

HỆ ĐIỀU HÀNH

CHƯƠNG 2. QUẢN LÝ TẬP TIN



MỤC TIÊU

Giúp sinh viên hiểu rõ **tập tin là gì** và các **mô hình tổ chức, quản lý và cài đặt tập tin** trong các Hệ điều hành.

NỘI DUNG

CÁC KHÁI NIỆM CƠ BẢN

CÁC PHƯƠNG PHÁP TRUY CẬP TẬP TIN

Mô HÌNH TỔ CHỨC VÀ QUẢN LÝ TẬP TIN

CÀI ĐẶT HỆ THỐNG TẬP TIN

Các khái niệm cơ bản

GIỚI THIỆU

- ▶ Máy tính có thể lưu trữ thông tin trên các thiết bị lưu trữ **không bay hơi** như đĩa từ, băng từ, đĩa quang, ...
- ▶ Một trong những chức năng quan trọng của HĐH là cung cấp một **giao diện đồng nhất** để người dùng có thể **lưu trữ và truy xuất** dữ liệu trên các loại thiết bị này.
- ▶ Hệ thống cung cấp chức năng trong các HĐH này được gọi là **Hệ thống lưu trữ** hay **Hệ thống quản lý tập tin**.

CÁC KHÁI NIỆM CƠ BẢN

- ▶ **Tập tin**: là một tập hợp thông tin có liên quan **được đặt tên**, được lưu trữ trên một thiết bị lưu trữ không bay hơi.
 - ▶ Một tập tin là một **đơn vị lưu trữ luận lý**, nằm trong một không gian địa chỉ luận lý **liên tục**.
 - ▶ Hệ thống quản lý tập tin (HĐH) sẽ ánh xạ các tập tin vào các **thiết bị lưu trữ vật lý** (không gian lưu trữ **có thể không liên tục**).
- ▶ Thông thường, một tập tin chứa **dữ liệu** (văn bản, hình ảnh, ...) hoặc một **chương trình máy tính**.

CÁC LOẠI TẬP TIN

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, perl, asm	source code in various languages
batch	bat, sh	commands to the command interpreter
markup	xml, html, tex	textual data, documents
word processor	xml, rtf, docx	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	gif, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	rar, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

CÁC THAO TÁC TRÊN TẬP TIN

- ▶ **Create**: tạo mới tập tin
 - ▶ tìm **không gian lưu trữ còn trống** trên thiết bị lưu trữ
 - ▶ tạo một **bản ghi tập tin** trên cấu trúc thư mục sẽ lưu trữ tập tin
 - ▶ thông tin cần thiết cho thao tác này là **tên tập tin** và **vị trí tập tin**
- ▶ **Write**: ghi tập tin
 - ▶ lưu dữ liệu của tập tin (từ bộ nhớ) lên thiết bị lưu trữ
 - ▶ thông tin cần thiết cho thao tác này là **tên tập tin** và **dữ liệu** cần ghi

CÁC THAO TÁC TRÊN TẬP TIN

- ▶ **Read:** đọc tập tin
 - ▶ đọc dữ liệu của tập tin từ thiết bị lưu trữ ra một vùng đệm (buffer)
 - ▶ thông tin cần thiết cho thao tác này là **tên tập tin** cần đọc và **vùng đệm** lưu dữ liệu
- ▶ **Delete:** xóa tập tin
 - ▶ xóa mẫu tin tập tin trong cấu trúc thư mục và giải phóng không gian lưu trữ đã cấp phát cho tập tin
 - ▶ thông tin cần thiết cho thao tác này là **tên tập tin** cần xóa
 - ▶ một số hệ điều hành cung cấp hai thao tác xóa: **luận lý** hay **vật lý**

CÁC THAO TÁC TRÊN TẬP TIN

- ▶ **Truncate:** xóa nội dung tập tin
 - ▶ giải phóng toàn bộ không gian lưu trữ của tập tin, chỉ giữ lại bản ghi tập tin trên cấu trúc thư mục
 - ▶ thông tin cần cho thao tác này là **tên tập tin** cần được xóa nội dung
- ▶ **Seek/Reposition:** di chuyển con trỏ vị trí hiện hành của 1 tập tin
 - ▶ các thao tác đọc/ghi trên tập tin được thực hiện thông qua **con trỏ vị trí hiện hành của tập tin** (current-file-position pointer), vị trí sẽ được truy xuất kế tiếp
 - ▶ thông tin cần cho thao tác này là **vị trí** cần di chuyển con trỏ đến (tương đối/tuyệt đối)

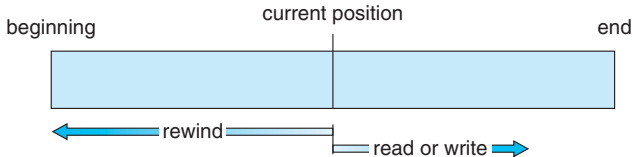
CÁC THAO TÁC TRÊN TẬP TIN

- ▶ **Open:** mở tập tin
 - ▶ Các thao tác trên tập tin cần phải thông qua bước tìm kiếm tập tin
⇒ hao phí thời gian
 - ▶ Vì vậy, trước khi thao tác trên tập tin thì phải thực hiện **mở tập tin**
 - ▶ tìm kiếm tập tin
 - ▶ tạo một **mục** trong **bảng các tập tin đang mở** cho tập tin
 - ▶ trả về **con trỏ** đến mục của tập tin trong bảng các tập tin đang mở
 - ▶ Khi cần thao tác trên tập tin thì chỉ cần sử dụng **con trỏ tập tin**
- ▶ **Close:** đóng tập tin
 - ▶ loại bỏ mục của tập tin tương ứng trong bảng các tập tin đang mở

Các phương pháp truy cập tập tin

TRUY CẬP TUẦN TỰ

- ▶ Truy cập dữ liệu **theo thứ tự**, từng mẫu tin (record).
- ▶ Các thao tác: **read next** (đọc phần tử kế tiếp), **write next** (ghi phía sau con trỏ vị trí hiện hành), **reset** (cho con trỏ hiện hành về đầu tập tin)



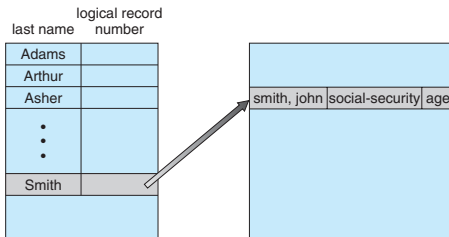
TRUY CẬP TRỰC TIẾP

- ▶ Truy cập các mẫu tin bất kỳ, **không theo thứ tự**.
- ▶ Thường áp dụng đối với các tập tin có **kích thước mẫu tin cố định** (để có thể tính được vị trí của mẫu tin cần truy xuất)
- ▶ Các chỉ thị tương tương giữa hai phương pháp truy cập:

sequential access	implementation for direct access
reset	cp = 0;
read_next	read cp ; cp = cp + 1;
write_next	write cp ; cp = cp + 1;

TRUY CẬP THEO CHỈ MỤC

- ▶ Dựa trên phương pháp truy cập trực tiếp
- ▶ Xây dựng thêm hệ thống chỉ mục (index) cho dữ liệu trên tập tin
- ▶ Hệ thống chỉ mục bao gồm mục từ và vị trí của mục từ trong tập tin
- ▶ Việc tìm kiếm trong tập tin sẽ dựa vào chỉ mục trước để tìm ra vị trí



Mô hình tổ chức và Quản lý tập tin

CẤU TRÚC CỦA MỘT TẬP TIN

- ▶ Một tập tin là một dãy các bit, bytes, dòng, mẫu tin (records),... mang ý nghĩa được định nghĩa bởi người tạo ra.
- ▶ Cấu trúc một tập tin:
 - ▶ đơn giản: bao gồm các dòng có chiều dài cố định hay thay đổi.
 - ▶ phức tạp: các tập tin có khuôn dạng là các bản ghi.
- ▶ Yếu tố quyết định cấu trúc tập tin: người dùng (users) hay hệ điều hành.

CÁC THUỘC TÍNH CỦA MỘT TẬP TIN

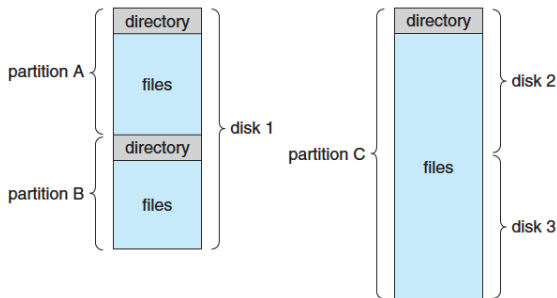
- ▶ **Tên** (name): thông tin dưới dạng người dùng có thể đọc được và có thể phân biệt chữ hoa, chữ thường (Linux, MacOS)
- ▶ **Kiểu** (type): cho phép HĐH và người dùng xác định loại tập tin (văn bản, chương trình,...). Thường xác định bằng phần mở rộng.
- ▶ **Vị trí** (position): trỏ tới vị trí của tập tin trên thiết bị lưu trữ.
- ▶ **Kích thước** (size): kích thước hiện hành của tập tin (bytes,...).
- ▶ **Bảo vệ** (protection): chỉ định ai có thể truy cập tập tin (đọc, ghi,...)
- ▶ **Thời gian** (time): thời điểm tạo, sửa đổi, sử dụng lần sau cùng.
- ▶ **Người sở hữu** (owner): định danh của người sở hữu tập tin

CÁC THUỘC TÍNH CỦA MỘT TẬP TIN

- ▶ Các thuộc tính của tập tin được lưu trong một cấu trúc được gọi là **bản ghi tập tin** (file record).
 - ▶ Thông thường, bản ghi tập tin chỉ lưu trữ tên và định danh tập tin.
 - ▶ Định danh tập tin sẽ xác định nơi lưu trữ các thuộc tính còn lại.
- ▶ Các bản ghi tập tin được lưu trữ trong một **cấu trúc thư mục** (directory structure).
- ▶ Cấu trúc thư mục cũng được lưu trữ trên thiết bị lưu trữ.

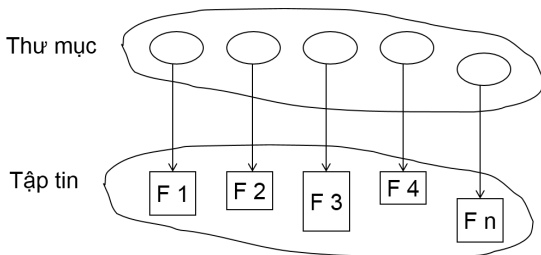
TỔ CHỨC CỦA MỘT HỆ THỐNG TẬP TIN

- ▶ Đĩa có thể được phân chia ra thành nhiều **phân vùng** (partition)
- ▶ Hoặc ngược lại, nhiều đĩa có thể được kết hợp thành 1 phân vùng
- ▶ Các tập tin có liên quan với nhau được tổ chức trong 1 **thư mục**



THƯ MỤC

- ▶ Thư mục là cấu trúc chứa các tập tin có liên quan với nhau.
- ▶ Mỗi tập tin thuộc thư mục sẽ có 1 nút (node, chính là bản ghi tập tin) tương ứng trong cấu trúc thư mục để lưu trữ thông tin về tập tin.
- ▶ Cấu trúc thư mục cũng được lưu trên thiết bị lưu trữ.
- ▶ **Mục tiêu:**
 - ▶ tổ chức, quản lý tập tin hiệu quả
 - ▶ định vị tập tin nhanh chóng

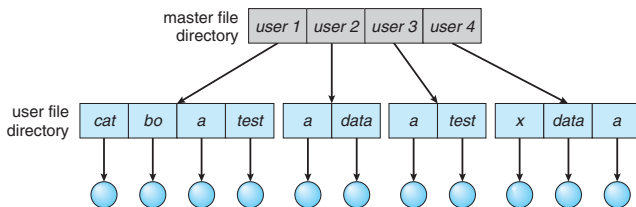


CÁC THAO TÁC TRÊN THƯ MỤC

- ▶ **Search** for a file: tìm kiếm một tập tin trong thư mục.
- ▶ **Create** a file: tạo một tập tin trên thư mục
- ▶ **Delete** a file: xóa một tập tin trên thư mục
- ▶ **List** a directory: liệt kê nội dung thư mục (danh sách các tập tin)
- ▶ **Rename** a file: thay đổi tên tập tin
- ▶ **Travel** the file system: duyệt qua danh sách các tập tin trong thư mục

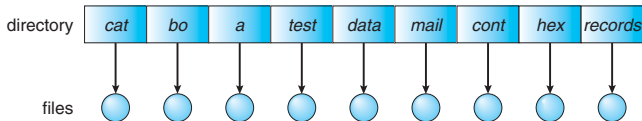
CẤU TRÚC THƯ MỤC 1 CẤP

- ▶ Là cấu trúc đơn giản nhất: **tất cả các file** nằm trong **cùng 1 thư mục**.
- ▶ Nhược điểm:
 - ▶ trong môi trường **nhiều người dùng** thì khả năng trùng tên cao.
 - ▶ ngay cả trong môi trường **đơn người dùng**, nếu số lượng tập tin lớn thì khả năng trùng tên cũng cao và khó quản lý (không thể gom nhóm).



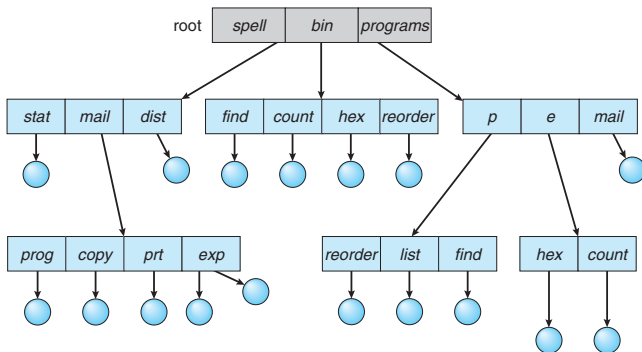
CẤU TRÚC THƯ MỤC 2 CẤP

- ▶ Mỗi người dùng sẽ có 1 thư mục riêng.
- ▶ Thư mục của mỗi người dùng là thư mục 1 cấp.
- ▶ Cho phép nhiều người dùng có thể dùng chung 1 tên tập tin.
- ▶ Tuy nhiên, vẫn tồn tại vấn đề về gom nhóm và trùng tên tập tin khi số lượng tập tin của mỗi người dùng lớn.



CẤU TRÚC THƯ MỤC CÂY

- ▶ Là sự tổng quát hóa của cấu trúc TM 2 cấp với **số cấp không giới hạn**.
- ▶ Được sử dụng trong hầu hết các HĐH hiện nay.

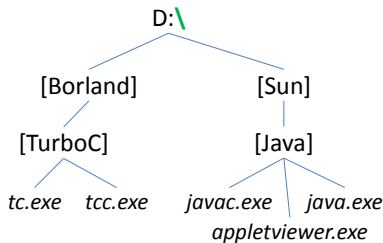
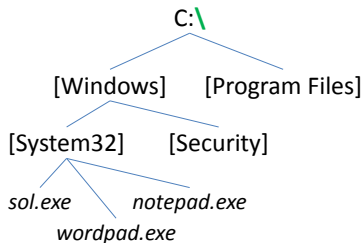


CẤU TRÚC THƯ MỤC CÂY

- ▶ Mỗi thư mục, ngoài các tập tin, có thể chứa các thư mục khác, được gọi là các **thư mục con**.
- ▶ Thư mục chứa các thư mục khác được gọi là **thư mục cha**.
- ▶ Quan giữa các thư mục cha-con tạo thành **1 cây thư mục**.
- ▶ Khi thao tác trên hệ thống tập tin, luôn tồn tại 1 **thư mục hiện hành**.
- ▶ Có 2 cách **định vị 1 tập tin** hay thư mục:
 - ▶ **đường dẫn tương đối**: tính từ vị trí của thư mục hiện hành.
 - ▶ **đường dẫn tuyệt đối**: tính từ **gốc** của cây thư mục (**thư mục gốc**) \Rightarrow không phụ thuộc vào vị trí thư mục hiện hành.

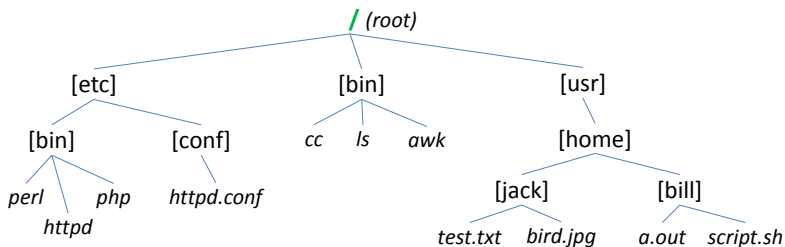
CẤU TRÚC THƯ MỤC CÂY – WINDOWS

- ▶ Một hệ thống lưu trữ bao gồm 1 hoặc nhiều **phân vùng** (partition).
- ▶ Mỗi phân vùng được gán 1 tên gọi là tên **ổ đĩa** (drive), là 1 **chữ cái**.
- ▶ Ký hiệu "****" được sử dụng để biểu diễn cho thư mục gốc của mỗi ổ đĩa.



CẤU TRÚC THƯ MỤC CÂY – UNIX

- ▶ Hệ thống cây thư mục **chỉ có 1 gốc**, ký hiệu là **" / "**.
- ▶ Tất cả các phân vùng đều được tổ chức trong cây thư mục này.
- ▶ Thao tác ánh xạ một phân vùng vào một mục trên cây thư mục này được gọi là **mount**.



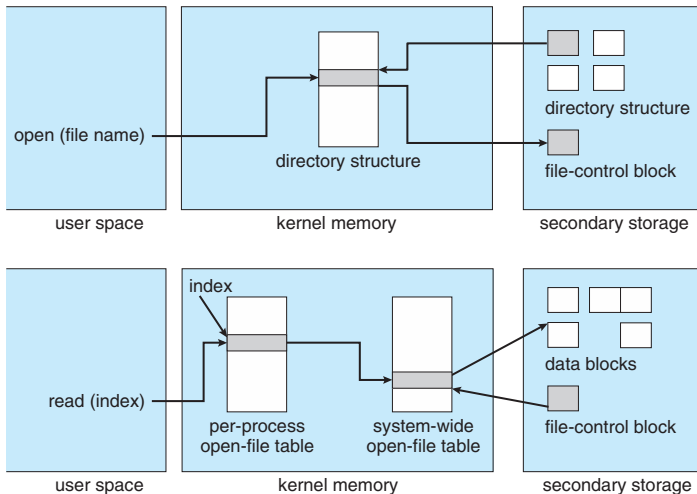
Cài đặt hệ thống tập tin

KHỐI ĐIỀU KHIỂN TẬP TIN – FCB

- ▶ Hiện có nhiều **hệ thống tập tin** được sử dụng bởi các HĐH (FAT, NTFS, UFS,...)
- ▶ Các tập tin được quản lý bằng các **khối điều khiển tập tin** (file control block, FCB), chứa các thông tin về tập tin.

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

VAI TRÒ CỦA KHỐI ĐIỀU KHIỂN TẬP TIN



CÀI ĐẶT HỆ THỐNG THƯ MỤC

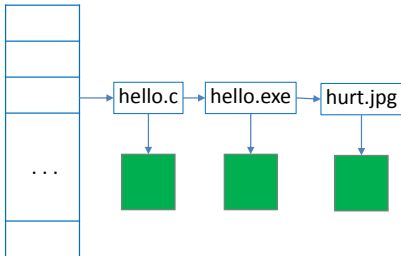
- ▶ Có 2 phương pháp tổ chức và truy xuất các FCB trong một thư mục:
 1. Danh sách tuyến tính các con trỏ tới các khối dữ liệu:
 - ▶ Cấu trúc cài đặt và xử lý đơn giản.
 - ▶ Chi phí thực hiện các thao tác trên thư mục cao.
 2. Bảng băm (hash) hay bảng băm với danh sách tuyến tính:
 - ▶ Giảm chi phí duyệt thư mục.
 - ▶ Đòi hỏi có hàm băm hiệu quả (cân bằng).
 - ▶ Phải tổ chức lại hàm cấu trúc thư mục khi hàm băm thay đổi.

CÀI ĐẶT HỆ THỐNG THƯ MỤC

danh sách (liên kết) tuyến tính



bảng băm

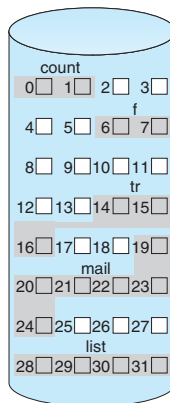


CÁC PHƯƠNG PHÁP CẤP PHÁT ĐĨA

- ▶ Là cách thức lưu nội dung vào các khối đĩa như thế nào.
- ▶ Có 3 phương pháp chính:
 1. Cấp phát kề nhau.
 2. Cấp phát kiểu liên kết.
 3. Cấp phát kiểu chỉ mục.

PHÁP CẤP KÈ NHAU

- ▶ Dữ liệu tập tin được lưu trong các khối đĩa **liên tục, kề nhau**
- ▶ Đây là phương pháp tổ chức đơn giản.
- ▶ Cho phép truy cập **tuần tự** lẫn **ngẫu nhiên**.
- ▶ **Nhược điểm:**
 - ▶ Dễ gây **phân mảnh** ngoài.
 - ▶ Khó **tăng kích thước** tập tin.

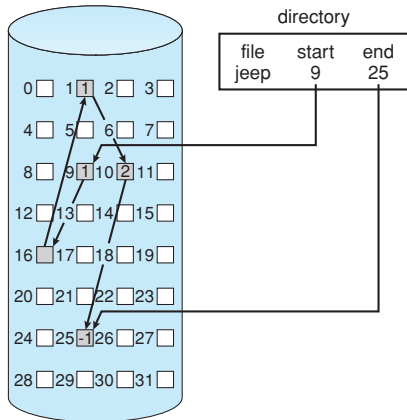


directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

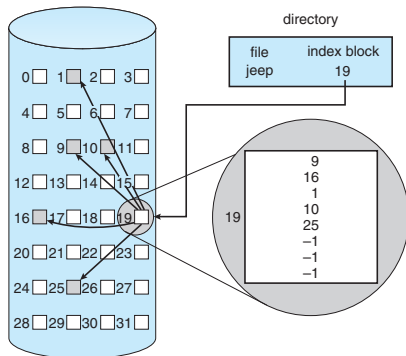
PHÁP CẤP KIỂU LIÊN KẾT

- ▶ Dữ liệu tập tin được lưu trong các khối đĩa **không liên tục**.
- ▶ Cuối mỗi khối là một **con trỏ**, trỏ đến khối tiếp theo (trừ khối cuối).
- ▶ Đây cũng là phương pháp tổ chức **đơn giản** và **tránh phân mảnh ngoài**.
- ▶ **Nhược điểm:**
 - ▶ Không thể truy cập ngẫu nhiên.
 - ▶ Mất không gian cho các con trỏ.
- ▶ **Áp dụng:** hệ thống tập tin **FAT**.



PHÁP CẤP KIỂU CHỈ MỤC

- ▶ Dữ liệu tập tin được lưu trong các khối đĩa **không liên tục**.
- ▶ Tất cả các khối của tập tin được lưu trữ trong 1 khối, gọi là **khối chỉ mục**.
- ▶ Phần tử thứ i trong khối chỉ mục trỏ tới khối dữ liệu thứ i của tập tin.
- ▶ Cho phép truy xuất **ngẫu nhiên**.
- ▶ Tránh phân mảnh ngoài nhưng tốn thêm khối chỉ mục.
- ▶ **Áp dụng:** **UFS** với một số cải tiến.



QUẢN LÝ KHÔNG GIAN TRỐNG

- ▶ Là cách thức quản lý các khối đĩa còn trống.
- ▶ Mục tiêu: dễ dàng tìm khối trống để cấp phát cho tập tin lần việc cập nhật danh sách các khối trống khi tập tin bị xóa.
- ▶ Có 4 phương pháp chính:
 1. Bản đồ bit (bit vector).
 2. Danh sách liên kết (linked list).
 3. Nhóm (grouping).
 4. Bộ đếm (counter).

BẢN ĐỒ BIT

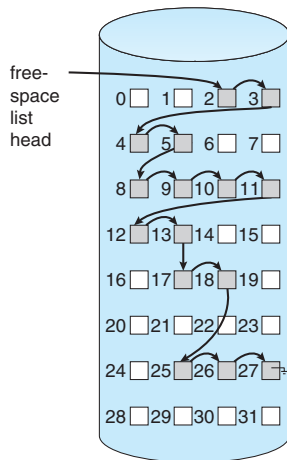
- ▶ Dùng một **vector các bit** (gọi là bản đồ bit) để đánh dấu một khối còn trống hay không.

0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 ...

- ▶ Mỗi khối đĩa được đại diện bằng 1 bit trong vector (0: còn trống, 1: đã cấp phát).
 - ▶ Bản đồ bit yêu cầu thêm không gian lưu trữ:
 - ▶ block size = 1MB (2^{12} bytes)
 - ▶ disk size = 1GB (2^{30} bytes)
- ⇒ Kích thước bản đồ bit = $2^{30}/2^{12} = 2^{18}$ bytes (32KB)

DANH SÁCH LIÊN KẾT

- ▶ Các khối trống được liên kết với nhau bằng một danh sách liên kết.
- ▶ Lưu trữ một con trỏ trỏ đến khối trống đầu tiên trong danh sách (free-space list head).
- ▶ **Nhược điểm:** khó cấp phát liên tục.



NHÓM & ĐẾM

- ▶ Nhóm (grouping):

- ▶ Là một biến thể của danh sách liên kết.
- ▶ Lưu trữ một danh sách liên kết các khối chỉ mục.
- ▶ Mỗi khối chỉ mục lưu địa chỉ một số khối trống và có con trỏ đến khối chỉ mục chứa danh sách các khối trống tiếp theo.

- ▶ Đếm (counting):

- ▶ Dùng quản lý các khối trống kề nhau.
- ▶ Gồm một số các mục từ (entry), mỗi mục từ lưu vị trí khối trống đầu tiên và số lượng n khối trống tiếp theo.

TỔNG KẾT

CÁC KHÁI NIỆM CƠ BẢN

CÁC PHƯƠNG PHÁP TRUY CẬP TẬP TIN

MÔ HÌNH TỔ CHỨC VÀ QUẢN LÝ TẬP TIN

CÀI ĐẶT HỆ THỐNG TẬP TIN