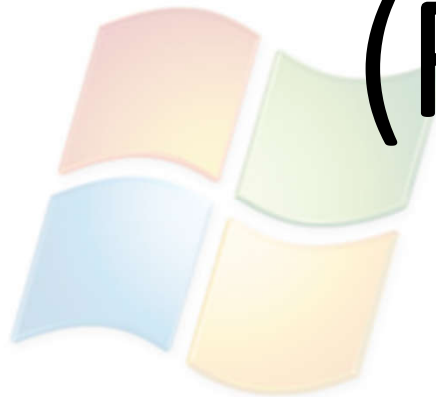


Chương 8

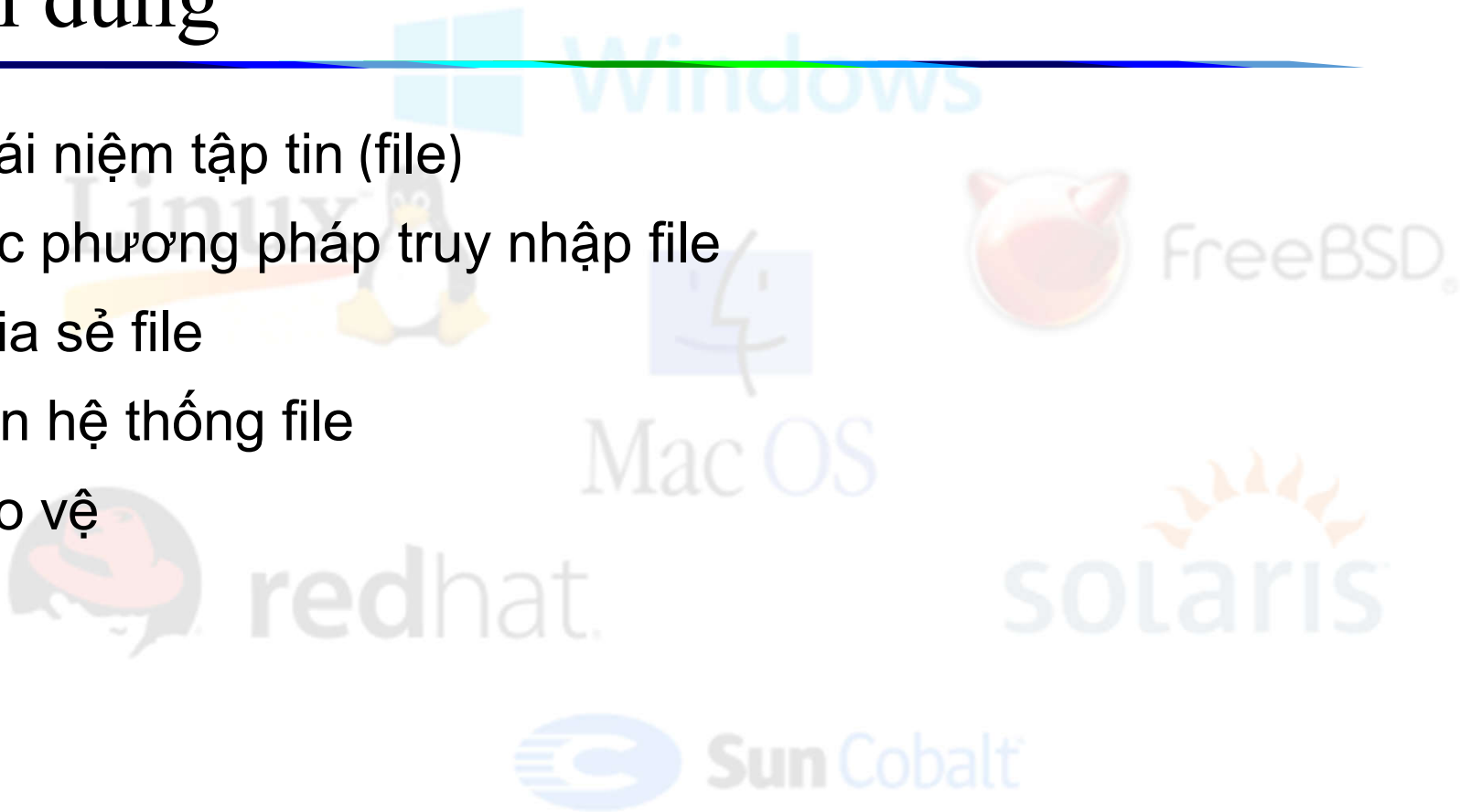
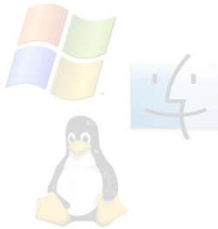
HỆ THỐNG TẬP TIN

(File System)

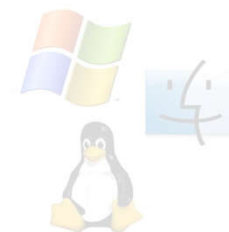


Nội dung

- Khái niệm tập tin (file)
- Các phương pháp truy nhập file
- Chia sẻ file
- Gắn hệ thống file
- Bảo vệ

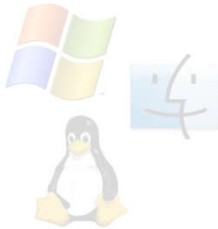


Hệ thống tập tin



- Hệ thống tập tin cung cấp kho lưu trữ lâu dài cho chương trình và dữ liệu.
 - Để đảm bảo việc lưu trữ lâu dài, các tập tin được lưu trữ trên đĩa
 - Để có thể sử dụng nội dung tập tin, CPU đọc tập tin vào bộ nhớ trong
- Hệ thống tập tin bao gồm
 - Một tập các tập tin
 - Một cấu trúc thư mục

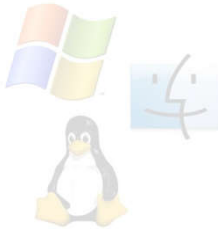




Khái niệm tập tin

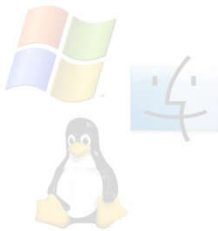
- Một tập dữ liệu được lưu trữ trong thiết bị lưu trữ thứ cấp.
- Thông thường, một tập tin lưu trữ chương trình hoặc dữ liệu
- Một chuỗi các bit, byte, dòng hoặc bản ghi
 - Ý nghĩa của các bản ghi này được định nghĩa bởi người tạo
- Hệ điều hành trừu tượng hóa chi tiết của mỗi thiết bị lưu trữ
- Người dùng thấy một mảng tuyến tính các bản ghi
- Hệ điều hành ánh xạ tập tin luận lý lên thiết bị lưu trữ

Cấu trúc tập tin



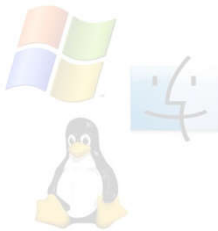
- Không có cấu trúc – chuỗi các từ (word), các byte
- Cấu trúc bản ghi đơn giản
- Các dòng
- Kích cỡ cố định
- Kích cỡ thay đổi
- Các cấu trúc phức tạp
- Tài liệu được định dạng
- Tập tin có thể định vị lại được
- Có thể cài đặt cấu trúc phức tạp bằng cách thêm một số kí tự điều khiển vào tập tin

Những thông tin về file mà HĐH quản lý



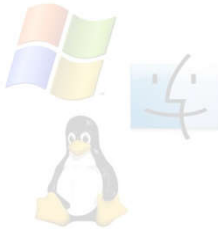
- File có các thuộc tính:
 - Tên (name)
 - Số định danh (identifier)
 - Kiểu (type)
 - Vị trí (location)
 - Kích thước(size)
 - Bảo vệ (protection) : thông tin điều khiển truy cập
 - OwnerID
 - Thời gian: tạo, sửa đổi lần cuối, truy cập lần cuối

Ví dụ về khối điều khiển file trong Linux



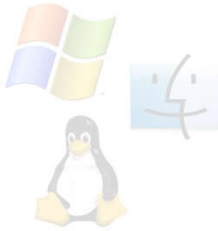
```
struct stat {  
    dev_t      st_dev;      /* device */  
    ino_t      st_ino;      /* inode */  
    mode_t     st_mode;     /* protection */  
    nlink_t    st_nlink;    /* number of hard links to file */  
    uid_t      st_uid;      /* user-id of owner */  
    gid_t      st_gid;      /* group-id of owner */  
    dev_t      st_rdev;     /* device type, for special file */  
    struct timespec st_atimespec; /* time of last access */  
    struct timespec st_mtimespec; /* time of last data mod */  
    struct timespec st_ctimespec; /* time of last status change */  
    off_t       st_size;     /* file size, in bytes */  
    quad_t      st_blocks;   /* blocks allocated for file */  
    u_long      st_blksize;  /* optimal file sys I/O ops blocksize */  
    u_long      st_flags;    /* user defined flags for file */  
    u_long      st_gen;      /* file generation number */  
};
```

Các thao tác trên file

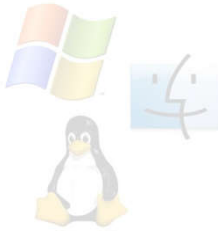


- File là một kiểu dữ liệu trừu tượng
- Thông thường, có các thao tác trên file sau
 - create(): tìm không gian trong hệ thống cho một file và sau đó thêm nó vào thư mục
 - write(): thêm dữ liệu vào file tại vị trí hiện tại
 - read(): đọc dữ liệu từ file bắt đầu từ vị trí hiện tại
 - seek(): thay đổi vị trí con trỏ đọc hoặc ghi đến một vị trí xác định trong file
 - delete(): xóa một file khỏi hệ thống file

Các file mở



- Các thông tin để quản lý các file đang mở:
 - Con trỏ file: trỏ đến vị trí đọc/ ghi cuối
 - Thông tin đơn tiến trình
 - Số lần mở file – Khi các tiến trình mở file thoát → xóa phần tử tương ứng với file trong bảng file mở
 - Thông tin hệ thống
 - Vị trí đĩa: Lưu lại thông tin truy nhập dữ liệu
 - Các quyền truy nhập: mode truy nhập đối với mỗi tiến trình



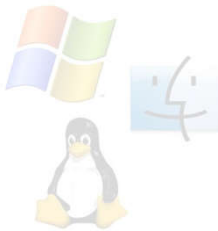
Khóa các file mở

- Một số hệ điều hành và hệ thống file hỗ trợ khóa các file mở
- Sắp xếp việc truy nhập file
- Bắt buộc/ Tự vấn:
 - Bắt buộc – truy vấn bị từ chối tùy thuộc khóa và yêu cầu
 - Tự vấn – các tiến trình kiểm tra trạng thái của khóa và xác định

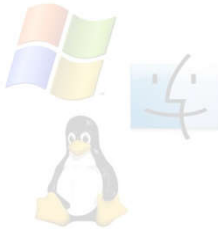


Ví dụ khóa file trong Java

```
import java.io.*;
import java.nio.channels.*;
public class LockingExample{
    public static final boolean EXCLUSIVE = false;
    public static final boolean SHARED = true;
    public static void main(String[] args) throws IOException{
        FileLock sharedLock = null;
        FileLock exclusiveLock = null;
        try {
            RandomAccessFile raf = new RandomAccessFile("file.txt", "rw");
            // get the channel for the file
            FileChannel ch = raf.getChannel();
            // this locks the first half of the file -exclusive
            exclusiveLock = ch.lock(0, raf.length()/2, EXCLUSIVE);
            /** Now modify the data . . . */
            // release the lock
            exclusiveLock.release();
        }
    }
}
```



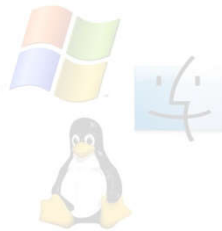
Ví dụ khóa file trong Java



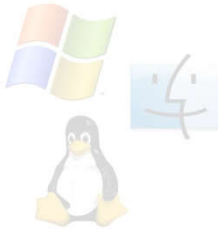
```
// this locks the second half of the file -shared
sharedLock= ch.lock(raf.length()/2+1, raf.length(),
SHARED);
/** Now read the data . . . */
// release the lock
exclusiveLock.release();
}catch (java.io.IOException ioe) {
    System.err.println(ioe);
}finally {
    if (exclusiveLock != null)
        exclusiveLock.release();
    if (sharedLock != null)
        sharedLock.release();
}
```

```
}
```

Các kiểu file – 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

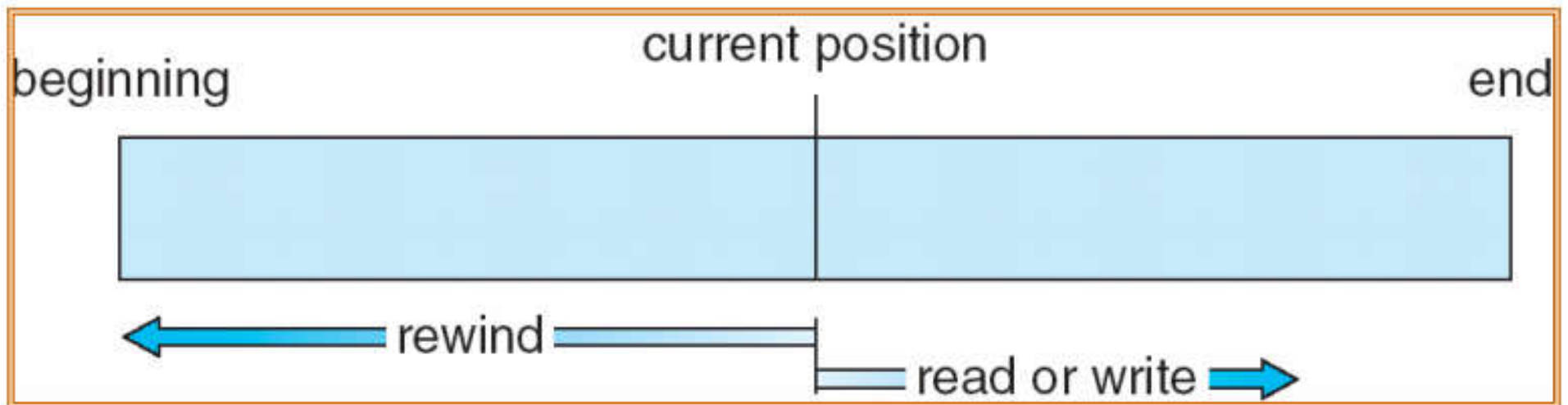
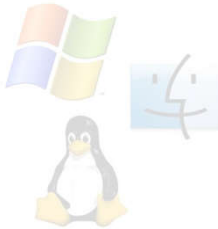


Các phương pháp truy cập

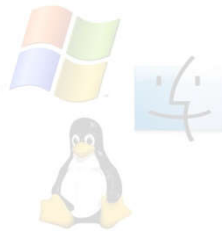
- Truy cập tuần tự
 - read next
 - write next
 - reset
 - no read after last write
 - (rewrite)
- Truy cập trực tiếp
 - read n
 - write n
 - position to n
 - read next
 - write next
 - rewrite n
 - $n =$ số hiệu tương đối của khối



File truy nhập tuần tự



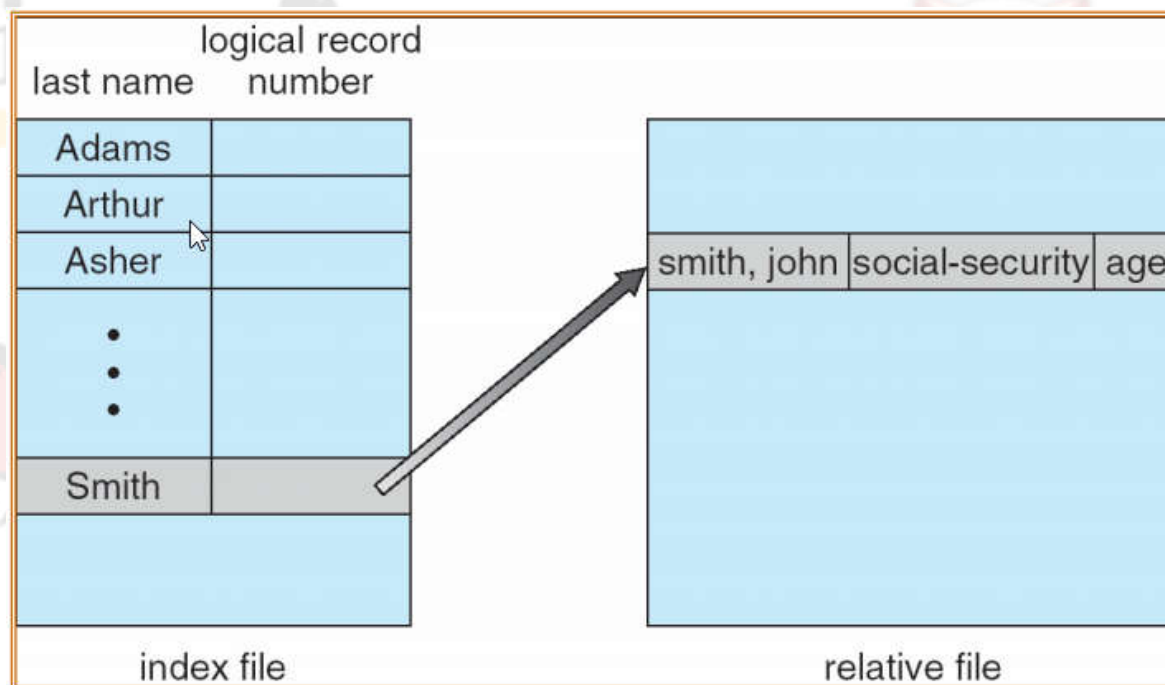
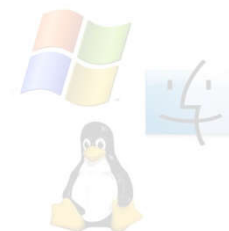
Mô phỏng truy cập tuần tự trên một file truy nhập trực tiếp



sequential access	implementation for direct access
<i>reset</i>	<i>cp = 0;</i>
<i>read next</i>	<i>read cp;</i> <i>cp = cp + 1;</i>
<i>write next</i>	<i>write cp;</i> <i>cp = cp + 1;</i>

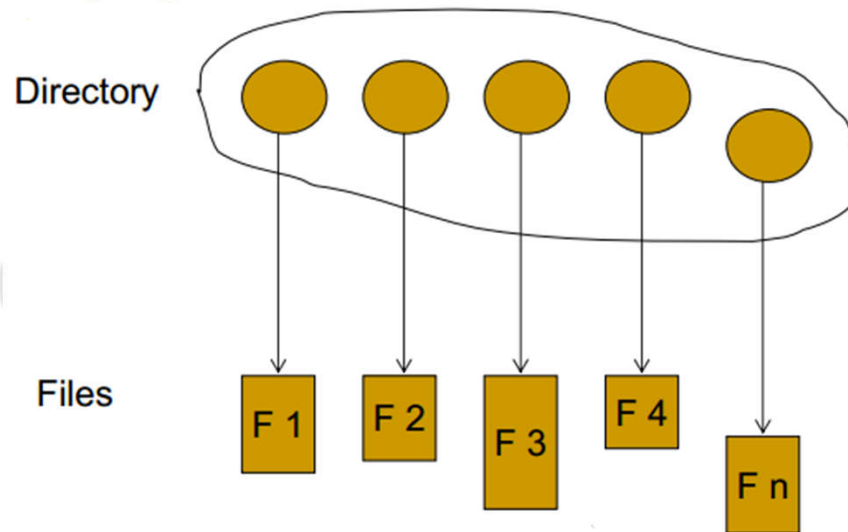


Ví dụ về chỉ số và các file tương đối

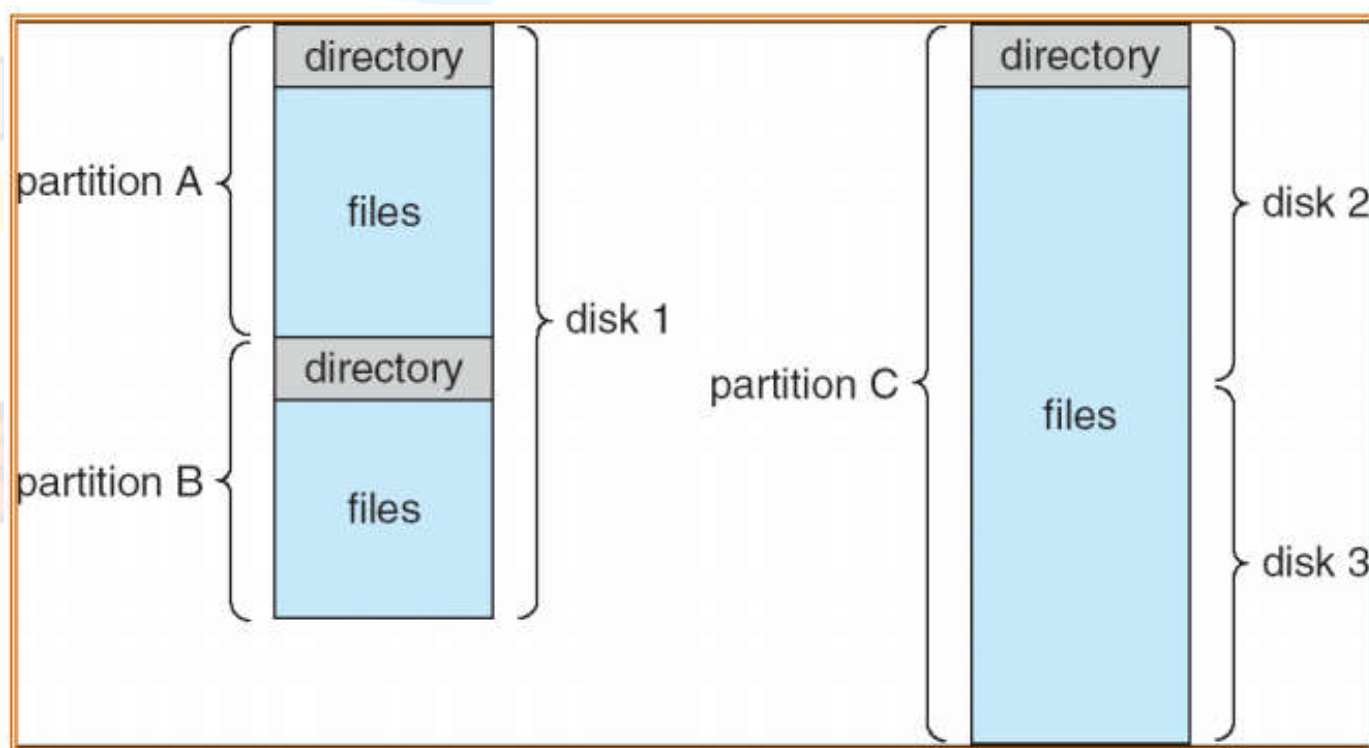
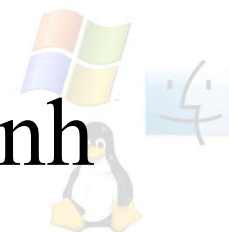


Cấu trúc thư mục

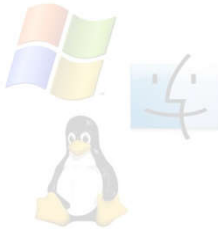
- Các đĩa thường được tổ chức thành các phân vùng.
- Các thư mục thu thập và tổ chức các file trên một phân vùng



Cách thức tổ chức một hệ thống file điển hình



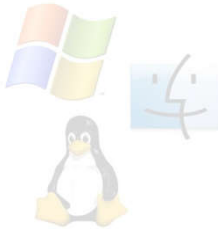
Các thao tác trên thư mục



- search: tìm một file hoặc tập các file khớp với điều kiện tìm kiếm
- create: tạo một file trên một thư mục
- delete: xóa một file khỏi một thư mục
- list: xem nội dung thư mục
- rename: thay đổi tên của một file
- traverse



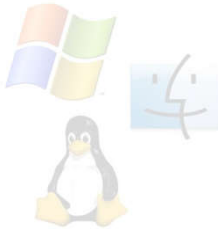
Tổ chức thư mục (mức luận lý)



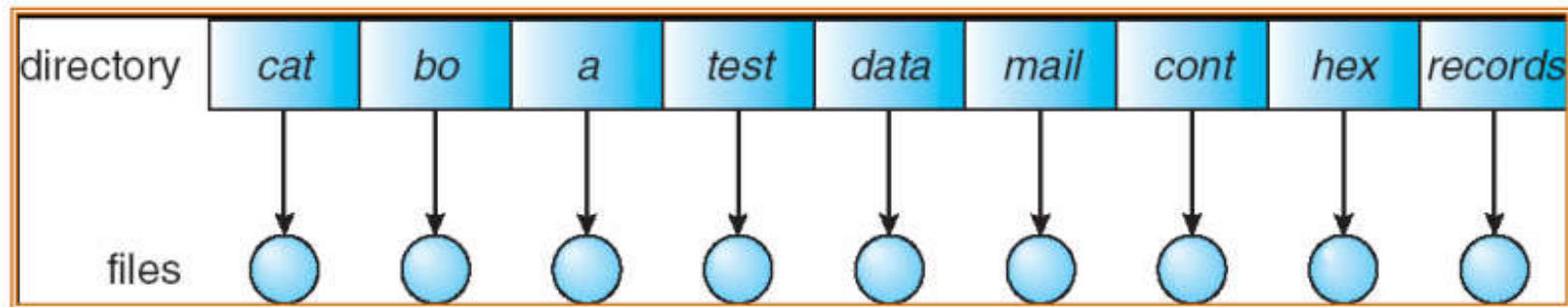
- Hiệu quả
 - Xác định vị trí các file một cách nhanh chóng
- Đặt tên – thuận tiện cho người dùng
 - Hai người dùng có thể dùng cùng 1 tên cho hai file khác nhau
 - File giống nhau có thể có các tên khác nhau
- Gộp nhóm – gộp các file có cùng đặc trưng lại thành các nhóm (ví dụ: các file chương trình java, tất cả các trò chơi, ...)



Cấu trúc đơn mức



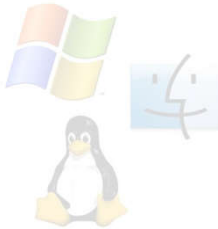
- Một thư mục đơn cho tất cả người dùng



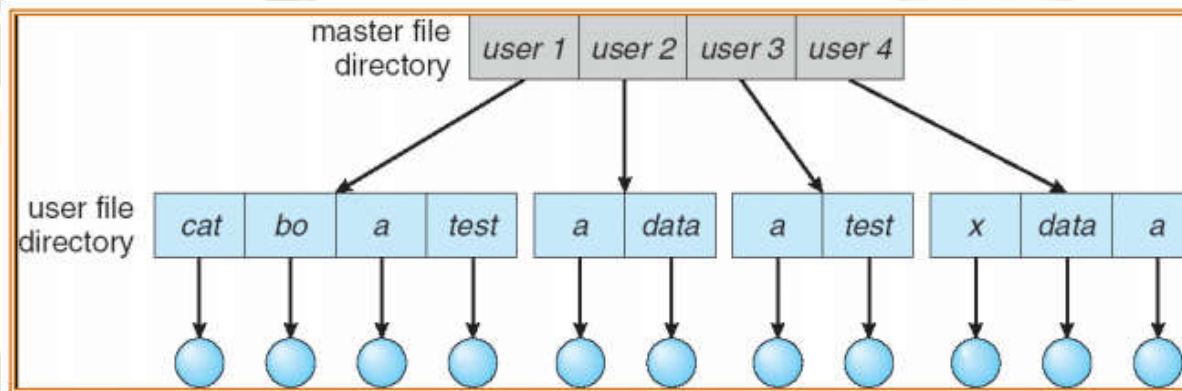
- Vấn đề
 - Đặt tên
 - Gộp nhóm



Cấu trúc hai mức

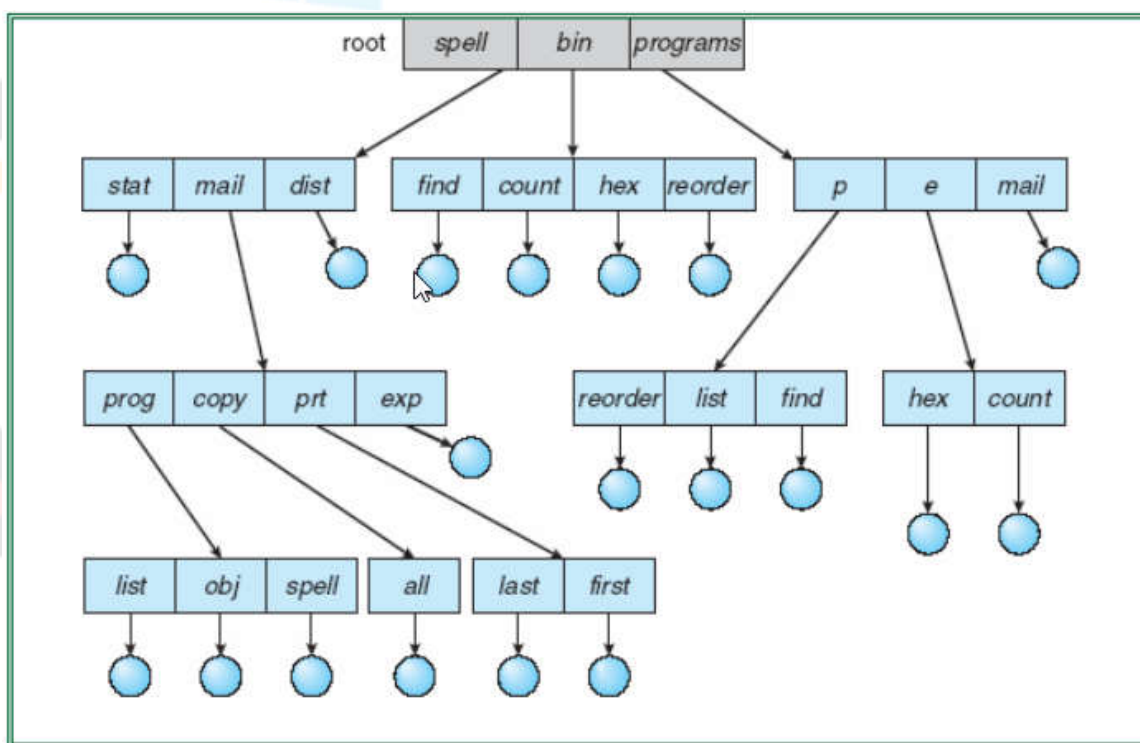
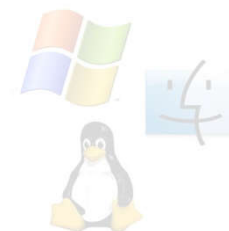


- Mỗi người dùng có một thư mục



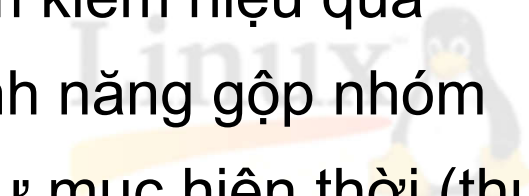
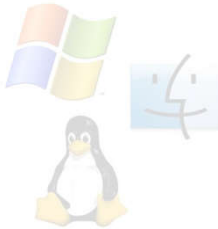
- Đường dẫn
- Cho phép hai người dùng khác nhau đặt cùng một tên file
- Tìm kiếm hiệu quả
- Chưa có tính năng gộp nhóm

Cấu trúc cây

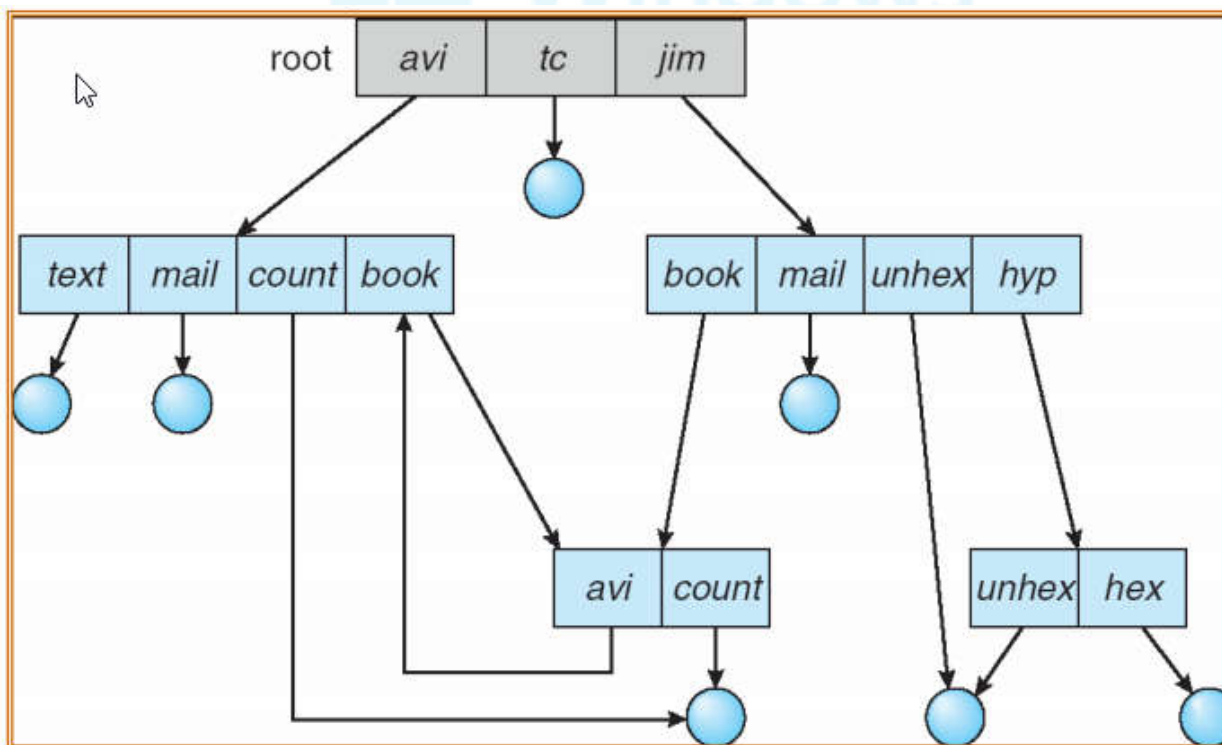
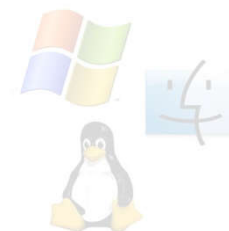


Cấu trúc cây

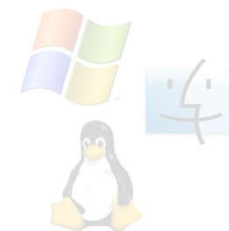
- Tìm kiếm hiệu quả
- Tính năng gộp nhóm
- Thư mục hiện thời (thư mục làm việc)
 - `cd /spell/mail/prog`
 - `type list`



Cấu trúc đồ thị không có chu trình



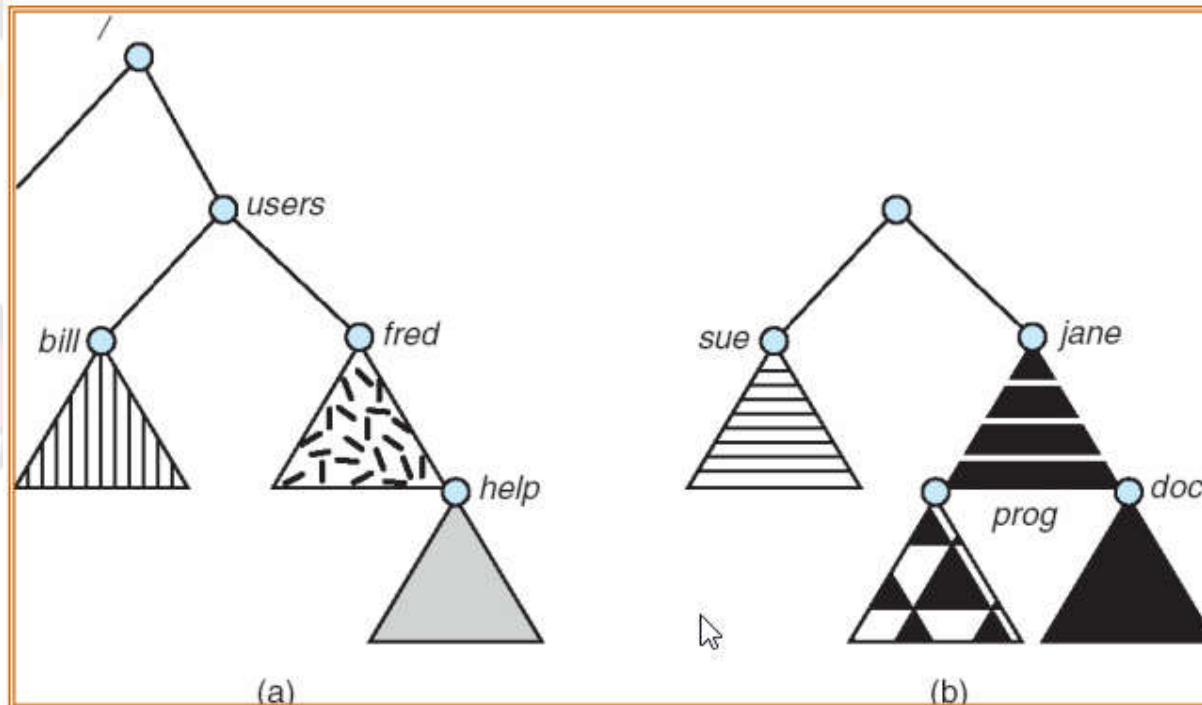
Cấu trúc đồ thị không có chu trình



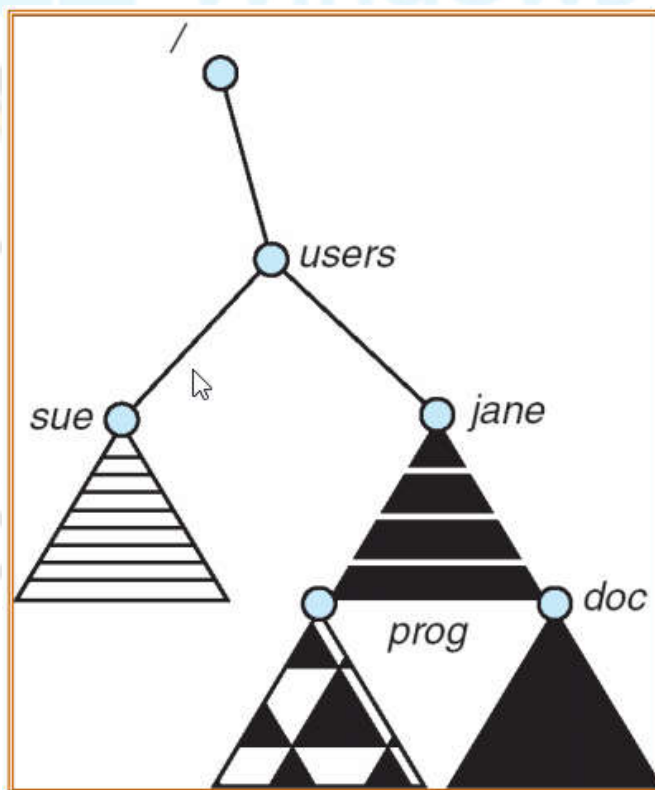
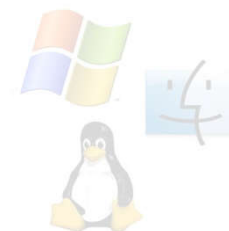
- Làm sao đảm bảo không có chu trình?
 - Chỉ cho phép tạo liên kết đến file mà không cho tạo liên kết đến các thư mục con
 - Thu rác (garbage collection)
 - Mỗi khi một liên kết mới được thêm, sử dụng một thuật toán xác định chu trình để kiểm tra xem có thêm được không

Gắn hệ thống file

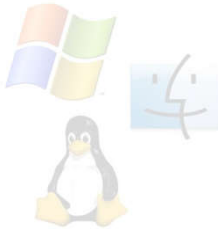
- Một hệ thống file cần được gắn trước khi có thể truy cập



Điểm gắn hệ thống file



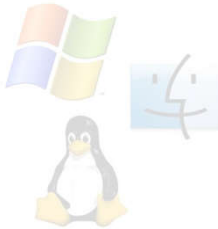
Chia sẻ file



- Nhu cầu chia sẻ file trong hệ thống đa người dùng
- Thực hiện chia sẻ file thông qua lược đồ bảo vệ
- Trên các hệ thống phân tán, các file có thể được chia sẻ qua mạng
- Hệ thống file mạng (NFS) là phương pháp chia sẻ file phổ biến trong Unix, Linux



Chia sẻ file – đa người dùng



- User Ids nhận diện người dùng, cho phép cấp quyền và bảo vệ file cấp người dùng.
- Group Ids xác định nhóm người dùng, cấp quyền truy nhập theo nhóm



redhat.

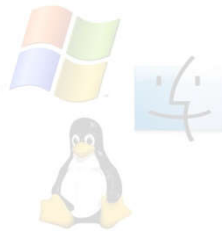
Mac OS

solaris



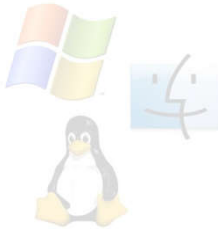
Sun Cobalt

Chia sẻ file – Các hệ thống file từ xa



- Thông qua mạng để truy cập hệ thống file giữa các hệ thống
 - Truy cập thủ công thông qua các chương trình như FTP
 - Truy nhập tự động thông qua hệ thống file chia sẻ
 - Truy nhập bán tự động thông qua world wide web
- Mô hình client – server cho phép khách gắn các hệ thống file của server
 - Server có thể phục vụ nhiều client
 - Việc nhận dạng người dùng trên máy client thường không bảo mật hoặc phức tạp
 - NFS là giao thức chia sẻ file client - server chuẩn của UNIX
 - CIFS là giao thức chuẩn của Windows.
 - Các lời gọi file chuẩn của hệ điều hành được chuyển thành các lời gọi file từ xa.
- Các hệ thống thông tin phân tán (các dịch vụ định dạng phân tán) như LDAP, DNS, NIS, Active Directory thiết lập truy cập hợp nhất đến thông tin chia sẻ từ xa

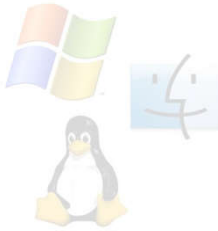
Chia sẻ file – Các mode thất bại



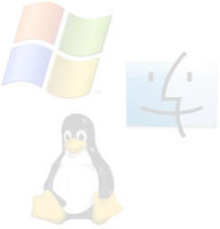
- Các hệ thống file từ xa có thêm các mode thất bại, do mạng, do server
- Khôi phục từ thất bại cần thông tin trạng thái của mỗi yêu cầu từ xa.
- Các giao thức không hướng kết nối như NFS trong mỗi yêu cầu cho phép khôi phục dễ dàng hơn nhưng ít bảo mật hơn



Chia sẻ file – Tính nhất quán về mặt ngữ nghĩa



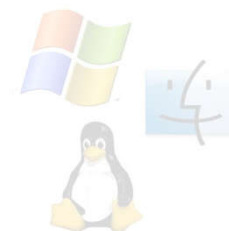
- Tính nhất quán về ngữ nghĩa xác định cách thức cho phép nhiều người dùng truy cập vào file cùng lúc.
 - Tương tự cách đồng bộ tiến trình cộng tác
 - Hệ thống file Andrew (AFS) triển khai ngữ cảnh chia sẻ file từ xa
 - Hệ thống file Unix (UFS) thiết lập:
 - Việc ghi lên một file mở có thể thấy bởi những người dùng khác ngay lập tức
 - Con trỏ đến file chia sẻ cho phép nhiều người dùng truy cập đến file đồng thời.
- AFS có các ngữ cảnh sessions
 - Việc cập nhật lên file chỉ thấy được trong những sessions sau khi file đã đóng



Bảo vệ

- Người tạo/ sở hữu phải có khả năng điều khiển
- Xác định ai có thể làm gì trên file
- Các kiểu truy nhập
 - Read
 - Write
 - Execute
 - Append
 - Delete
 - List

Nhóm và quyền truy nhập



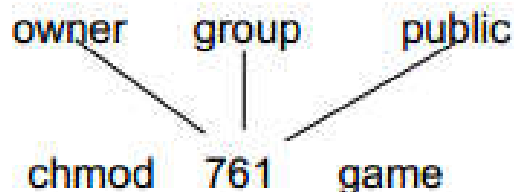
- Các mode truy nhập: read, write, execute
- Ba lớp người dùng

a) owner access rwx 7 \Rightarrow 1 1 1

b) group access rwx 6 \Rightarrow 1 1 0

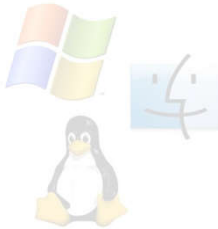
c) public access rwx 1 \Rightarrow 0 0 1

- Tạo một nhóm G (tên duy nhất), và thêm người dùng vào nhóm đó.
- Với một file (vd: game) hay thư mục con, định nghĩa một quyền truy nhập xác định

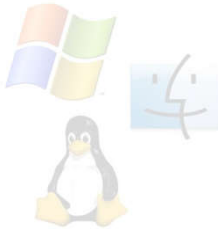


Cài đặt hệ thống file

- Cấu trúc và cài đặt hệ thống file
- Cài đặt thư mục
- Các phương pháp phân phối
- Quản lý không gian rời
- Hiệu quả, hiệu suất
- Sao lưu và Khôi phục



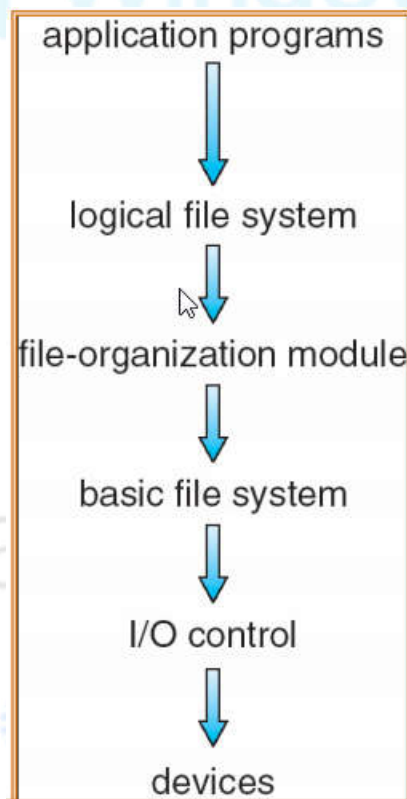
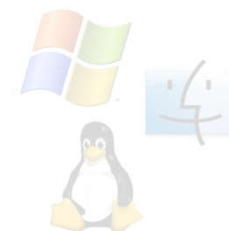
Cấu trúc và cài đặt hệ thống file



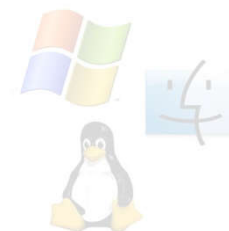
- Cấu trúc file
 - Đơn vị lưu trữ mức luận lý
 - Tập các thông tin có liên quan đến nhau
- Hệ thống file được lưu trên thiết bị lưu trữ thứ cấp (các đĩa từ)
- Hệ thống file được tổ chức thành các tầng
- Khối điều khiển file – cấu trúc lưu trữ chứa thông tin về một file



Hệ thống file phân tầng



Một khối điều khiển file điển hình



file permissions

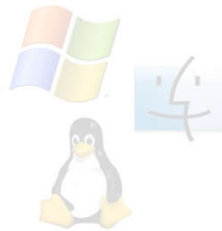
file dates (create, access, write)

file owner, group, ACL

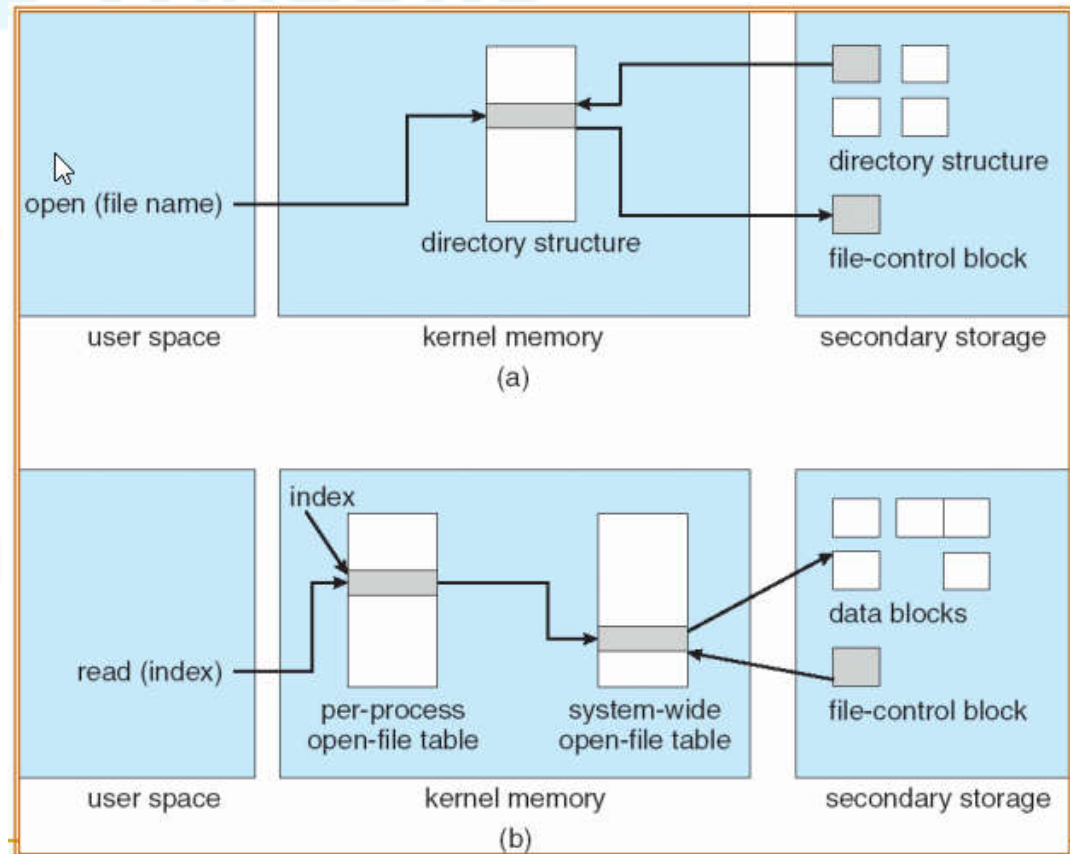
file size

file data blocks or pointers to file data blocks

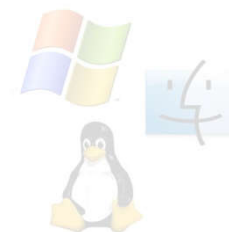
Các cấu trúc hệ thống file trong bộ nhớ



- Các cấu trúc hệ thống file cần thiết và được cung cấp bởi hầu hết các hệ điều hành.
- Hình a mô tả quá trình mở một file.
- Hình b mô tả quá trình đọc một file



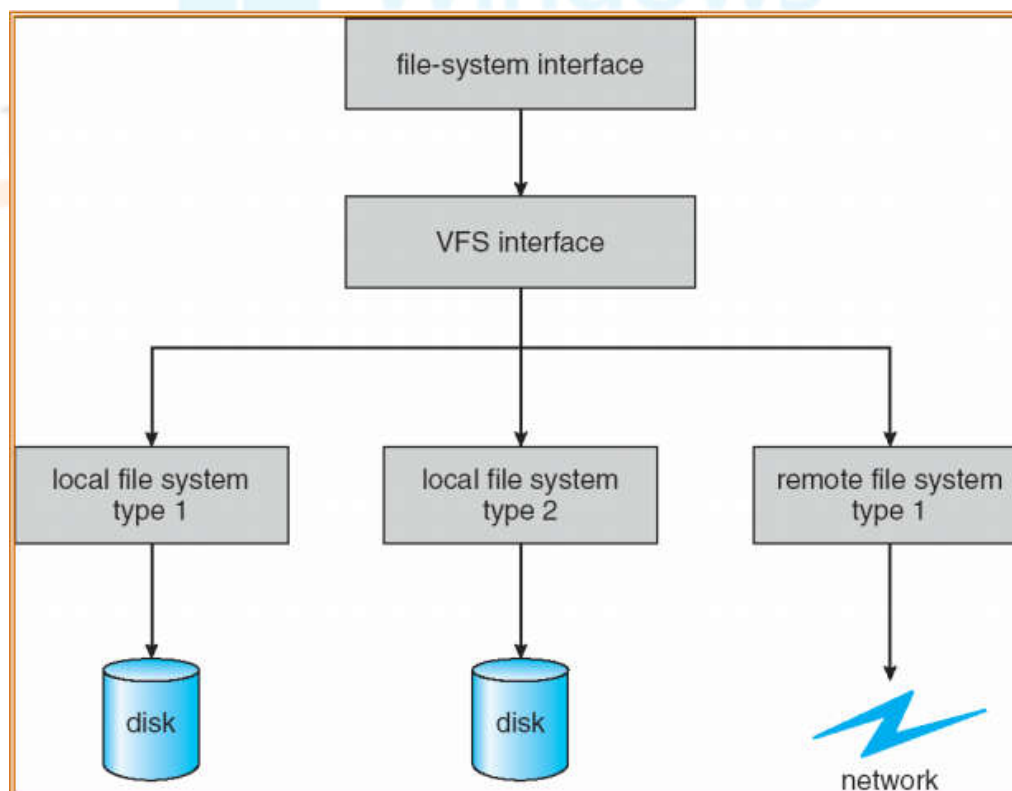
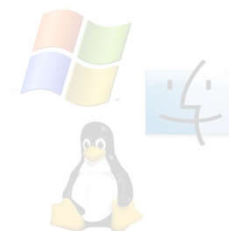
Các hệ thống file ảo (virtual file system)

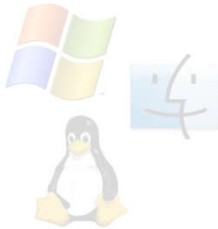


- Các hệ thống file ảo (VFS) cung cấp một cách thức hướng đối tượng để cài đặt hệ thống file
- VFS cho phép thiết lập giao diện lời gọi hệ thống (API) chung cho các loại hệ thống file khác nhau
- API là giao diện VFS thay vì giao diện của bất kì một hệ thống file cụ thể nào



Hệ thống file ảo

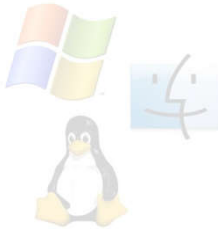




Cài đặt thư mục

- Danh sách liên kết chứa tên file và con trỏ đến các file
 - Đơn giản
 - Tốn kém thời gian
- Bảng băm.
 - Giảm thời gian tìm kiếm thư mục
 - Va chạm – tình huống xảy ra khi hai tên file được ánh xạ vào cùng một địa điểm
 - Kích cỡ cố định

Các phương pháp phân phối



- Cách thức phân phối các khối đĩa cho các file:
 - Phân phối liên tục
 - Phân phối liên kết
 - Phân phối chỉ số



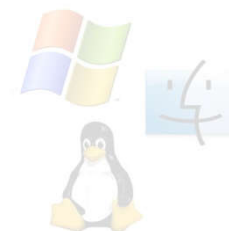
redhat.

Mac OS

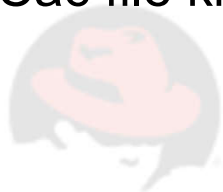
solaris



Phân phối liên tục...



- Mỗi file lưu trong một tập các khối liên tục trên đĩa
 - Đơn giản – chỉ cần điểm bắt đầu (block #) và kích cỡ (số các blocks)
 - Truy nhập ngẫu nhiên
 - Không tận dụng không gian tối ưu
 - Các file không thể thay đổi kích cỡ



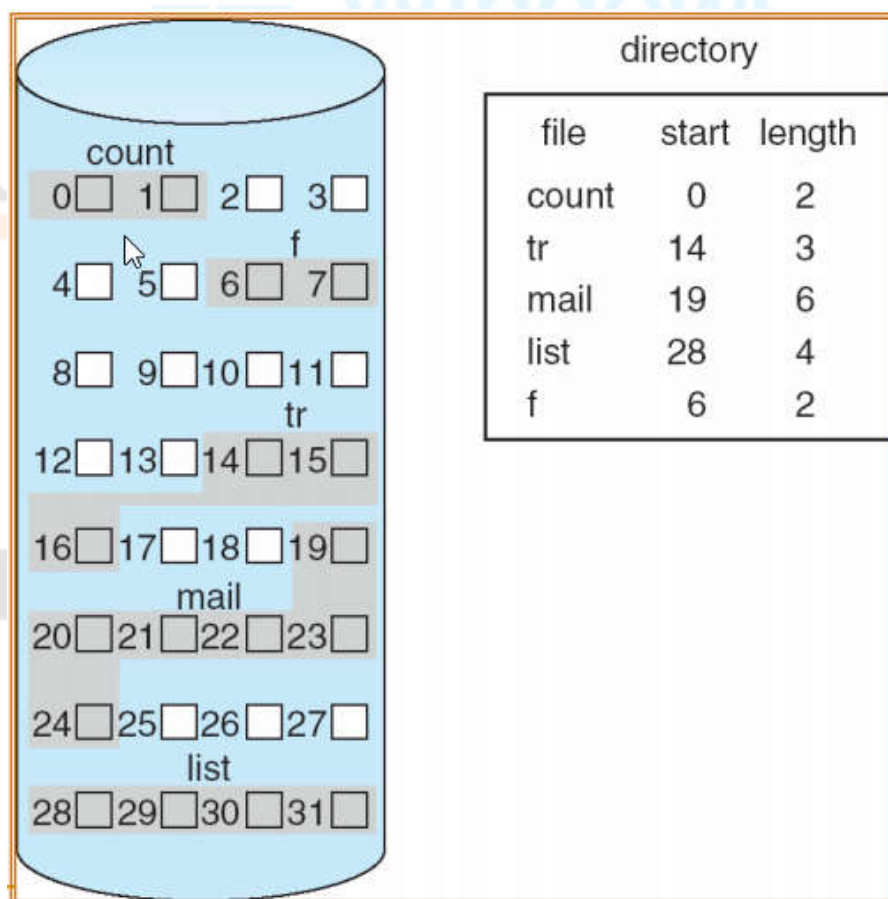
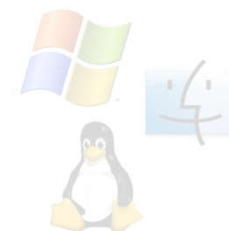
redhat.



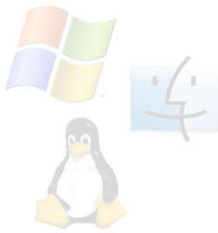
Sun Cobalt

solaris

Phân phối không gian đĩa liên tục

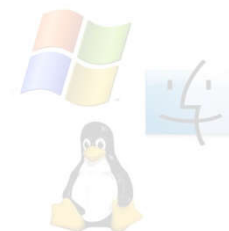


Các hệ mở rộng dựa trên phân phối liên tục



- Nhiều hệ thống file mới (ví dụ: Veritas File System) sử dụng một lược đồ phân phối liên tục có sửa đổi
- Các hệ thống file mở rộng phân phối khối đĩa trong các extents
 - Extents được phân phối cho một file
 - Một file có thể chứa một hay nhiều extent.

Phân phối liên kết...

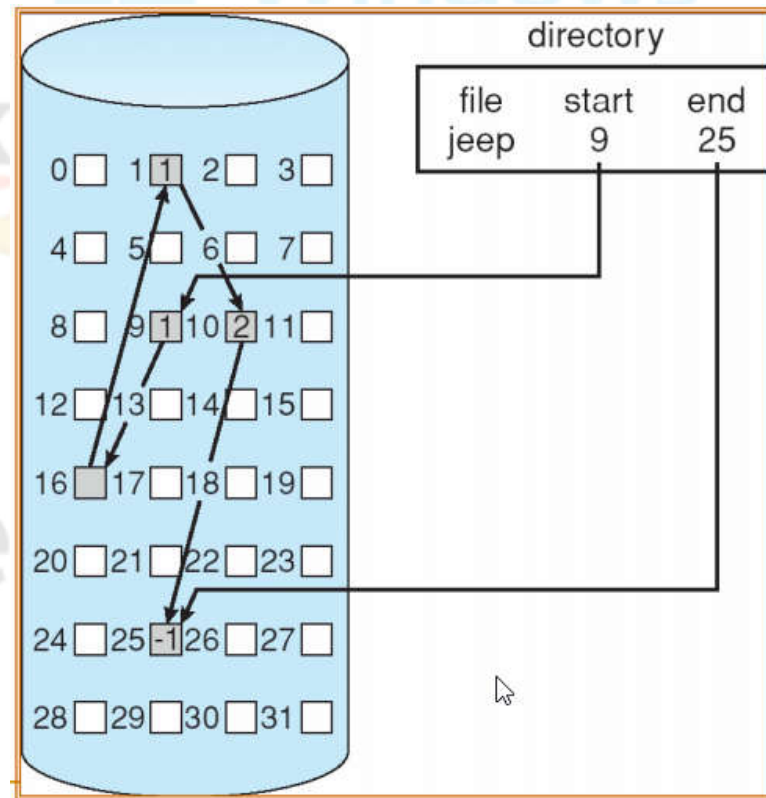
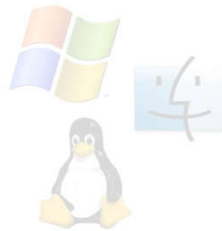


- Một file có thể là một danh sách các khối đĩa: các khối đĩa có thể nằm rải rác

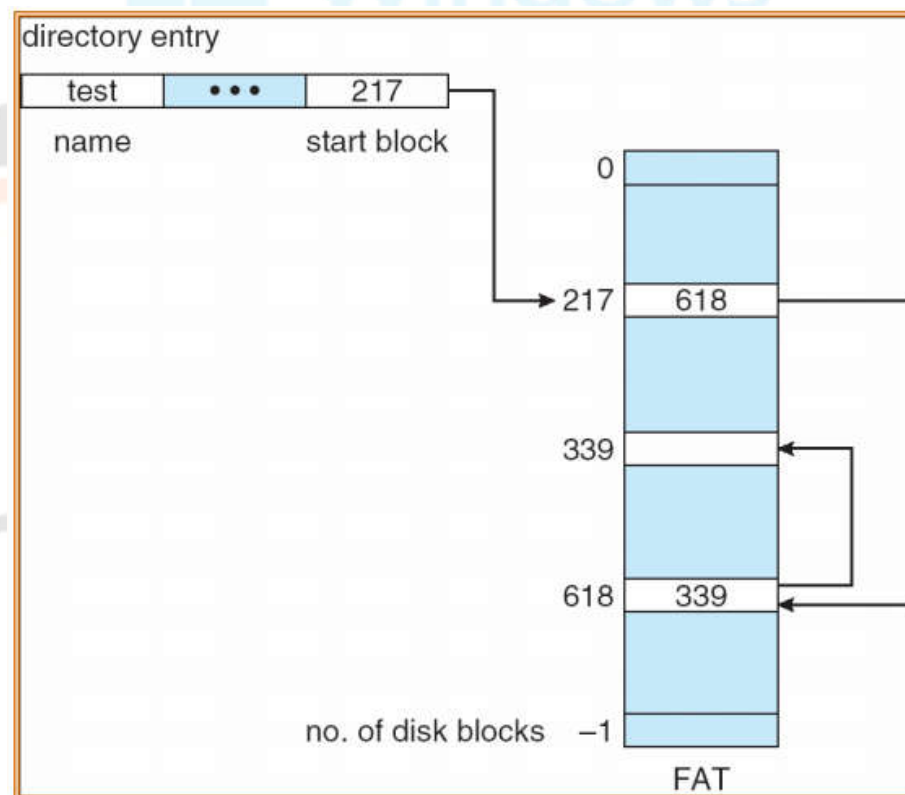
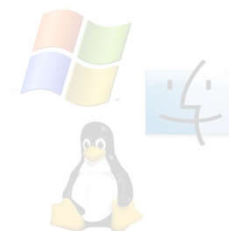
Khối =

Con trỏ

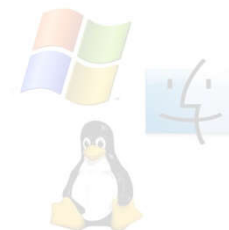
Phân phối liên kết



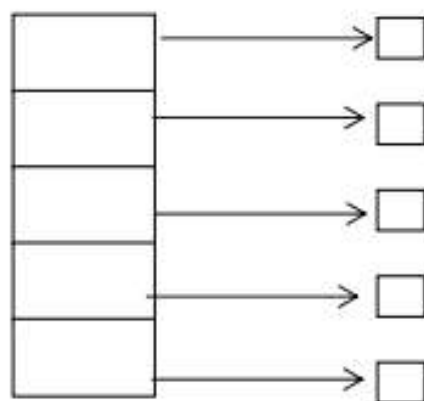
Bảng phân phối file



Phân phối chỉ số...

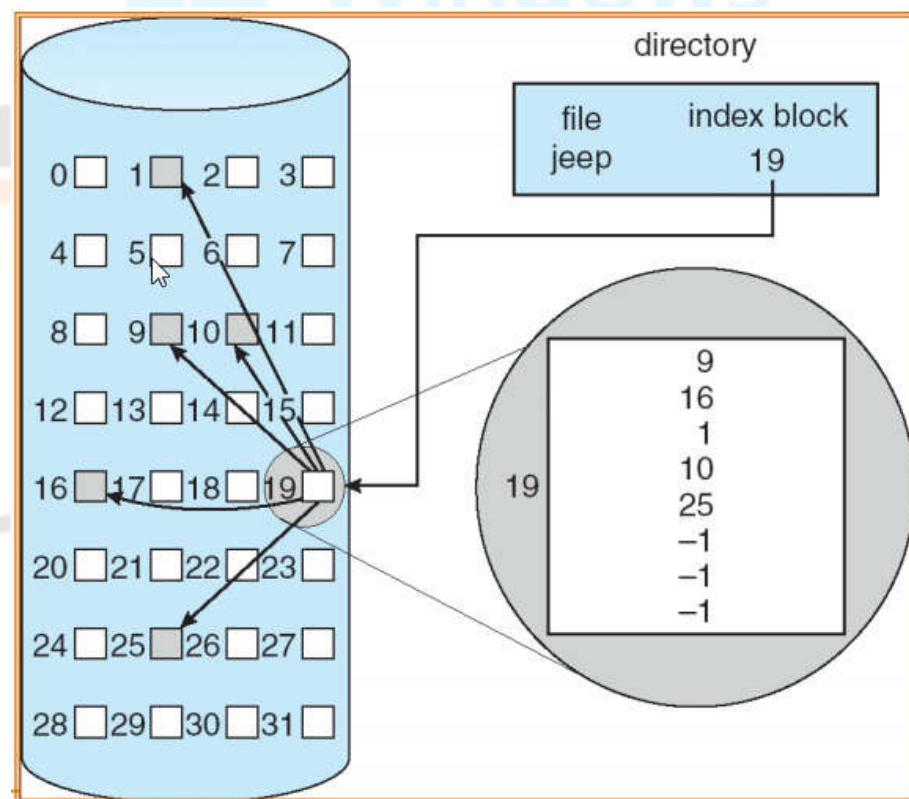
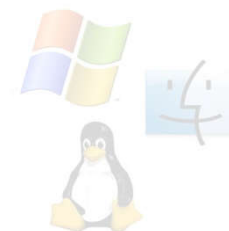


- Khối index: chứa toàn bộ con trỏ đến các khối đĩa

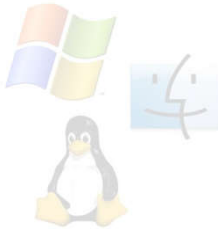


Bảng chỉ số

Ví dụ về phân phối chỉ số



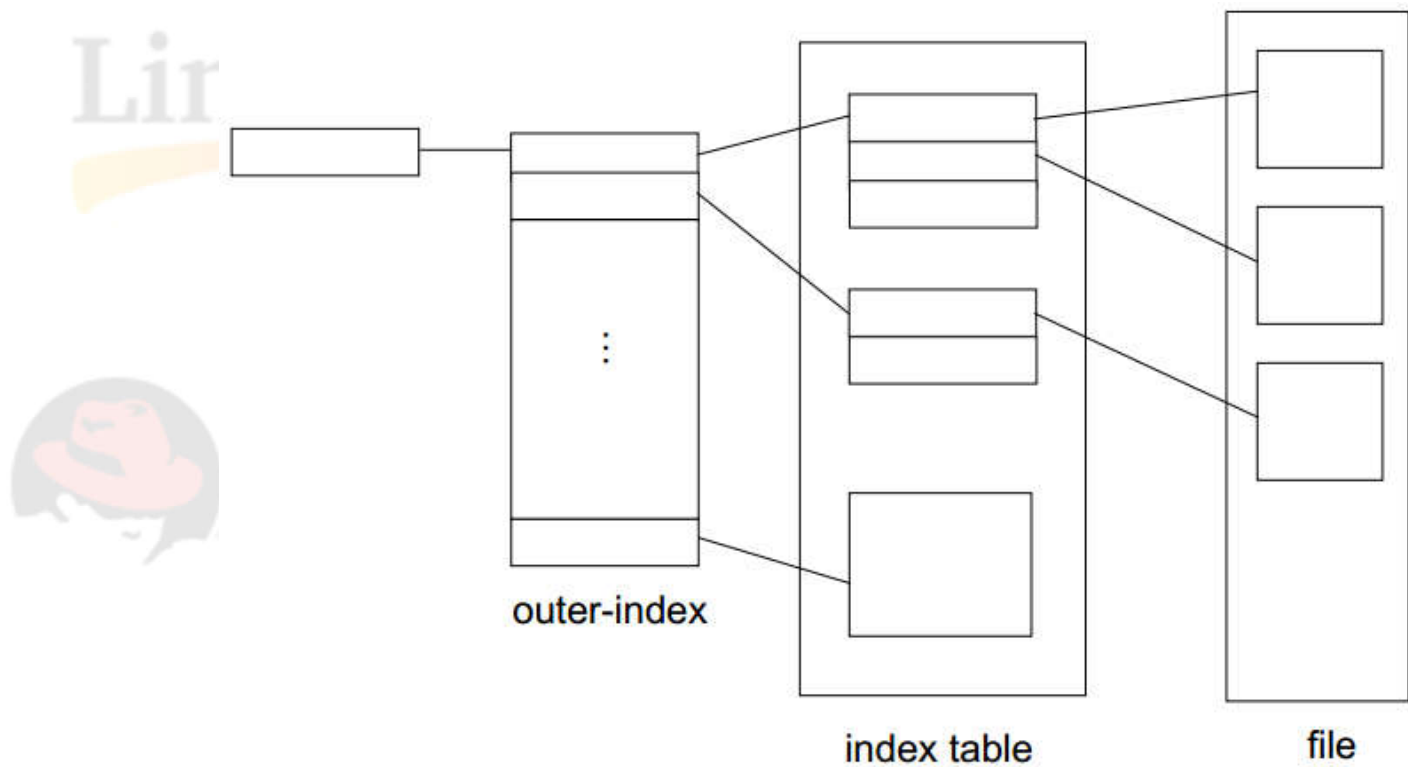
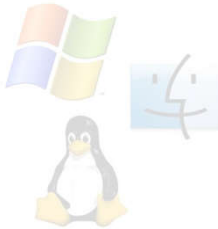
...Phân phối chỉ số...



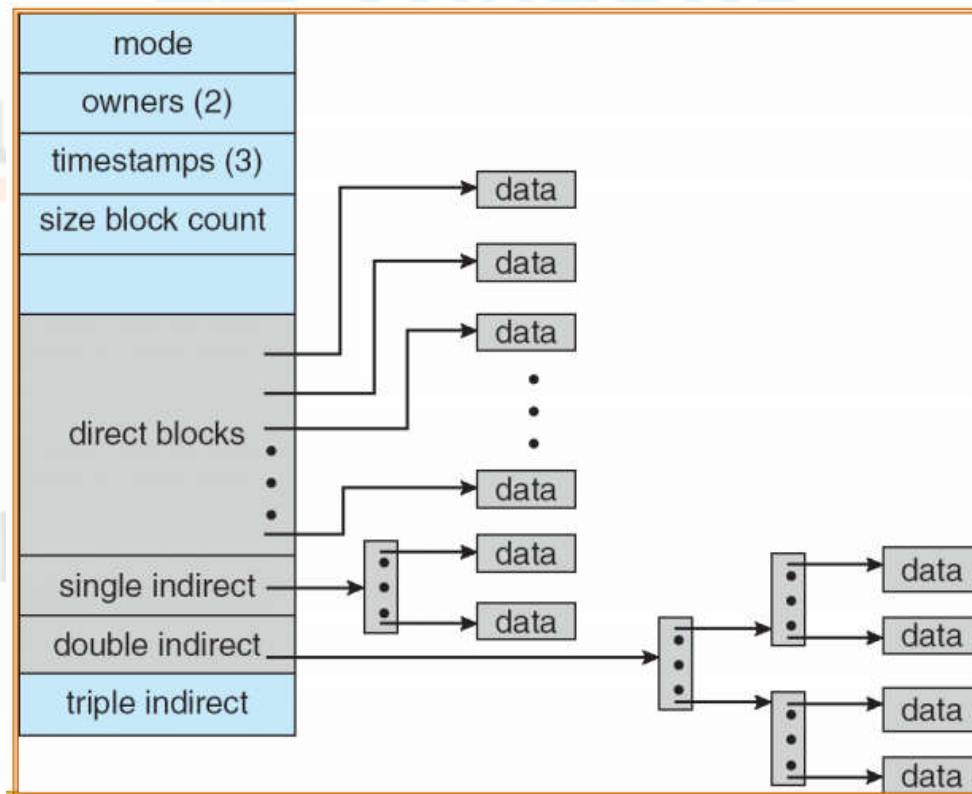
- Cần bảng chỉ số
- Truy cập ngẫu nhiên
- Phân phối động mà không sinh ra phân mảnh ngoài (chỉ tốn không gian cho bảng index)
- Ánh xạ từ không gian luận lý sang không gian vật lý trong một file kích cỡ tối đa là 512KB và kích cỡ khối là 1024B chúng ta chỉ cần một khối cho bảng index

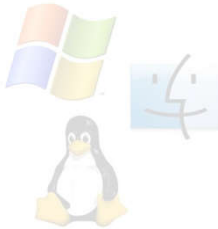


...Phân phối file chỉ số - Ánh xạ



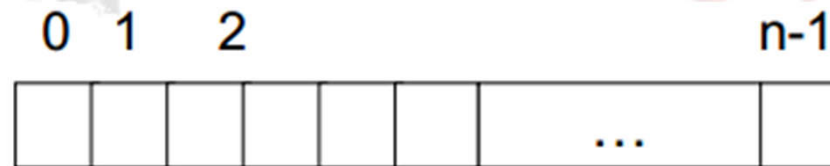
Lược đồ kết hợp: UNIX (4K bytes mỗi khối)





Quản lý không gian rỗi

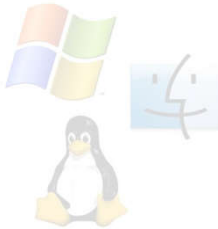
- Vector bit (n khối)



$$bit[i] = \begin{cases} 0 \Rightarrow block[i] \text{ free} \\ 1 \Rightarrow block[i] \text{ occupied} \end{cases}$$

- Tính toán số khối:
(số lượng bit mỗi từ) * (số lượng từ nhận giá trị 0) +
(gia số bit 1 đầu tiên)

...Quản lý không gian rồi...



- Ánh xạ bit cần thêm không gian

- Ví dụ:

Kích cỡ khối = 2^{12} bytes

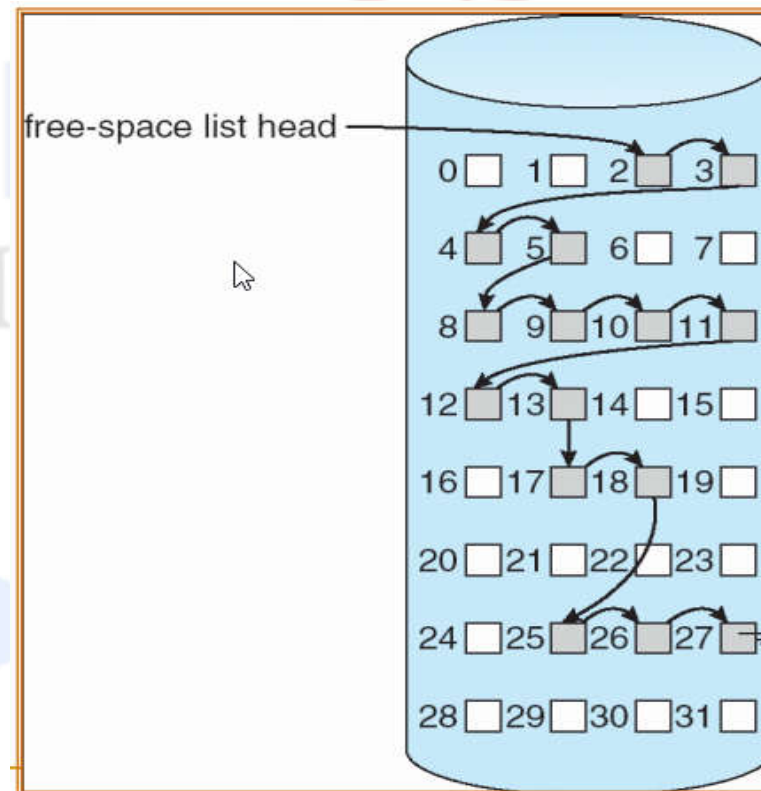
Kích cỡ đĩa = 2^{30} bytes (1 gigabyte)

$$N = \frac{2^{30}}{2^{12}} = 2^{18} \text{ bits (or 32K bytes)}$$

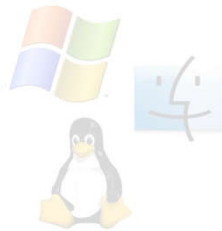
- Dễ dàng truy nhập đến file liên tục
- Danh sách liên kết (danh sách liên kết các khối rồi)
 - Không có được không gian liên tục
 - Không lãng phí không gian

...Quản lý không gian rồi...

- Dùng danh sách liên kết (danh sách liên kết các khối rỗi)
 - Khó có được không gian liên tục
 - Không lãng phí không gian



...Quản lý không gian rồi



- Cần phải bảo vệ:
 - Con trỏ đến danh sách rồi
 - Ánh xạ bit
 - Phải được giữ trên đĩa
 - Bản sao trong đĩa và trong bộ nhớ có thể khác nhau
 - Không cho phép khối[i] ở trong trạng thái mà $\text{bit}[i] = 1$ trong bộ nhớ và $\text{bit}[i] = 0$ trên đĩa
- Giải pháp:
 - Thiết lập $\text{bit}[i] = 1$ trong đĩa
 - Phân phối khối[i]
 - Thiết lập $\text{bit}[i] = 1$ trong bộ nhớ

Grouping và counting (1)

□ Phương pháp *grouping*

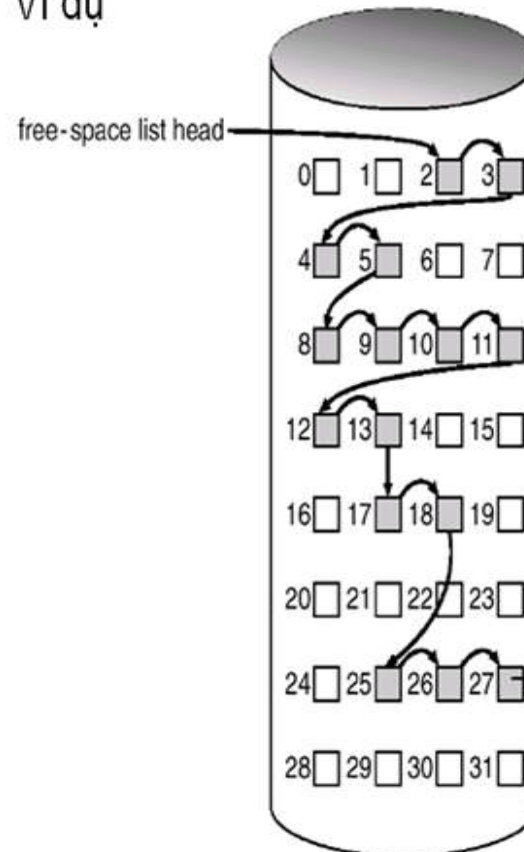
- Địa chỉ của n khối trống được lưu trong khối nhớ đầu tiên.
- Khối nhớ thứ n chứa địa chỉ của m khối nhớ trống kế tiếp.

□ Phương pháp *counting*

- Tổ chức bảng chỉ số, trong đó mỗi phần tử của bảng chứa địa chỉ trên đĩa của khối trống đầu tiên trong nhóm khối trống liên tục kèm theo một số đếm (cho biết số khối trống trong nhóm).
- Có thể cấp phát hoặc thu hồi đồng thời nhiều khối nhớ liên tục

Grouping và counting (2)

Ví dụ



Grouping

Block 2 → 3, 4, 5

Block 5 → 8, 9, 10

Block 10 → 11, 12, 13

Block 13 → 17, 18, 25

Block 25 → 26, 27

Counting

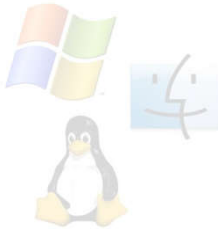
2 4

8 6

17 2

25 3

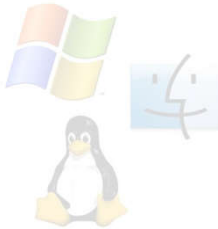
Hiệu năng



- Hiệu quả phụ thuộc vào:
 - Các thuật toán phân phối đĩa và thư mục
 - Các kiểu dữ liệu được giữ trong đầu vào thư mục chứa file
- Năng suất
 - Cache đĩa – lưu lại một phần đĩa thường xuyên được truy nhập
 - Giải phóng sau- đọc trước – kĩ thuật tối ưu truy nhập tuần tự
 - Tăng năng suất làm việc cho PC



Cache trang



- Một cache trang lưu lại các trang thay vì các khối đĩa sử dụng bởi các kĩ thuật bộ nhớ
- Ảnh xạ bộ nhớ I/O sử dụng cache trang
- Các thao tác vào ra với hệ thống file sử dụng page(disk) cache



redhat.

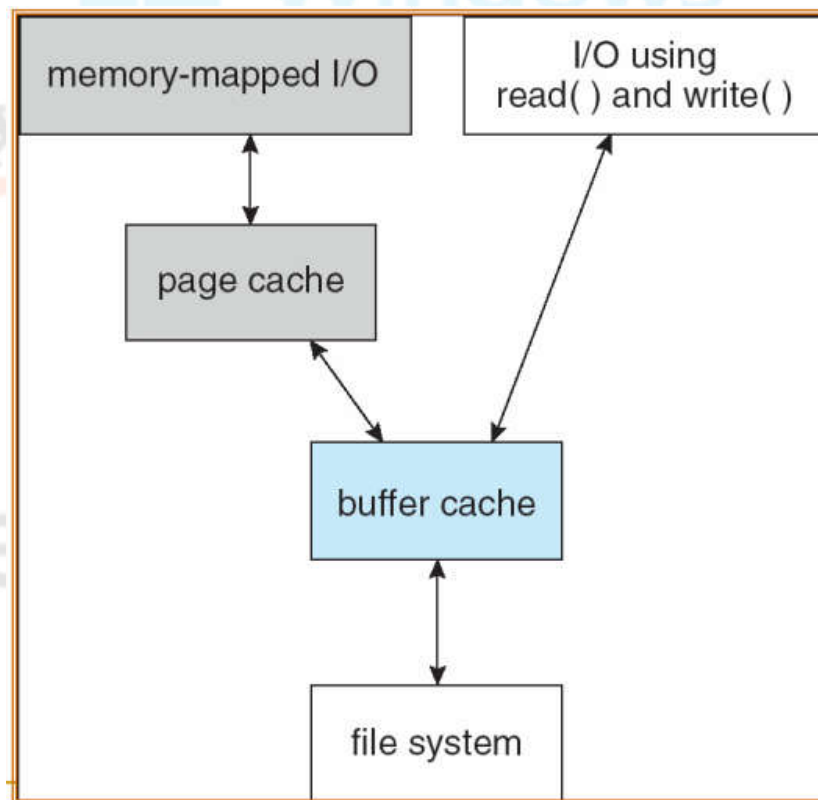
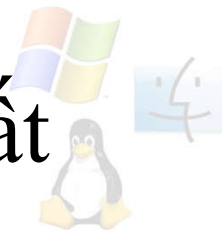
Mac OS

solaris

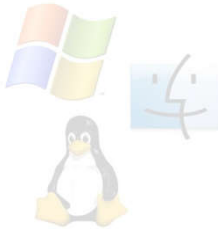


Sun Cobalt

I/O mà không có một tổ chức cache hợp nhất



Sử dụng cache bộ đệm hợp nhất



- Một cache bộ đệm hợp nhất: sử dụng không gian cache page để
 - cache cả các trang ảnh xạ bộ nhớ
 - vào/ra các hệ thống file thông thường



redhat.

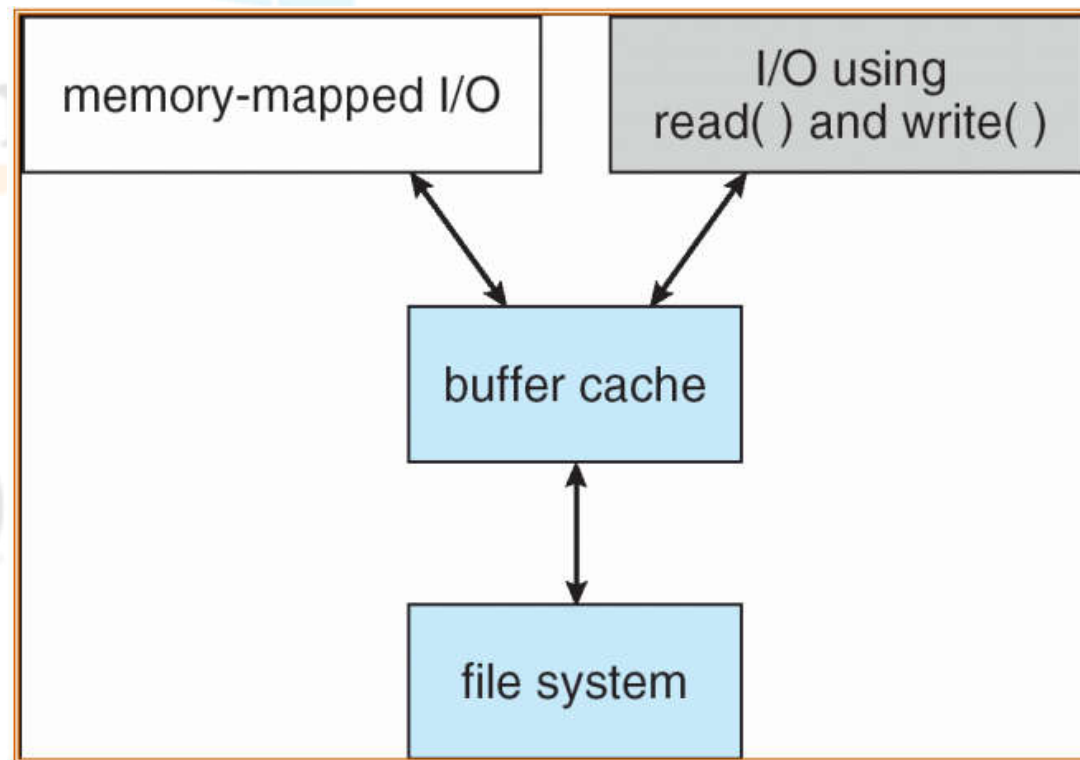
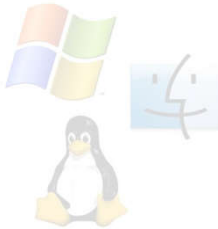
Mac OS

solaris

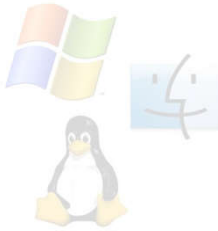


Sun Cobalt

I/O sử dụng cache bộ đệm hợp nhất

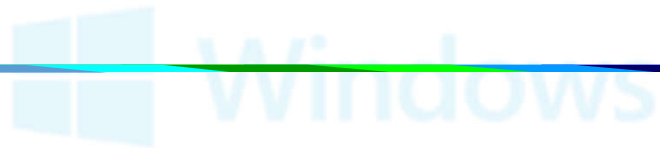
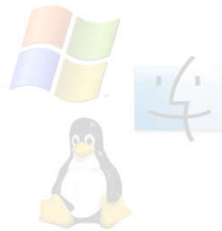


Sao lưu và khôi phục (Backup and Restore)



- Kiểm tra tính nhất quán – so sánh dữ liệu trong cấu trúc thư mục và so sánh với khối đĩa, cố gắng giải quyết tính không nhất quán
- Sử dụng các chương trình hệ thống để back up dữ liệu từ đĩa sang các thiết bị lưu trữ khác (floppy disk, magnetic tape, other magnetic disk, optical)
- Khôi phục file hay thư mục bị mất bằng cách khôi phục lại bản sao lưu





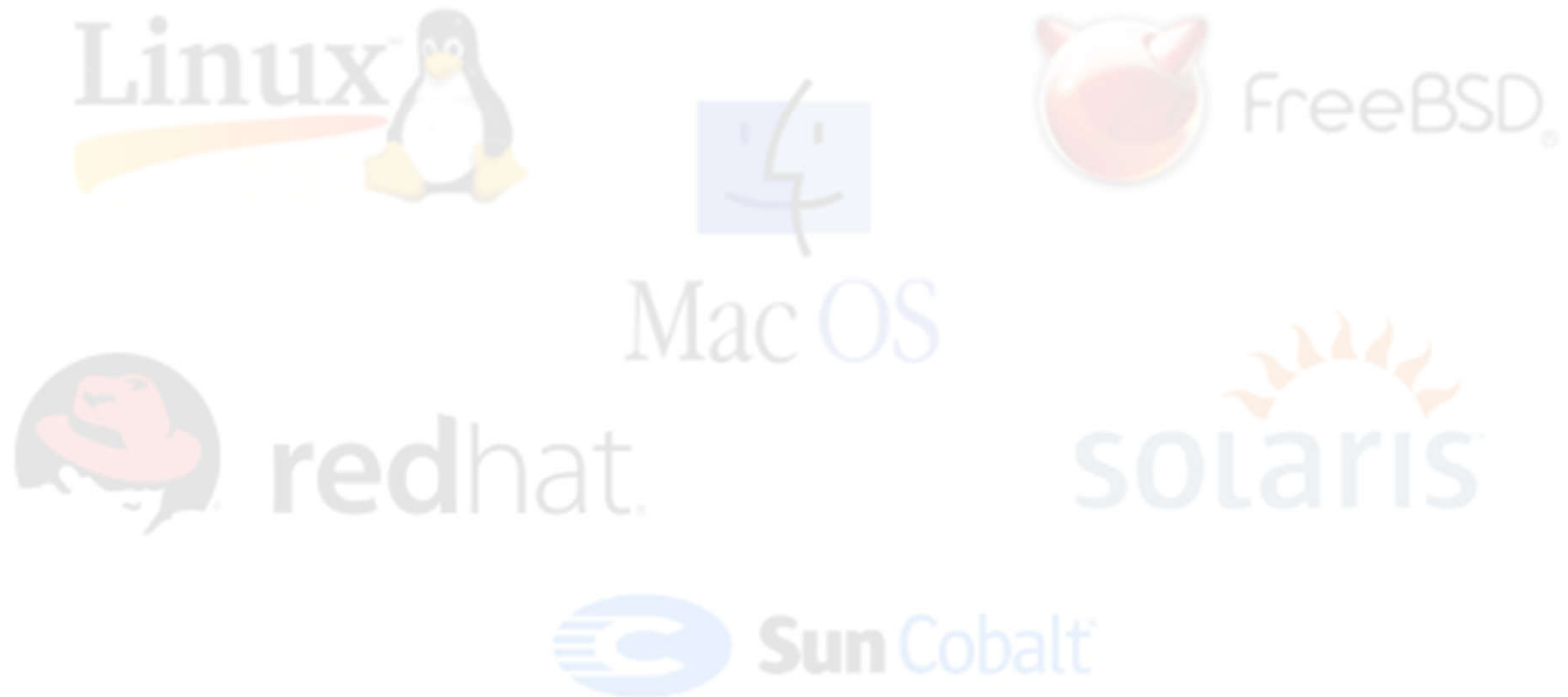
Question?

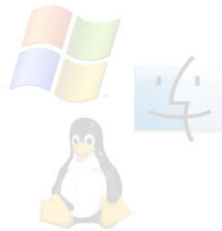


redhat.



Câu hỏi ôn tập...





Windows

Lin



FreeBSD



solaris

dreamstime.com