MEM có thể đòi hỏi khá nhiều không gian nhớ

Ví dụ: Trong trường hợp sau, các khối 0, 1, 8, 9, 10 đã được cấp phát, các khối còn lại chưa bị sử dụng:

0011111100011111...

Quản lý không gian trống trên đĩa

Danh sách kết nối



- ▶ Các khối trống được liên kết với nhau thành danh sách
- ▶ Mỗi khối trống chứa con trỏ chỉ tới khối trống tiếp theo
- ▶ Địa chỉ khối trống đầu tiên được lưu ở vị trí đặc biệt trên đĩa và được HDH giữ trong MEM khi cần làm việc với các file
- ▶ Đòi hỏi truy cập lần lượt khi cần duyệt danh sách này
- ▶ HDH có thể cấp phát ngay các khối ở đầu danh sách

- ▶ Các khối nằm liên nhau thường được cấp phát và giải phóng đồng thời
- ▶ Lưu vị trí khối trống đầu tiên của vùng các khối trống liên tiếp và số lượng các khối trống nằm liên sau đó
- ▶ Thông tin trên được lưu vào danh sách riêng

- ▶ File là nơi lưu trữ thông tin rất quan trọng đối với người dùng, thậm chí quan trọng với chính bản thân hệ thống tính toán.
- ▶ Hệ thống file phải có tính bền vững cao, phải đảm bảo lưu trữ thông tin sao cho thông tin được toàn vẹn và không bị mất mát.
- ▶ Thông tin của hệ thống file lưu trên đĩa và các thiết bị nhớ ngoài khác có thể bị sai lệch, mất mát do:
 - Lỗi phần cứng/phần mềm
 - Lỗi sự cố kỹ thuật
 - Mặc dù không thể ngăn chặn các sự cố như vậy => cũng cần được xây dựng sao cho có khả năng phát hiện và khắc phục hậu quả trong khả năng cao nhất

- ▶ Trong quá trình sản xuất cũng như sử dụng, đĩa có thể chứa các khối hỏng, không thể dùng cho đọc/ghi thông tin.
 - Đĩa mềm, các khối hỏng thường xuất hiện trong quá trình sử dụng
 - Đĩa cứng: thường có các khối hỏng ngay trong quá trình sản xuất. Những khối hỏng này được nhà sản xuất phát hiện trong quá trình kiểm tra đĩa và được lưu trong danh sách khối hỏng đi cùng với đĩa

► Phát hiện và loại trừ khối hỏng

- Phương pháp 1:

- Một sector trên đĩa được dành riêng chứa danh sách các khối hỏng
- Một số khối không hỏng được dành riêng để dự trữ
- Các khối hỏng thay thế bởi các khối dự trữ bằng thay thế địa chỉ
- Truy cập tới khối hỏng thành truy cập tới khối dự trữ

- Phương pháp 2:

- Tập trung tất cả các khối hỏng thành 1 file
- Đã cấp phát và không được sử dụng nữa

- ▶ Tạo ra một bản sao của đĩa trên một vật mang khác
- ▶ Sao lưu toàn bộ (full backup):
 - Ghi toàn bộ thông tin trên đĩa ra vật mang tin khác
 - Chắc chắn nhưng tốn nhiều thời gian
- ▶ Sao lưu tăng dần (incremental backup):
 - Được sử dụng sau khi đã tiến hành full backup ít nhất 1 lần
 - Chỉ ghi lại các file đã bị thay đổi sau lần sao lưu cuối cùng
 - Hệ thống lưu trữ thông tin về các lần lưu trữ file
 - DOS: file thay đổi, archive bit = 1
- ▶ Kết hợp:
 - Full backup: hàng tuần/ tháng
 - Incremental backup: hàng ngày

- ▶ Hệ thống file chứa nhiều CTDL có mối liên kết:
 - Thông tin về liên kết bị hư hại thì tính toàn vẹn của hệ thống bị phá vỡ
 - Liên kết đứt đoạn, không biết lưu ở khối nào tiếp theo
- ▶ Các khối không có mặt trong danh sách các khối trống, đồng thời cũng không có mặt trong một file nào
- ▶ Một khối có thể vừa thuộc về một file nào đó vừa có mặt trong danh sách khối trống
- ▶ HDH có các chương trình kiểm tra tính toàn vẹn của hệ thống file, được chạy khi hệ thống khởi động, đặc biệt là sau sự cố

Ví dụ: chương trình SCANDISK củ WIN hoặc DOS

- ▶ Mặc dù có thể kiểm tra tính toàn vẹn của các liên kết trong hệ thống file. Tuy nhiên, kiểm tra toàn bộ hệ thống file đòi hỏi nhiều thời gian. Việc kiểm tra chỉ cho phép phát hiện lỗi sau khi đã xảy ra và không đảm bảo khôi phục dữ liệu đối với một số lỗi.
- ▶ Giao tác (transaction) là một tập hợp các thao tác cần phải được thực hiện trọn vẹn cùng với nhau. Với hệ thống file: mỗi giao tác sẽ bao gồm những thao tác thay đổi liên kết cần thực hiện cùng nhau (Ví dụ các thao tác cập nhật liên kết liên quan với một khối trên đĩa)
- ▶ Toàn bộ trạng thái hệ thống file được ghi lại trong file log
- ▶ Mỗi khi thực hiện giao tác, hệ thống kiểm tra xem giao tác có được thực hiện trọn vẹn không.
- ▶ Nếu giao tác không được thực hiện trọn vẹn, HDH sử dụng thông tin từ log để khôi phục hệ thống file về trạng thái không lỗi trước khi thực hiện giao tác

- ▶ Ngăn cản việc truy cập trái phép các thông tin lưu trữ trong file và thư mục
- ▶ Đối với hệ thống nhỏ dành cho một người dùng vấn đề bảo mật tương đối đơn giản, có thể thực hiện bằng các biện pháp vật lý => Không cho người khác tiếp cận tới hệ thống,
- ▶ Trong những hệ thống tính toán đa người dùng, việc bảo mật cho file và thư mục thực hiện bằng:
 - Kiểm soát các thao tác truy cập tới file hoặc thư mục
 - Dùng mật khẩu => Người dùng phải nhớ nhiều mật khẩu => Mỗi khi thao tác với tài nguyên lại gõ mật khẩu
- ▶ Sử dụng danh sách quản lý truy cập ACL (Access Control List)
 - Mỗi file được gán danh sách đi kèm, chứa thông tin định danh người dùng và các quyền người đó được thực hiện với file
 - ACL thường được lưu trữ như thuộc tính của file/ thư mục

- Thường được sử dụng cùng với cơ chế đăng nhập
- Các quyền truy cập cơ bản:
 - ▶ Quyền đọc (r): người có quyền này được phép đọc nội dung file
 - ▶ Quyền ghi, thay đổi (w): được phép ghi vào file, tức là thay đổi nội dung file
 - ▶ Quyền xóa: được phép xóa file, quyền này tương đương với quyền thay đổi file
 - ▶ Quyền thay đổi chủ file (change owner)
- ▶ Sử dụng danh sách quản lý truy cập ACL (Access Control List)

File 1		
A	B	C
chủ file		
đọc	đọc	đọc
ghi		ghi

File 2	
B	C
chủ file	
đọc	đọc
ghi	

- ▶ Hệ thống FAT được thiết kế để sử dụng trong HDH DOS, sau đó đã được sử dụng trong một số phiên bản của HDH Window 3.0, 3.1, 95/98.
- ▶ Hiện nay, FAT vẫn là hệ thống thông dụng, được sử dụng trong hầu hết các HDH hiện nay để quản lý thẻ nhớ, đĩa mềm, đĩa CD
- ▶ 3 phiên bản: FAT12, FAT16, FAT32
- ▶ Chữ số chỉ kích thước ô bảng FAT tương ứng 12, 16 và 32 bit
- ▶ Hiện nay, FAT32 thường được sử dụng cho đĩa cứng, FAT16 được sử dụng cho thiết bị nhớ ngoài có dung lượng nhỏ như CD, thẻ nhớ ngoài.

Chương 2

- ▶ Cấp phát không gian cho file
- ▶ Quản lý không gian trống trên đĩa
- ▶ Độ tin cậy của hệ thống file
- ▶ Bảo mật cho hệ thống file

Tiếp theo

- ▶ Hệ thống file FAT
 - Đĩa logic
 - Boot sector
 - Bảng FAT
 - Root Thư mục gốc
 - Hàm đọc đĩa
 - Bài tập thực hành