ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA

NGUYÊN LÝ HỆ ĐIỀU HÀNH

Khoa Công nghệ thông tin
ThS. Nguyễn Thị Lệ Quyên

Nội dung môn học

- Giới thiệu
- Tiến trình và luồng
- ***Thi giữa kỳ***
- Quản lý bộ nhớ
- Vào/ Ra
- **Hệ thống file**
- Thực hành

HỆ THỐNG FILE

Nội dung

- **File-system Interface**
- File-system Implementation
- File-System Internals

File-system Interface

- Khái niệm
- Phương thức truy cập
- Cấu trúc thư mục
- Cơ chế bảo vệ
- Memory-mapped Files (file ánh xạ bộ nhớ)
- Summary

File-system Interface

- Cần nắm:
  - Giải thích chức năng của hệ thống file
  - Mô tả các interface của hệ thống file
  - Thảo luận về sự đánh đổi trong thiết kế hệ thống file, bao gồm các phương pháp truy cập, chia sẻ file, khóa file và cấu trúc thư mục
  - Khám phá cơ chế bảo vệ hệ thống file

File-system Interface

- **Khái niệm**
- Phương thức truy cập
- Cấu trúc thư mục
- Cơ chế bảo vệ
- Memory-mapped Files (file ánh xạ bộ nhớ)
- Summary

Khái niệm file

- OS trừu tượng hóa các thuộc tính vậy lý của thiết bị lưu trữ để xác định đơn vị logic, gọi là file. Các file được OS ánh xạ vào các thiết bị vật lý.
- File là tập hợp các thông tin liên quan được đặt tên và ghi vào bộ nhớ thứ cấp.
- Thông thường, file thường đại diện cho chương trình (cả dạng source và object) và dữ liệu.
- Nói chung file là 1 chuỗi các bit, byte, line hoặc bản ghi (record), ý nghĩa của nó được xác định bởi người tạo và người dùng file → khai niệm về file rất chung chung

Khái niệm file

- Bởi vì file là phương pháp mà user và application sử dụng để lưu trữ và truy xuất dữ liệu, đồng thời vì chúng có mục đích chung nên việc sử dụng chúng đã vượt ra người giới hạn ban đầu
  - VD UNIX, Linux và 1 số OS khác cung cấp hệ thống file Proc sử dụng file-system interface để cung cấp quyền truy cập vào thông tin hệ thống (chẳng hạn chi tiết tiến trình)
- Thông tin trong file được xá định bởi người tạo ra nó. Nhiều loại thông tin khác nhau có thể được lưu trữ trong 1 file – các chương trình nguồn hoặc các chương trình thực thi, dữ liệu số hoặc văn bản, ảnh, nhạc, video, ...

Thuộc tính file

- Name – Tên file là thông tin duy nhất được lưu trữ ở dạng con người có thể đọc được
- Identifie – Thẻ duy nhất, thường là 1 số, xác định file trong hệ thống file; là tên file mà con người không thể đọc được
- Type – Thông tin cần thiết cho các hệ thống hỗ trợ các loại file khác nhau
- Location – Thông tin là 1 con trỏ tới một thiết bị và tới vị trị file trên thiết bị đó
- Size – kích thước hiện tại của file (tính bằng bytes, words hoặc blocks) và có thể kích thước tối đa cho phép được bao gồm trong thuộc tính này
- Protection – Thông tin kiểm soát truy cập xác định ai có thể đọc, ghi, thực thi,...
- Timestamps và user identification – Thông tin này có thể được lưu trữ về thời gian tạo, sửa đổi lần cuối và sử dụng lần cuối. Dữ liệu hữu ích cho việc bảo vệ, bảo mật, giám sát việc sử dụng

Hoạt động liên quan đến file

- Tạo file
- Mở file
- Ghi file
- Đọc file
- Tìm file
- Xóa file
- Cắt bớt file
- Ngoài ra còn có thêm thông tin vào cuối file, đổi tên file, copy file, ...

File Types

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 compressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

Figure 13.3 Common file types.

File-system Interface

- Khái niệm
- **Phương thức truy cập**
- Cấu trúc thư mục
- Cơ chế bảo vệ
- Memory-mapped Files (file ánh xạ bộ nhớ)
- Summary

Phương thức truy cập

- Truy cập tuần tự
- Truy cập trực tiếp
- Phương thức truy cập khác

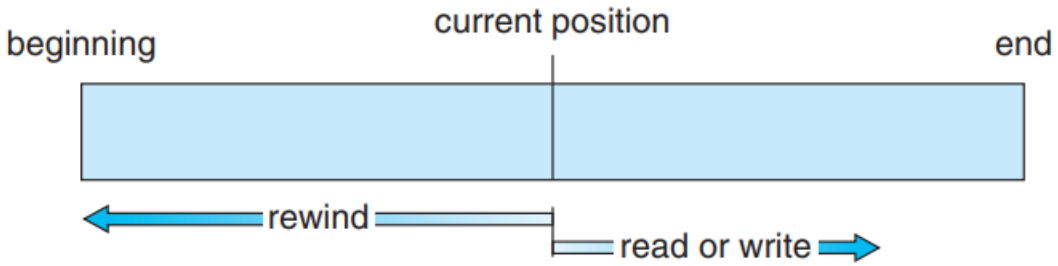


Figure 13.4 Sequential-access file.

Truy cập tuần tự

- Phương thức truy cập đơn giản nhất.
- Thông tin trong file được xử lý theo thứ tự, từng record một.
- Phương thức truy cập này được sử dụng phổ biến
  - VD editor và compiler thường truy cập file theo kiểu này
- Read và write chiếm phần lớn thao tác trên 1 file.
  - Thao tác đọc – read_next() – đọc phần tiếp theo của file và tự động chuyển con trỏ file để theo dõi vị trí I/O.
  - Thao tác ghi – write_next() – ghi vào cuối file
- Hoạt động tốt trên các thiết bị truy cập tuần tự và thiết bị truy cập ngẫu nhiên

Truy cập trực tiếp

- Còn gọi là truy cập tương đối (relative access)
- Phương pháp dựa trên mô hình đĩa của 1 file, vì đĩa cho phép truy cập ngẫu nhiên vào bất kỳ khối file nào
- Các file truy cập trực tiếp được sử dụng để truy cập vào 1 lượng lớn thông tin
  - CDSL thuộc loại này. Khi có 1 truy vấn liên quan đến chủ đề cụ thể, tính toán khối nào chứa câu trả lời và sau đó đọc trực tiếp khối đó để lấy thông tin mong muốn
- Sử dụng thao tác read(n) và write(n) với n là block number
- Một cách tiếp cận khác là giữ lại read_next() và write_next() và thêm position_file(n) với n là block number. Khi thực hiện thao tác đọc, đầu tiên sẽ thực hiện position_file(n), sau đó là read_next().

Truy cập trực tiếp

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

Figure 13.5 Simulation of sequential access on a direct-access file.

Phương thức truy cập khác

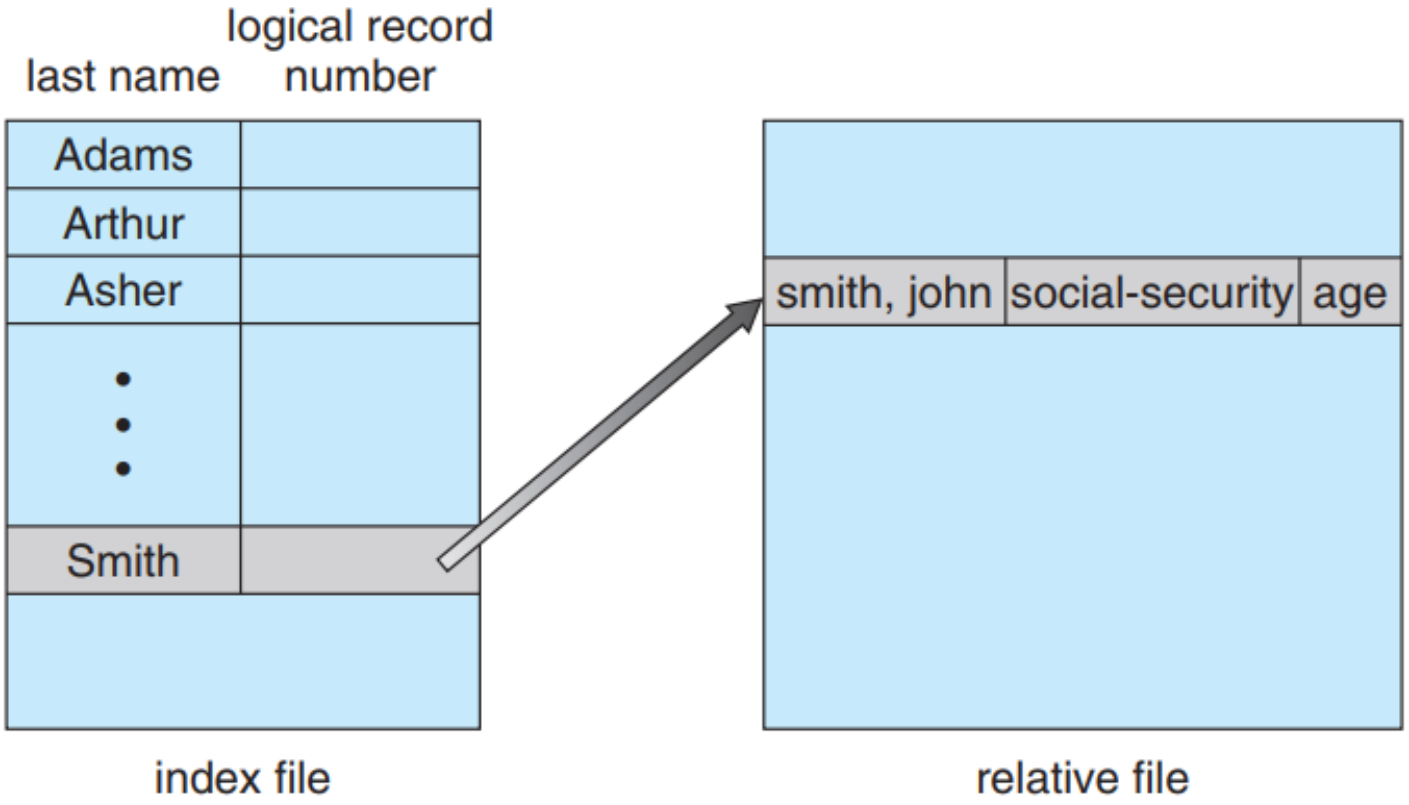


Figure 13.6 Example of index and relative files.

File-system Interface

- Khái niệm
- Phương thức truy cập
- **Cấu trúc thư mục**
- Cơ chế bảo vệ
- Memory-mapped Files (file ánh xạ bộ nhớ)
- Summary

Cấu trúc thư mục

- Hoạt động liên quan đến thư mục:
  - Tìm kiếm 1 file
  - Tạo 1 file
  - Xóa 1 file
  - Liệt kê 1 thư mục
  - Đổi tên 1 file
  - Duyệt qua hệ thống file

Cấu trúc thư mục

- Single-level directory
- Two-level directory
- Tree-Structured directories
- Acyclic-Graph directories
- General graph directory

Single-level directory

- Cấu trúc thư mục đơn giản nhất
- Tất cả các file nằm chung trong 1 thư mục, dễ hỗ trợ, dễ hiểu

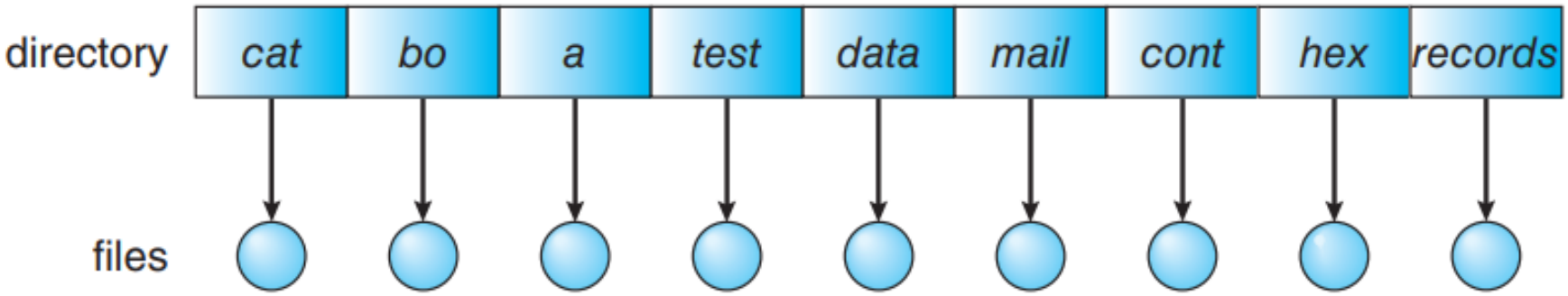


Figure 13.7 Single-level directory.

Two-level directory

- Mỗi user có 1 UFD (user file directory)
- Các UFD có chung cấu trúc

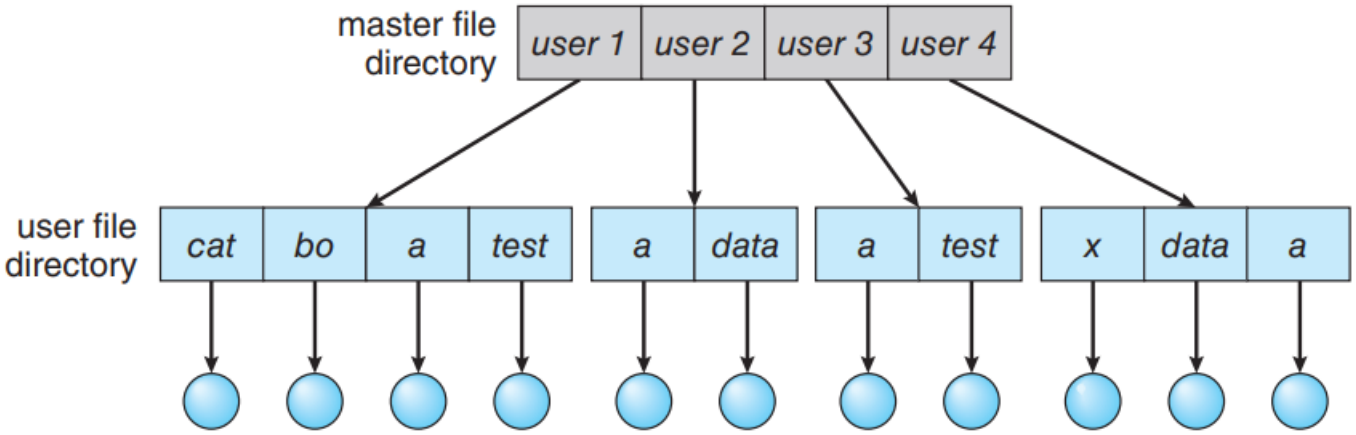


Figure 13.8 Two-level directory structure.

Two-level directory

- Ưu điểm:
  - Giải quyết được vấn đề xung đột tên
  - Tách biệt không gian cho từng người dùng hoạt động độc lập
- Nhược điểm:
  - Khi người dùng muốn hợp tác và truy cập vào file của người dùng khác sẽ không được

Tree-Structured Directories

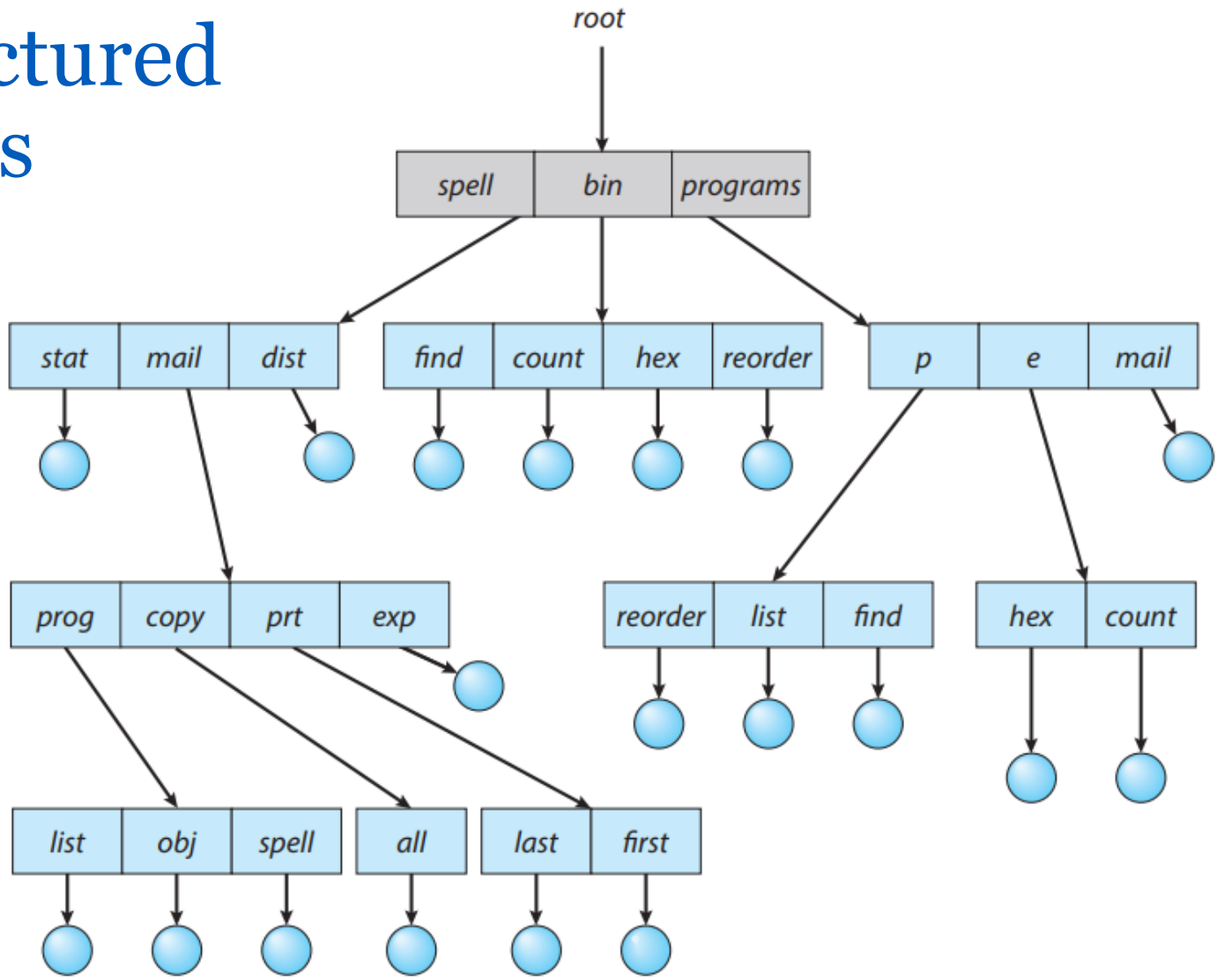


Figure 13.9 Tree-structured directory structure.

Tree-Structured Directories

- Tên đường dẫn thường có 2 loại:
  - Đường dẫn tuyệt đối
    - Trong UNIX và Linux bắt đầu tại root (thường là "/")
  - Đường dẫn tương đối
    - Định nghĩa từ thư mục hiện hành
- Chính sách của tree-structured directory xử lý thế nào khi xóa 1 thư mục
  - Nếu thư mục rỗng → xóa bình thường
  - Nếu thư mục không rỗng (chứa các file và các thư mục con), có 2 cách
    - Bắt buộc user xóa toàn bộ file và thư mục con rồi mới cho xóa
    - Tự động xóa toàn bộ file và thư mục con

Acyclic-Graph Directories

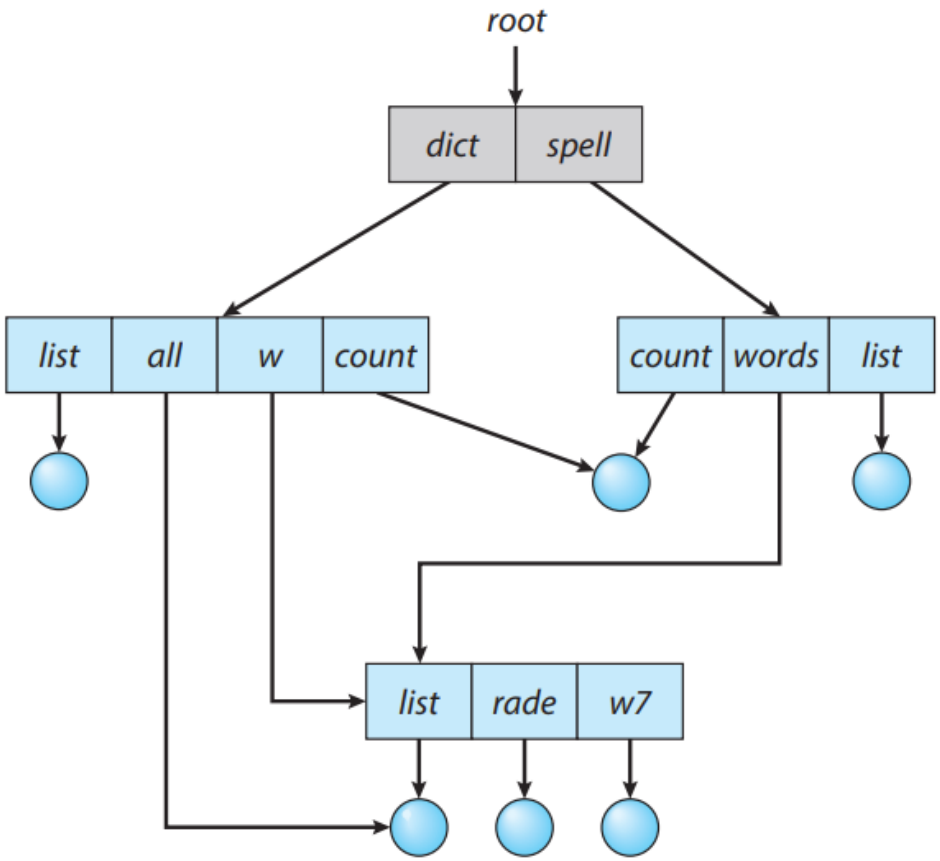


Figure 13.10 Acyclic-graph directory structure.

General Graph Directory

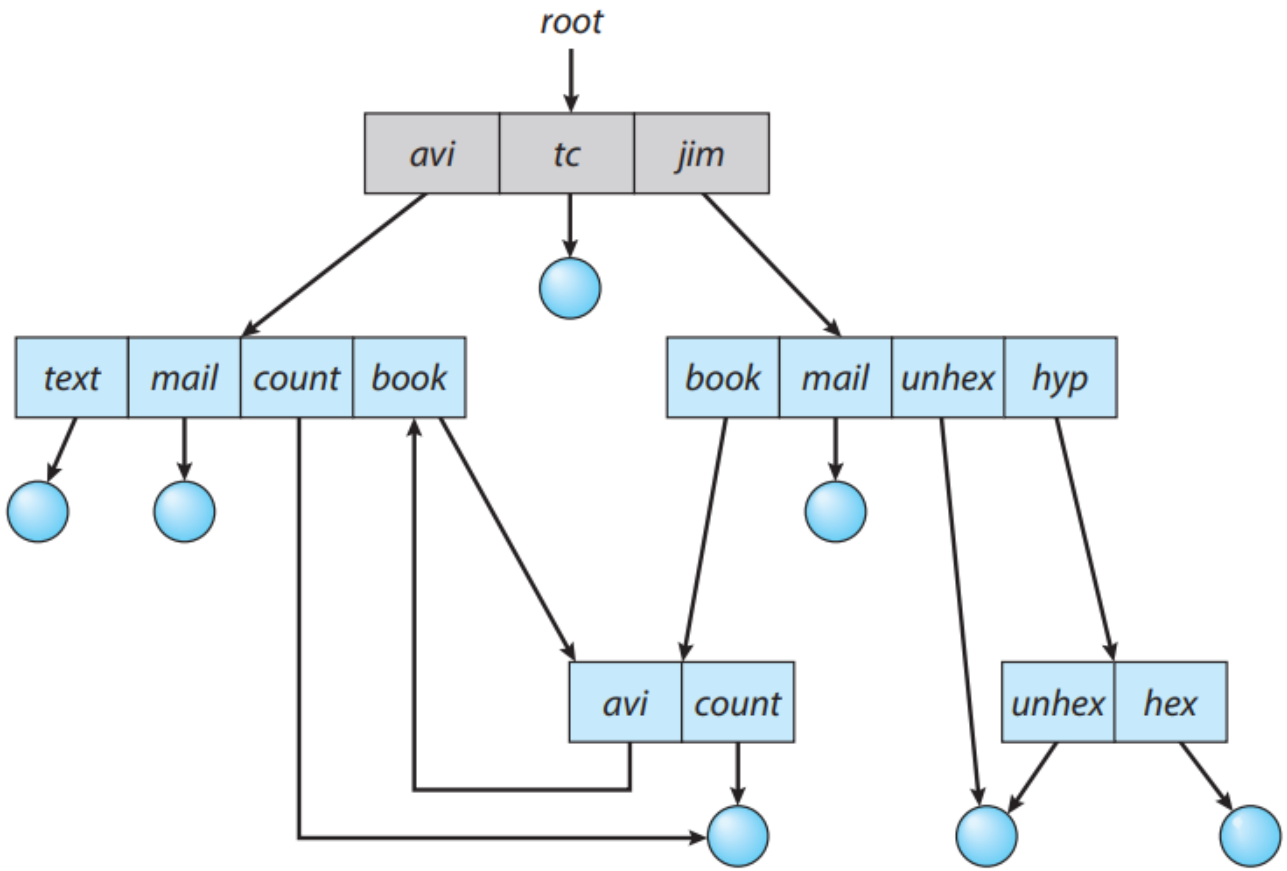


Figure 13.11 General graph directory.

File-system Interface

- Khái niệm
- Phương thức truy cập
- Cấu trúc thư mục
- **Cơ chế bảo vệ**
- Memory-mapped Files (file ánh xạ bộ nhớ)
- Summary

Cơ chế bảo vệ

- Loại truy cập
- Kiểm soát truy cập
- Phương pháp bảo vệ khác

Loại truy cập

- Read
- Write
- Execute
- Append
- Delete
- List
- Attribute change

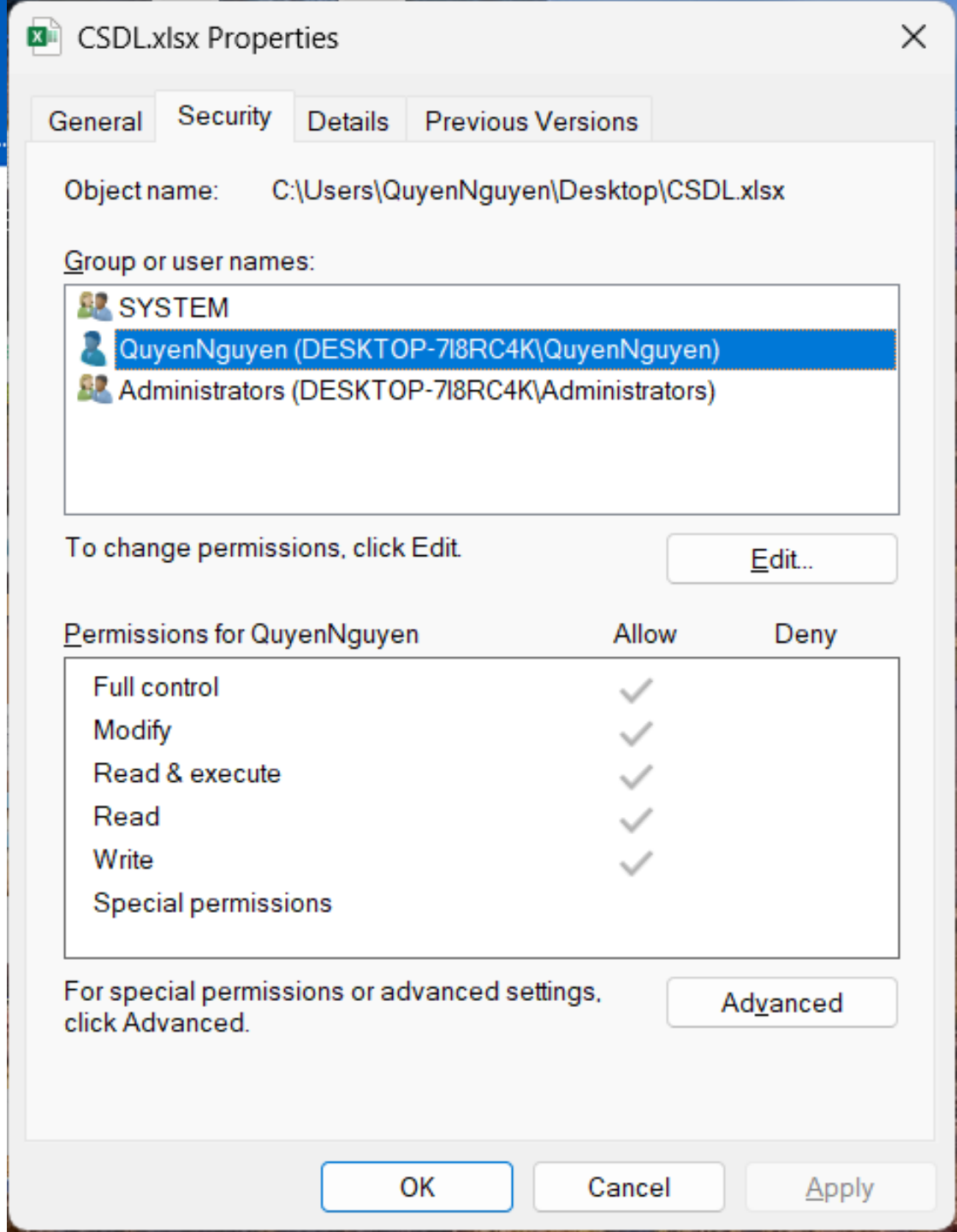
Kiểm soát truy cập

- Phân loại người dùng liên quan đến mỗi file:
  - Owner – người tạo ra file
  - Group – Tập user cùng chia sẻ file và cần quyền truy cập tương tự nhau trong một group
  - Other – Tất cả user trên hệ thống

```
-rw-rw-r--   1 pbg   staff     31200  Sep 3 08:30    intro.ps
drwx------   5 pbg   staff       512  Jul 8 09.33    private/
drwxrwxr-x   2 pbg   staff       512  Jul 8 09:35    doc/
drwxrwx---   2 jwg   student     512  Aug 3 14:13    student-proj/
-rw-r--r--   1 pbg   staff      9423  Feb 24 2017    program.c
-rwxr-xr-x   1 pbg   staff     20471  Feb 24 2017    program
drwx--x--x   4 tag   faculty     512  Jul 31 10:31   lib/
drwx------   3 pbg   staff      1024  Aug 29 06:52   mail/
drwxrwxrwx   3 pbg   staff       512  Jul 8 09:35    test/
```

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA

Kiểm soát truy cập



Phương pháp bảo vệ khác

- Sử dụng password cho mỗi file. Muốn truy cập thì cần phải nhập password
- Nhược điểm:
  - Cần nhớ nhiều pasword
  - Nếu chỉ sử dụng 1 password thì khi password bị lộ thì những file khác mất sự bảo vệ

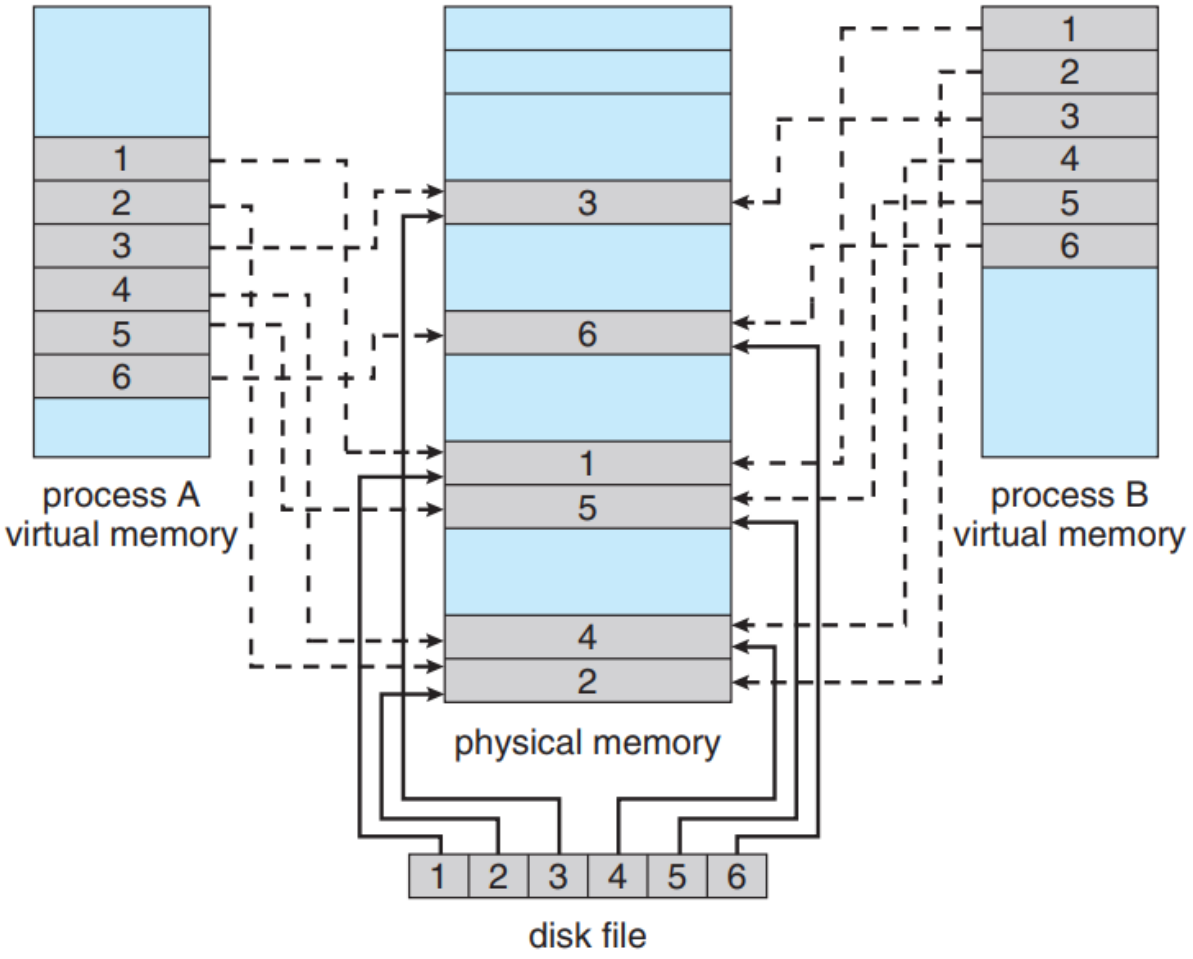
File-system Interface

- Khái niệm
- Phương thức truy cập
- Cấu trúc thư mục
- Cơ chế bảo vệ
- **Memory-mapped Files (file ánh xạ bộ nhớ)**
- Summary

Memory-Mapped Files

- Ánh xạ bộ nhớ 1 file được thực hiện bằng cách ánh xạ 1 block trên đĩa tới 1 page hoặc nhiều page trong bộ nhớ.
- Việc truy cập file ban đầu được tiến hành thông qua demand paging, dẫn tới page fault. Việc truy cập file được xử lý như truy cập bộ nhớ thông thường.
- Thao tác với file thông qua bộ nhớ sẽ tăng tốc độ truy cập và sử dụng file vì không phải phát sinh chi phí system call read() và write()

Memory-Mapped Files



Memory-Mapped Files

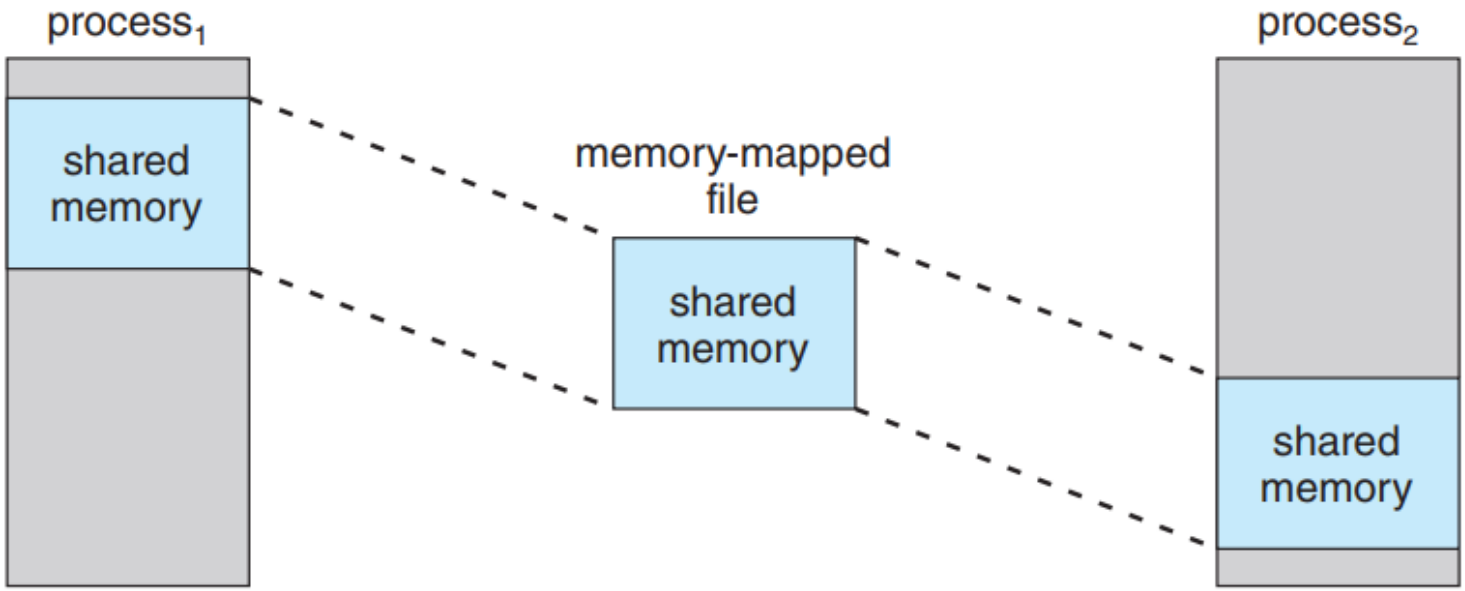


Figure 13.14 Shared memory using memory-mapped I/O.

File-system Interface

- Khái niệm
- Phương thức truy cập
- Cấu trúc thư mục
- Cơ chế bảo vệ
- Memory-mapped Files (file ánh xạ bộ nhớ)
- **Summary**

Summary

- File là một kiểu dữ liệu trừu tượng được xác định và triển khai OS. Đó là một chuỗi các bản ghi logic. Bản ghi logic có thể là 1 byte, 1 line (có độ dài cố định hoặc thay đổi) hoặc một mục dữ liệu phức tạp hơn. OS có thể hỗ trợ cụ thể nhiều loại bản ghi khác nhau hoặc có thể để lại sự hỗ trợ đó cho chương trình ứng dụng.
- Nhiệm vụ chính của OS là ánh xạ khái niệm file logic vào các thiết bị lưu trữ vật lý như đĩa cứng hoặc thiết bị NVM. Vì kích thước bản ghi vật lý của thiết bị có thể không giống với kích thước bản ghi logic nên có thể cần phải sắp xếp các bản ghi logic vào các bản ghi vật lý. Tác vụ này có thể được OS hỗ trợ hoặc để lại cho chương trình ứng dụng.

Summary

- Trong một hệ thống file, việc tạo các thư mục để cho phép các file được tổ chức là rất hữu ích. Thư mục single-level trong hệ thống nhiều người dùng gây ra vấn đề về đặt tên vì mỗi file phải có một tên duy nhất. Thư mục two-level giải quyết vấn đề này bằng cách tạo một thư mục riêng cho từng file của user. Thư mục liệt kê các tệp theo tên và bao gồm vị trí của file trên đĩa, độ dài, loại, chủ sở hữu, thời gian tạo, thời gian sử dụng lần cuối, v.v.
- Vì các file là cơ chế lưu trữ thông tin chính trong hầu hết các hệ thống máy tính nên việc bảo vệ file là cần thiết trên các hệ thống nhiều người dùng. Quyền truy cập vào các file có thể được kiểm soát riêng cho từng loại quyền truy cập—read, write, execute, append, delete, list directory, v.v. Việc bảo vệ file có thể được cung cấp bằng danh sách truy cập, mật khẩu hoặc các kỹ thuật khác.

Nội dung

- File-system Interface
- **File-system Implementation**
- File-System Internals

File-system Implementation

- Cấu trúc hệ thống file
- Vận hành hệ thống file
- Triển khai thư mục
- Phương thức phân bổ
- Quản lý không gian trống (free-space)
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

File-system Implementation

- **Cấu trúc hệ thống file**
- Vận hành hệ thống file
- Triển khai thư mục
- Phương thức phân bổ
- Quản lý không gian trống (free-space)
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

Cấu trúc hệ thống file

- 1 hệ thống file đặt ra 2 vấn đề
  - Xác định hệ thống file trông như thế nào đối với user → xác định file và các thuộc tính của nó, các thao tác được phép trên 1 file và cấu trúc thư mục để tổ chức file
  - Tạo ra các thuật toán và CTDL để ánh xạ hệ thống file logic vào các thiết bị lưu trữ thứ cấp vật lý

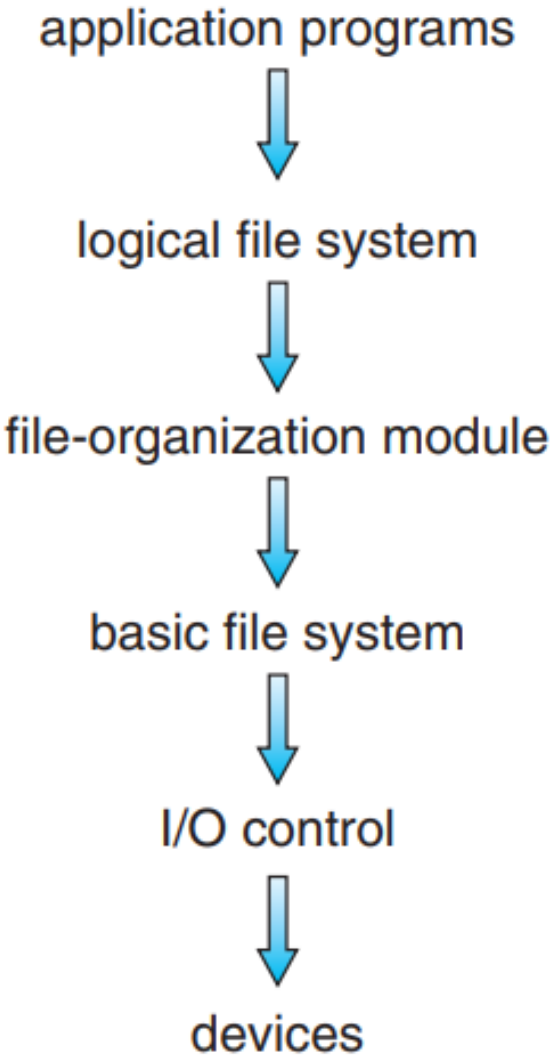


Figure 14.1 Layered file system.

Cấu trúc hệ thống file

- **I/O control** bao gồm device drivers và interrupt handlers để truyền thông tin giữa bộ nhớ chính và hệ thống đĩa.
- **Hệ thống file cơ bản** (được gọi là block I/O subsystem trong Linux) chỉ cần đưa ra các lệnh chung cho device driver thích hợp để đọc và ghi các khối trên thiết bị lưu trữ.
  - Đưa ra các lệnh tới ổ đĩa dựa trên các địa chỉ khối logic
  - Liên quan đến I/O scheduling
  - Quản lý bộ đệm, bộ nhớ đệm (chứa các khối hệ thống file, thư mục và dữ liệu khác nhau)

Cấu trúc hệ thống file

- **Module tổ chức file** biết về các file và các khối logic của chúng. Các khối logic của mỗi file được đánh số từ 0 (hoặc 1) đến N.
  - Bao gồm trình quản lý không trang trống, theo dõi các khối chưa được phân bổ và cung cấp các khối này cho module tổ chức file khi được yêu cầu
- **Hệ thống file logic** quản lý thông tin siêu dữ liệu (metadata). Metadata bao gồm tất cả cấu trúc hệ thống file ngoại trừ dữ liệu thực tế (hay nội dung file).
  - Hệ thống file logic quản lý cấu trúc thư mục để cung cấp cho module tổ chức file thông tin mà nó cần.
  - Duy trì cấu trúc file thông qua các khối điều khiển file (FCB – file control block) – chứa thông tin về file, bao gồm owner, permission và vị trí
  - Chịu trách nhiệm bảo vệ file

File-system Implementation

- Cấu trúc hệ thống file
- **Vận hành hệ thống file**
- Triển khai thư mục
- Phương thức phân bổ
- Quản lý không gian trống (free-space)
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

Vận hành hệ thống file

- Có 2 cấu trúc triển khai hệ thống file
  - On-storage structure
  - In-memory structure
- Các cấu trúc này phụ thuộc vào OS và hệ thống file nhưng vẫn có những nguyên tắc chung

Vận hành hệ thống file

- On-storage structure – hệ thống file chứa thông tin về cách khởi động OS được lưu trữ ở đó, tổng số khối, số lượng và vị trí các khối trống, cấu trúc thư mục và các file riêng lẻ
  - Khối điều khiển khởi động (boot control block – per volume) có thể chứa thông tin mà hệ thống cần để khởi động OS từ ổ đó. Nếu đĩa không chứa OS, khối này trống.
  - Khối điều khiển ổ đĩa (volume control block – per volume) chứa thông tin chi tiết về ổ đĩa, kích thước của các khối, số lượng khối trống và con trỏ khối trống cũng như số lượng FCB trống và con trỏ FCB
  - Cấu trúc thư mục (per file system) được sử dụng để sắp xếp/ tổ chức file
  - FCB cho mỗi file chứa nhiều thông tin chi tiết về file

Vận hành hệ thống file

- In-memory structure – được sử dụng cho cả việc quản lý hệ thống file và cải thiện hiệu suất thông qua bộ nhớ đệm
  - Bảng in-memory mount chứa thông tin về từng ổ đĩa được gắn kết
  - Cấu trúc thư mục in-memory chứa thông tin thư mục của các thư mục được truy cập gần đây
  - Bảng mở file trên toàn hệ thống (system-wide open-file table) chứa bản sao FCB của mỗi file đang mở cũng như các thông tin khác
  - Bản mở file trên mỗi tiến trình (per-process open-file table) chứa các con trỏ tới các mục thích hợp trong bảng mở file trên toàn hệ thống, cũng như các thông tin khác, cho tất cả các file mà tiến trình đã mở
  - Bộ đệm giữ các khối hệ thống file khi chúng được đọc hoặc ghi và hệ thống file

Vận hành hệ thống file

- Để tạo 1 file mới, tiến trình gọi hệ thống file logic (biết định dạng của cấu trúc thư mục)
- Hệ thống file logic cấp phát 1 FCB mới

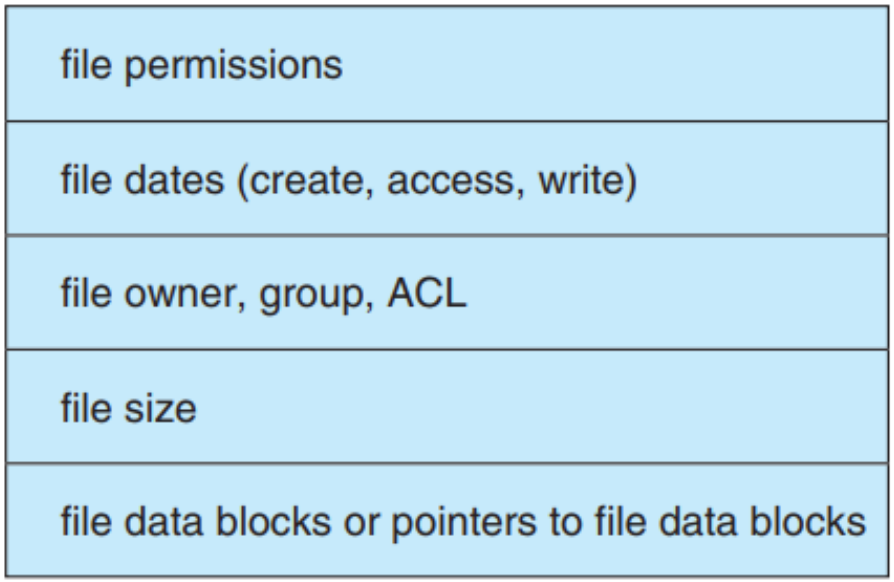


Figure 14.2 A typical file-control block.

Vận hành hệ thống file

- Khi file đã được tạo, nó có thể được sử dụng cho I/O
- Sử dụng lệnh open() để mở file, lệnh này chuyển tên file tới hệ thống file logic
  - Trước tiên sẽ tìm kiếm trên bảng mở file toàn hệ thống (system-wide open-file) để xem liệu file đó đã được sử dụng bởi 1 tiến trình khác hay chưa
  - Nếu đúng, 1 mục file mở trên mỗi tiến trình (per-process open-file table entry) sẽ được tạo trỏ tới system-wide open-file
  - Nếu chưa, cấu trúc thư mục sẽ tìm kiếm tên file đã cho. Sau khi tìm thấy, FCB sẽ được sao chép vào system-wide open-file
  - Tiếp theo, 1 mục được tạo trong bảng per-process open-file, với 1 con trỏ tới mục trong bảng system-wide open-file và một số trường khác
  - Lệnh gọi open() trả về 1 con trỏ tới mục thích hợp trong bảng per-process file-system

Vận hành hệ thống file

- Bảng system-wide open-file không chỉ lưu trữ FCB mà còn theo dõi số lượng tiến trình mở file
- Khi 1 tiến trình đóng file, mục trong bảng per-process bị xóa và số lần mở (open count) trong bảng system-wide sẽ giảm đi
- Khi tất cả người dùng đã mở file đóng nó lại, mọi metadata đã được cập nhật sẽ được sao chép trở lại cấu trúc thư mục dựa trên đĩa và mục trên bảng system-wide open-file sẽ bị xóa

Vận hành hệ thống file

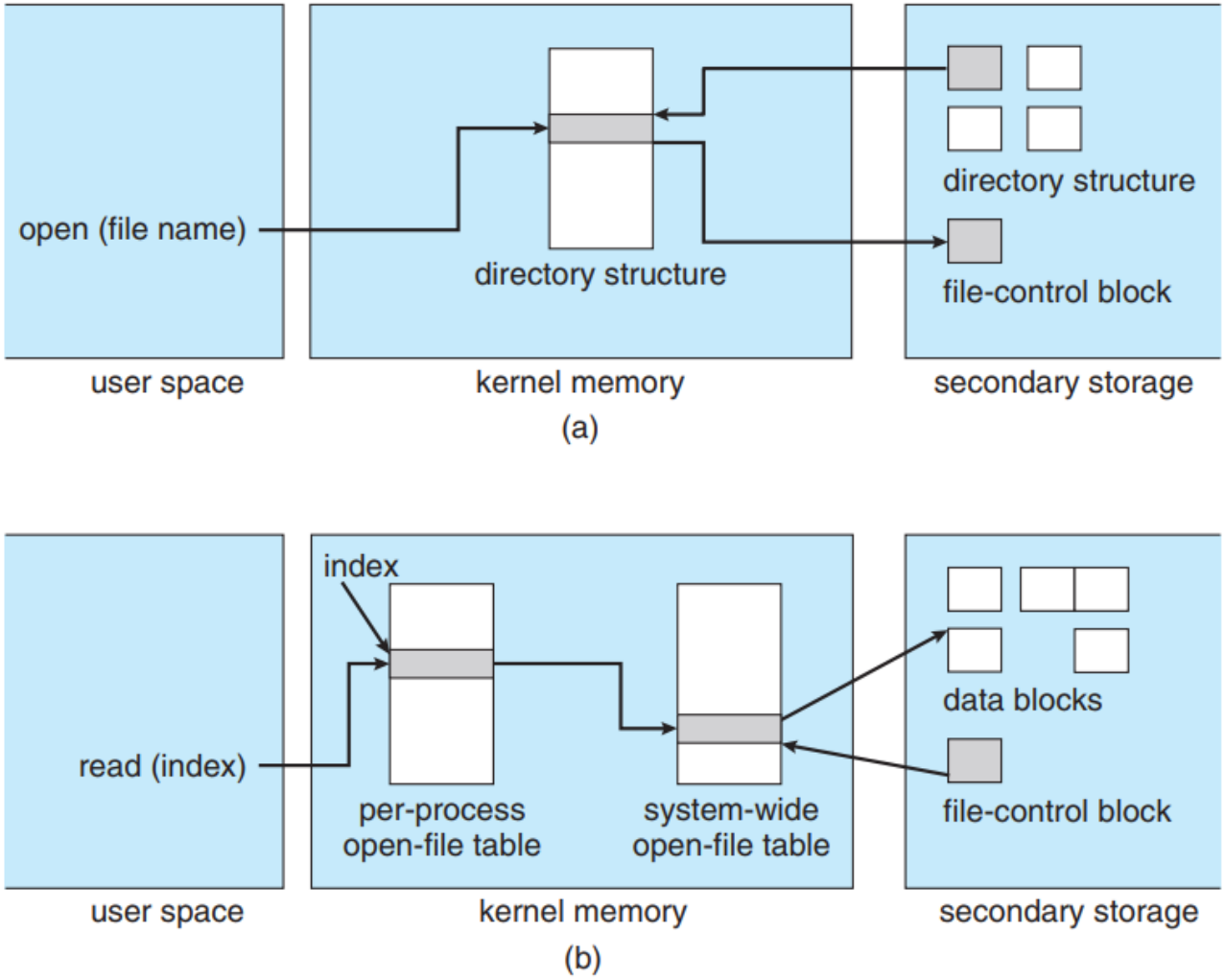


Figure 14.3 In-memory file-system structures. (a) File open. (b) File read.

File-system Implementation

- Cấu trúc hệ thống file
- Vận hành hệ thống file
- **Triển khai thư mục**
- Phương thức phân bổ
- Quản lý không gian trống (free-space)
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

Triển khai thư mục

- Linear List
- Hash Table

Linear list

- Phương pháp đơn giản nhất là sử dụng danh sách tuyến tính các tên file trỏ tới các khối dữ liệu
- Để tạo 1 file mới, cần chắc chắn không có file nào trùng tên trong thư mục, sau đó thêm 1 entry vào cuối thư mục.
- Để xóa 1 file, tìm kiếm file được đặt tên trong thư mục và sau đó giải phóng không gian được phân bổ cho nó
- Nhược điểm:
  - Tốn thời gian thực hiện
  - Tìm kiếm tuyến tính 1 file

Hash Table

- Một danh sách tuyến tích lưu trữ các mục trong thư mục + cấu trúc dữ liệu bảng băm
- Bảng băm lấy 1 giá trị được tình từ tên file và trả về một con trỏ tới tên file trong danh sách tuyến tính → giảm đáng kể thời gian tìm kiếm thư mục
- Việc chèn và xóa cũng khá đơn giản
- Nhược điểm:
  - Bảng băm có kích thước cố định → cần tạo mới bảng băm nếu có nhiều file hơn
  - Giải pháp: Sử dụng bảng băm tràn (mỗi mục băm có thể là 1 danh sách liên kết thay vì 1 giá trị riêng lẻ) → cũng giải quyết vấn đề xung đột

File-system Implementation

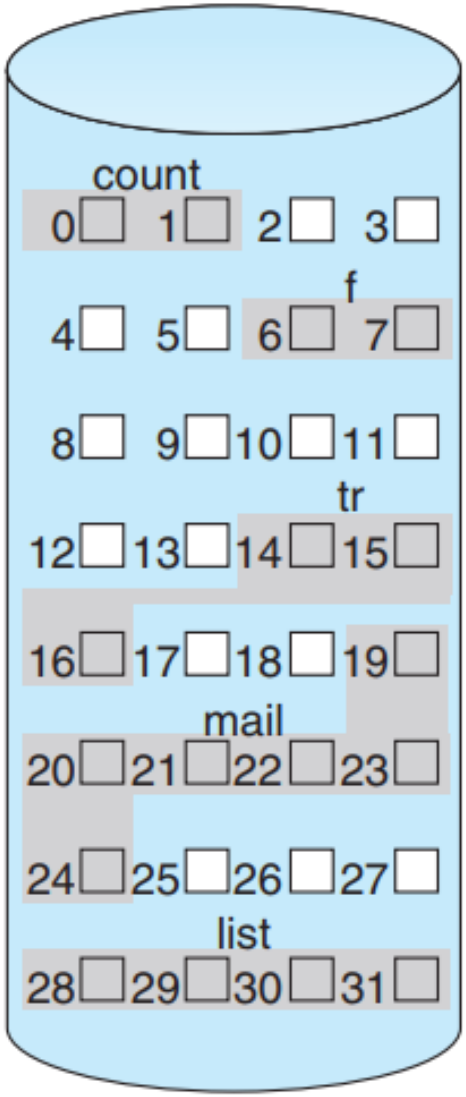
- Cấu trúc hệ thống file
- Vận hành hệ thống file
- Triển khai thư mục
- **Phương thức phân bổ**
- Quản lý không gian trống (free-space)
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

Phương thức phân bổ/ cấp phát

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation

Cấp phát liên tục

- Mỗi file chiếm các khối liền kề trên thiết bị
- Số lần tìm kiếm và thời gian tìm kiếm cần thiết để truy cập là tối thiểu
- Cấp phát liền kề của 1 file được xác định bởi địa chỉ của khối đầu tiên và độ dài (tính theo đơn vị khối) của file



directory

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

Figure 14.4 Contiguous allocation of disk space.

Cấp phát liên tục

- Nhược điểm:
  - Tìm không gian cho file mới → chậm hơn so với những hệ thống khác
  - Phân mảnh ngoài (external fragmentation)
  - Xác định không gian cần thiết cho 1 file

Cấp phát liên kết

- Cấp phát liên kết giải quyết các vấn đề của cấp phát liên tục
- Mỗi file là 1 danh sách các khối lưu trữ được liên kết; các khối có thể nằm rải rác ở bất cứ đâu trên thiết bị
- Thư mục chứa con trỏ tới khối đầu tiên và khối cuối cũng của file

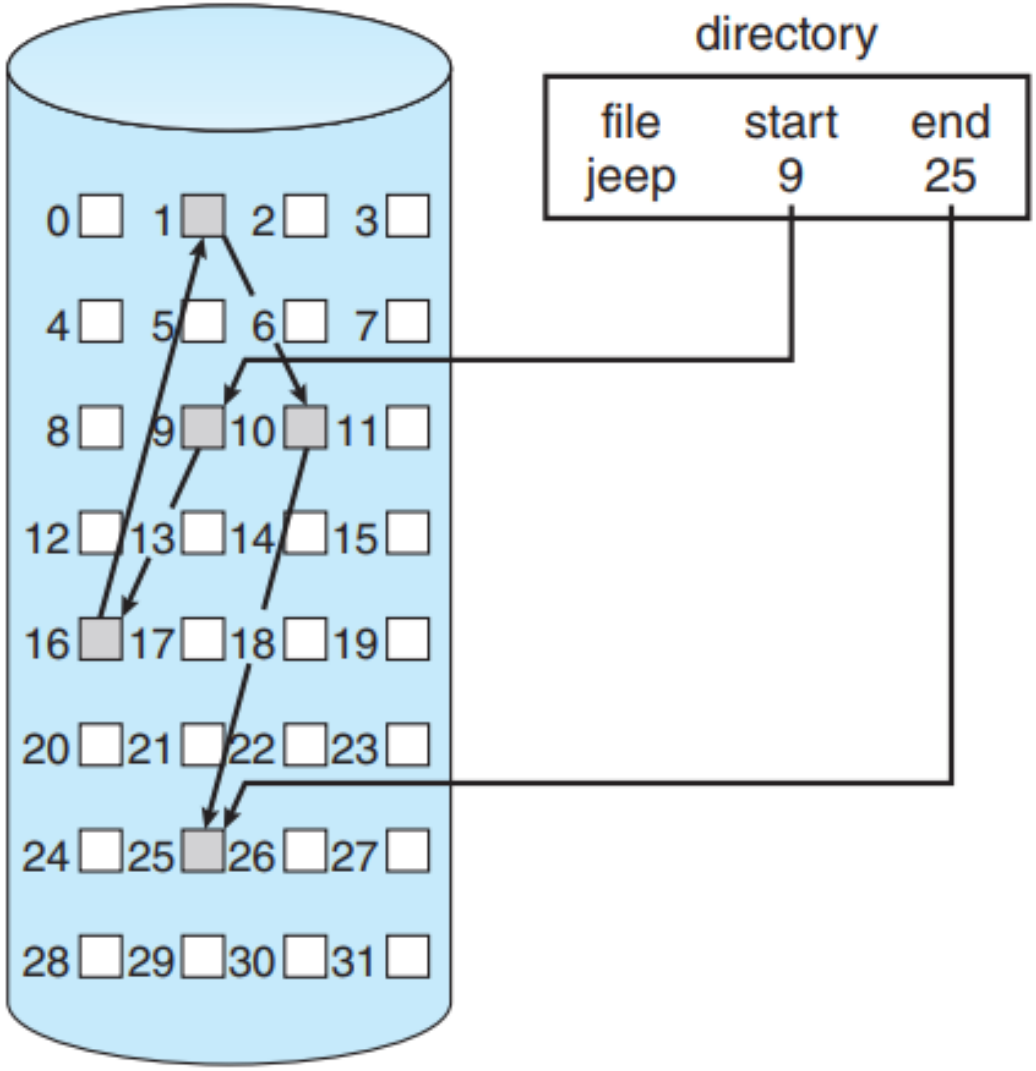


Figure 14.5 Linked allocation of disk space.

Cấp phát liên kết

- Để tạo 1 file mới, chỉ cần tạo 1 entry trong thư mục, mỗi entry chứa 1 con trỏ tới khối đầu tiên của file
- Để đọc file, chỉ cần đi theo con trỏ từ khối này sang khối khác
- Không có external fragmentation
- Kích thước của file không cần khai báo khi tạo file, file có thể tiếp tục được mở rộng miễn là còn các khối trống
- Nhược điểm:
  - Chỉ sử dụng hiệu quả cho file truy cập tuần tự
  - Mỗi file yêu cầu nhiều dung lượng hơn 1 chút do cần lưu con trỏ
  - Độ tin cậy – trường hợp 1 con trỏ bị mất hoặc bị hỏng

Cấp phát lập chỉ mục

- Giải quyết được vấn đề truy cập ngẫu nhiên
- Mỗi file có khối chỉ mục riêng, là 1 mảng địa chỉ khối lưu trữ. Mục thứ i trong khối chỉ mục trỏ đến khối thứ i của file

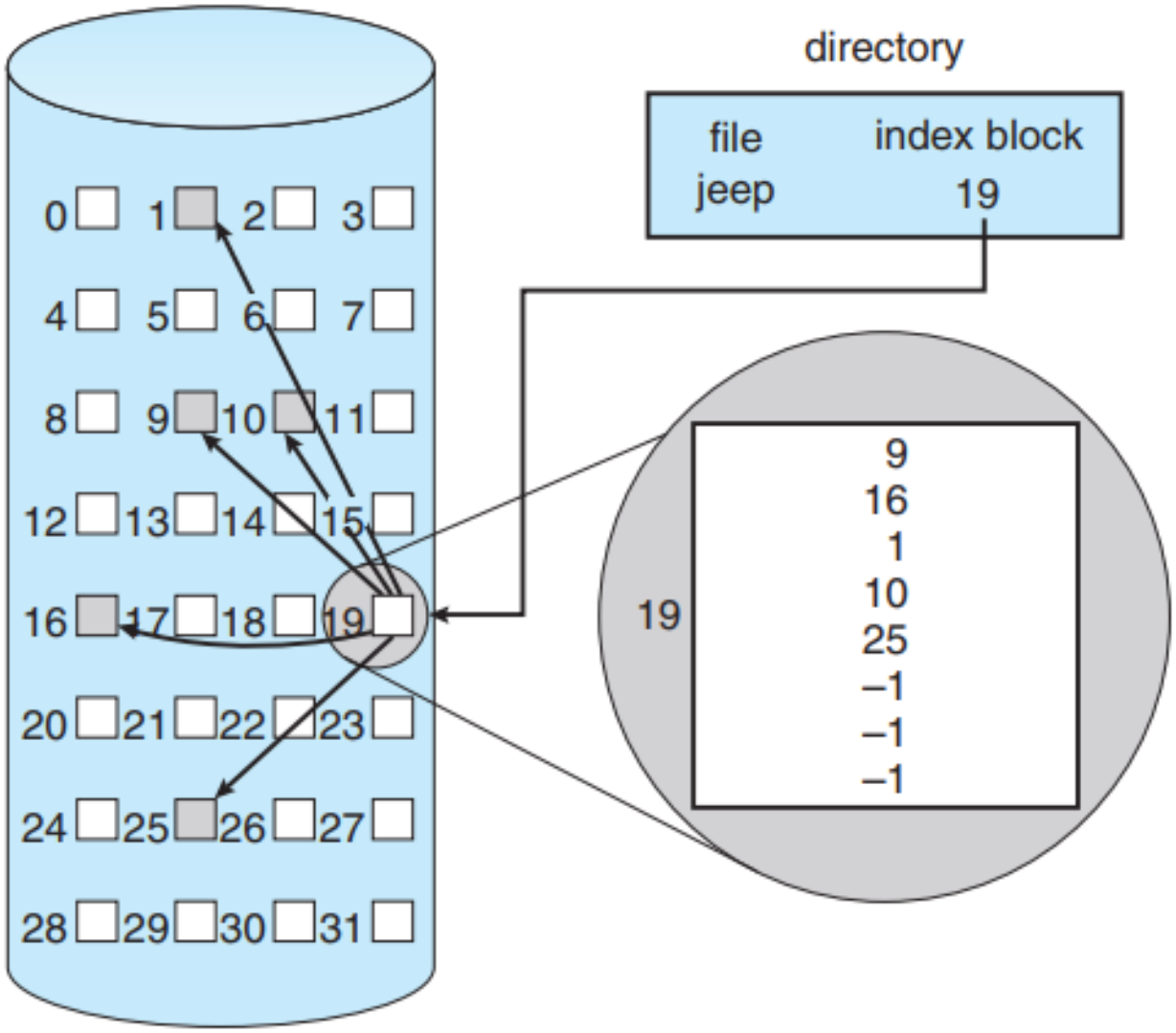


Figure 14.7 Indexed allocation of disk space.

Cấp phát lập chỉ mục

- Phân bổ chỉ mục hỗ trợ truy cập trực tiếp, không bị phân mảnh ngoài
- Nhược điểm:
  - Lãng phí không gian – chi phí con trỏ của khối chỉ mục thường lớn hơn chi phí con trỏ của cấp phát liên kết

File-system Implementation

- Cấu trúc hệ thống file
- Vận hành hệ thống file
- Triển khai thư mục
- Phương thức phân bổ
- **Quản lý không gian trống (free-space)**
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

Quản lý không gian trống

- Bit Vector (bitmap)
- Linked List
- Grouping
- Couting
- Scape Maps
- TRIMing Unused Blocks

File-system Implementation

- Cấu trúc hệ thống file
- Vận hành hệ thống file
- Triển khai thư mục
- Phương thức phân bổ
- Quản lý không gian trống (free-space)
- Hiệu quả và hiệu suất
- Khôi phục
- VD: Hệ thống file WAFL
- Summary

Nội dung

- File-system Interface
- File-system Implementation
- **File-System Internals**

File-System Internals

- Hệ thống file
- Mount hệ thống file
- Phân vùng và mounting
- Chia sẻ file
- Hệ thống file ảo
- Hệ thống file từ xa (Remote File Systems)
- Ngữ nghĩa nhất quán (Consistency Sematics)
- NFS
- Summary