Hệ thống tệp

Thursday, April 16, 2020 9:15 PM

Quan điểm về tệp

Câu hỏi
Thế nào là một tệp? Trong những thứ sau cái nào là tệp?

1) Văn bản có tên a.txt trên đĩa
2) Thư mục có tên TaiLieu
3) Thiết bị bàn phím
4) Thiết bị màn hình
5) Thiết bị máy in
6) Thiết bị lưu trữ (đĩa cứng)
7) Socket mạng

Khái niệm tệp trong môn Tin học cơ sở:
Tệp là một đối tượng trong máy tính cho phép đọc/ghi dữ liệu thông qua các lệnh/lời gọi do hệ điều hành cung cấp.

1) Văn bản a.txt cho phép đọc/ghi dữ liệu => là một tệp
2) Thư mục cho phép đọc/ghi nội dung trong thư mục => là một tệp. Đây là một tệp có cấu trúc, mỗi bản ghi chứa tên tệp trong thư mục, thuộc tính tệp (quyền truy cập, kích thước, ngày tháng vv.)
3) Bàn phím là một thiết bị vào/ra chỉ đọc (RO), cho phép đọc dữ liệu nhập bởi người dùng => là một tệp.
   Ví dụ:
   - Windows cmd: $copy\ CON\ a.txt$, copy từ tệp bàn phím Console vào tệp a.txt
   - Tệp bàn phím được quy định là đầu vào mặc định stdin = 0 trong ngôn ngữ C: $fscanf(stdin, "\%d", \&i);\ /*\ read\ from\ keyboard\ */$
4) Màn hình là thiết bị vào/ra chỉ ghi (WO), cho phép ghi dữ liệu ra => là một tệp.
   Ví dụ:
   - Tệp /dev/tty trong Linux/Unix, chẳng hạn $clear\ > /dev/tty$
   - Tệp màn hình được quy ước là đầu ra mặc định của stdout=1, và là đầu ra của stderr=2, chẳng hạn $printf(stderr, "Lỗi\backslash n");$
5) Máy in là thiết bị vào/ra chỉ ghi (WO), cho phép ghi dữ liệu ra => là một tệp.
   Ví dụ:
   - Windows cmd: $copy\ a.txt\ prn$
   - Linux/Unix: $sudo\ cat\ a.txt\ > /dev/lp$
6) Các thiết bị lưu trữ cho phép đọc/ghi dữ liệu => là một tệp.

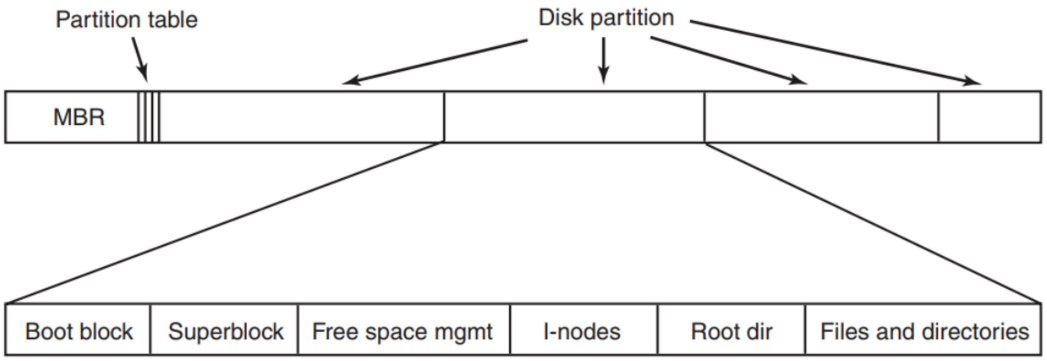
Ví dụ:

- Đĩa cứng thứ nhất được đặt tên tệp là /dev/sda (SCSI) hoặc /dev/hda (IDE)
- Đĩa cứng thứ hai được đặt tên tệp là /dev/sdb (SCSI) hoặc /dev/hdb (IDE)

7) Socket mạng: phần này được học trong môn Mạng máy tính. Socket cho phép đọc/ghi dữ liệu => là một tệp.
   Ví dụ:

```
s = socket(domain, type, protocol);
write(s, buf, strlen(buf));
```

Tổ chức đĩa và hệ thống tệp



Tổ chức đĩa

Partition Table: Chứa thông tin về khối bắt đầu và kết thúc của các partitions. Trong các partitions, có một partition là active, được dùng để khởi động hệ thống.

Master Boot Record: Khi máy được bật lên, BIOS đọc và thực hiện đoạn mã MBR. Đoạn mã MBR định vị partition khởi động và tải khối Boot Block vào trong bộ nhớ. Chương trình trong Boot Block sẽ tải hệ điều hành trong partition đó để hoàn tất khởi động hệ thống.

Tổ chức hệ thống tệp

Một partition được tổ chức như sau

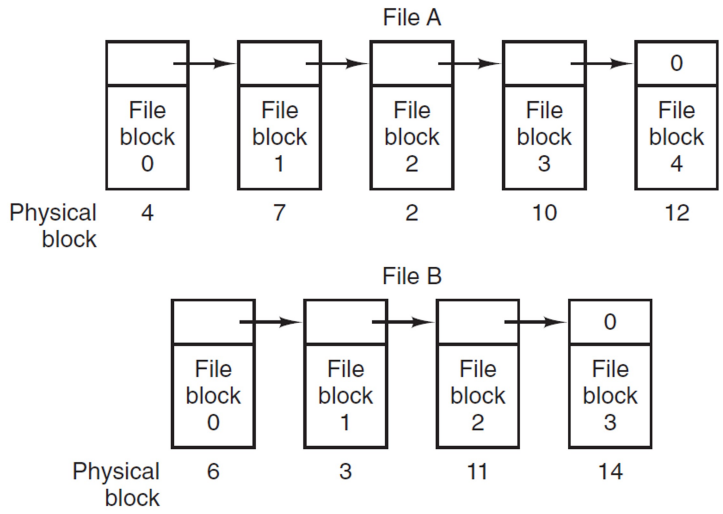
- Boot block: chứa đoạn mã mồi tải hệ điều hành vào bộ nhớ để khởi động hệ thống

- Superblock: chứa thông tin về các khối đĩa khác, chẳng hạn vùng bảng chỉ mục, thư mục gốc, khối đĩa dữ liệu vv.
- Free space management: quản lý danh sách các khối đĩa rỗi, dùng cấp phát cho tệp
- I-nodes: bảng chỉ mục i-nodes dùng để quản lý các khối đĩa được cấp phát cho 1 tệp
- Root dir: vùng cấp phát cho tệp thư mục gốc
- Files and Directories: gồm các khối đĩa dùng cấp phát cho các thư mục con và các tệp

Lưu trữ nội dung tệp

Giả thiết là cho trước vị trí khối đĩa đầu tiên của tệp, chúng ta tìm hiểu cách thức liên kết và quản lý các khối đĩa còn lại của tệp

Móc nối các khối đĩa



Mỗi khối đĩa được chia làm 2 phần

- Con trỏ móc nối tới khối tiếp theo
- Phần còn lại chứa nội dung tệp

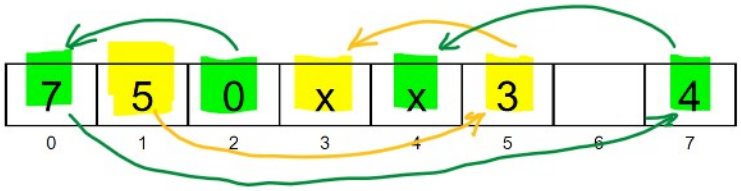
Nhận xét:

- Kích thước phần còn lại của khối đĩa lưu trữ nội dung tệp không còn là bội mũ 2 => không thuận tiện định vị bản ghi trong tệp
- Để nhảy tới nội dung trong một khối đĩa n, bắt buộc phải truy cập tuần tự tới $n-1$ khối đĩa trước đó

Móc nối qua bảng cấp phát khối đĩa

Tương tự như phương pháp móc nối các khối đĩa, nhưng phần con trỏ móc nối được tách ra và lưu riêng vào một bảng (bảng A). Giả sử khối đĩa i chứa nội dung tệp thì khối đĩa tiếp theo của khối i được trỏ bởi $A[i]$ trong bảng. Nếu i là khối cuối thì $A[i] = NULL$.

Ví dụ: Tệp a.txt có khối đĩa đầu tiên là 1, khối đĩa đầu của tệp b.txt là 2

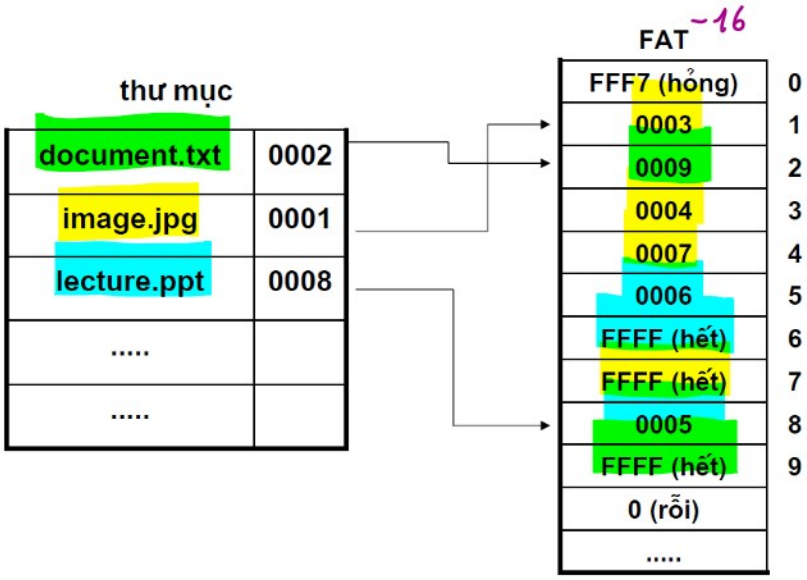


Minh họa với File Allocation Table trên DOS/Windows:

Tổ chức đĩa

Boot block	FAT	FAT copy	Root dir	Data blocks ...

Bảng cấp phát tệp



2^{16} khối x 16kB
$= 2^{16} \times 2^4$ kB
$= 2^{20}$ kB = 2GB

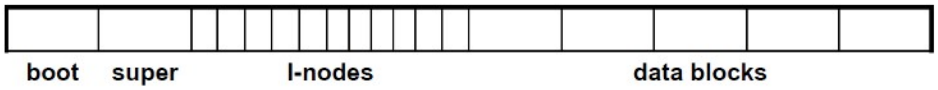
$2^{32} \times 4B = 2^{39}$ B
$= 2^{29}$ aB
$= 2^{19}$ MB
$= 2^4$ GB

Nhận xét về bảng cấp phát tệp:

- Phải lưu bảng FAT trong RAM => chiếm kích thước lớn, không phù hợp để quản lý các hệ thống đĩa lớn
- FAT không hỗ trợ quản lý quyền thâm nhập tệp theo người dùng

Bảng chỉ mục với cấu trúc i-node

Mỗi tệp sẽ được cấp một i-node. Cấu trúc i-node chứa con trỏ tới các khối đĩa chứa nội dung tệp.

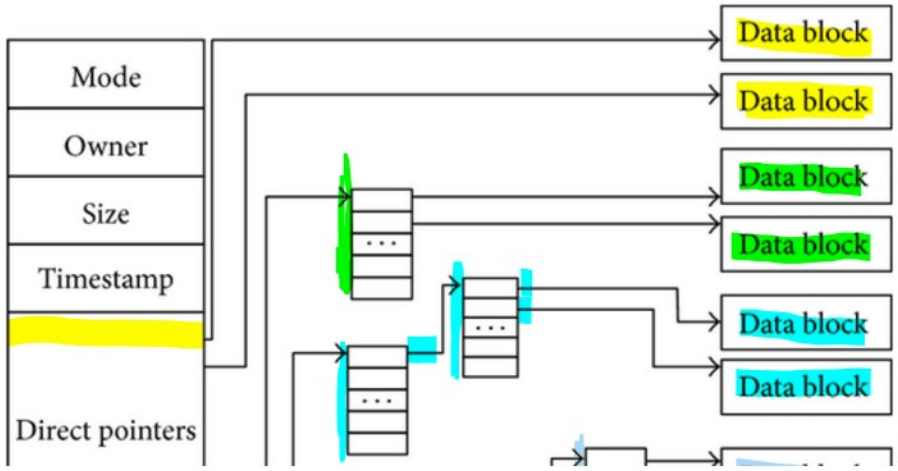


Cấu trúc i-node

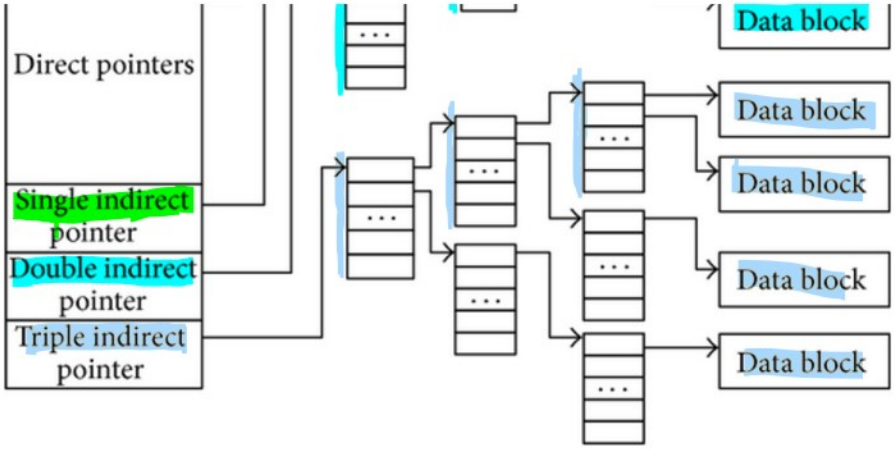
Mode
Link count
UID
GID
File size
Time
12 direct ptrs
single indirect
double indirect
triple indirect

Mode: các bits quyền truy cập tệp -rwx rwx rwx
Linkcount: đếm số tệp sử dụng chung một i-node
UID/GID: chủ/nhóm sở hữu tệp
Size, time: kích thước, ngày giờ
Direct pointers: trỏ trực tiếp tới khối đĩa dữ liệu
Indirect pointers: trỏ tới bảng chỉ mục con

Cấu trúc i-node và bảng chỉ mục nhiều mức



4KB/khối
12.4 = 48KB



Các con trỏ trực tiếp

- Trỏ trực tiếp tới các khối đĩa chứa nội dung tệp
- Phù hợp với các tệp cỡ nhỏ
- Ví dụ: inode chứa 12 con trỏ trực tiếp, mỗi khối đĩa 4KB, kích thước tệp nhỏ tối đa là $4\times 12 = 48$ KB

Con trỏ gián tiếp đơn, sử dụng khi tệp có kích thước lớn hơn

- Trỏ tới một bảng chỉ mục chứa các con trỏ trực tiếp tới khối đĩa
- Ví dụ: 4KB/khối đĩa, con trỏ chỉ mục 32 bits, mỗi bảng chỉ mục chứa được 1024 con trỏ. Vậy kích thước tệp tối đa là $4\times (12 + 1024) = 4144$ KB

Con trỏ gián tiếp kép, sử dụng khi tệp có kích thước lớn hơn nữa

- Trỏ tới một bảng chỉ mục chứa các con trỏ gián tiếp đơn
- Ví dụ: kích thước tệp tối đa là $4\times (12 + 1024 + 1024^2) \sim = 4$ GB

Con trỏ gián tiếp mức 3, sử dụng cho tệp kích thước rất lớn

- Trỏ tới một bảng chi mục chứa các con trỏ gián tiếp kép
- Ví dụ: kích thước tệp tối đa là $4\times (12 + 1024 + 1024^2 + 1024^3) \sim = 4$ TB

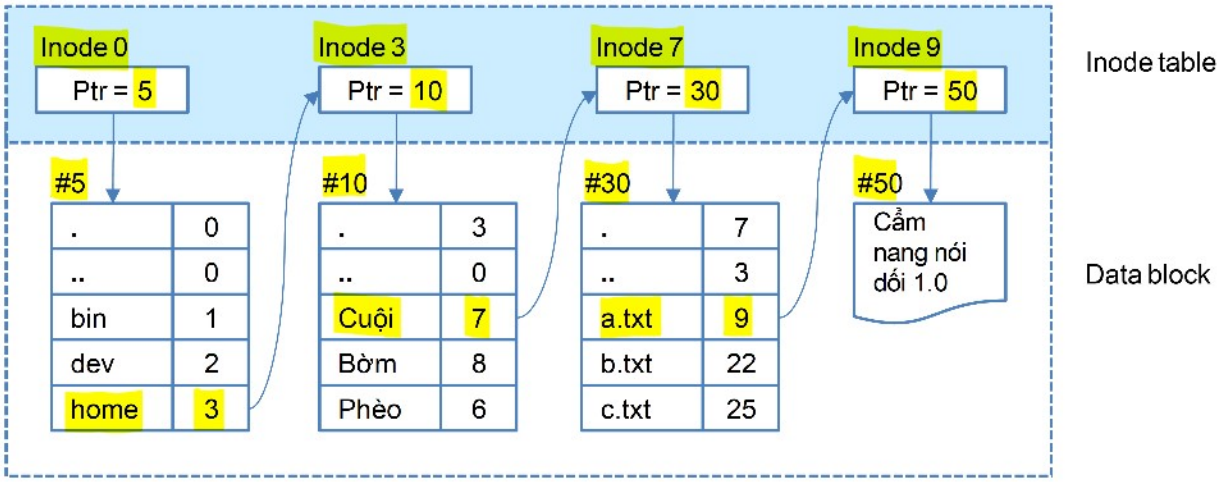
Truy cập tới khối đĩa đầu tiên của tệp

Thư mục là một tệp bản ghi, mỗi bản ghi có cấu trúc

Tên tệp	Số i-node

Các bước phân tích đường dẫn tệp để tìm tới khối đĩa đầu tiên:

Xét trường hợp /home/Cuội/a.txt



1) Khối đĩa đầu của thư mục gốc '/' trỏ bởi i-node số 0, tức là khối #5
2) Đọc khối đĩa #5, tìm tệp 'home', lấy được i-node số 3, i-node 3 trỏ tới khối đĩa #10
3) Đọc khối đĩa #10, tìm tệp 'Cuội', lấy được i-node số 7, i-node 7 trỏ tới khối đĩa #30
4) Đọc khối đĩa #30, tìm tệp 'a.txt', lấy được i-node số 9, i-node 9 trỏ tới khối đĩa #50
5) Khối đĩa #50 là khối đĩa đầu tiên của tệp /home/Cuội/a.txt

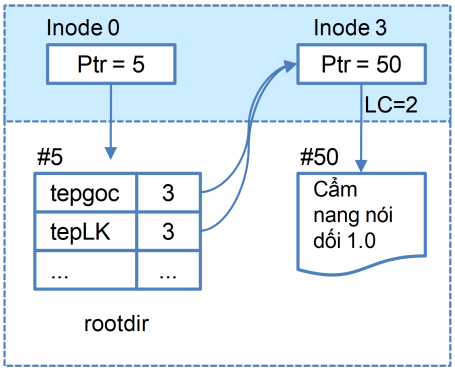
Tệp liên kết

Liên kết cứng - hard link

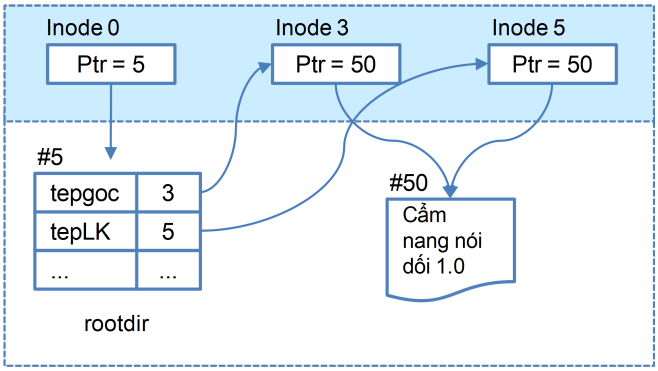
Ví dụ về lệnh ln tạo liên kết cứng

```
$ whoami
      ntb
$ ln tepgoc  teplienket
      -rw-rw-rw-   2 nva   sinhvien       15 Mar 28 16:31 tepgoc
      -rw-rw-rw-   2 nva   sinhvien       15 Mar 28 16:31 teplienket
$ rm tepgoc
      -rw-rw-r--   1 nva   sinhvien       15 Mar 28 16:31 teplienket
$ cat teplienket
      ... nội dung tệp gốc ban đầu ...
```

Hai phương án cài đặt



PA1: 2 tệp dùng chung 1 inode



PA2 : 2 tệp dùng 2 inode riêng, trỏ cùng tới 1 khối đĩa

Phương án nào khả thi hơn?

Phương án 1 do chỉ cần dựa trên biến đếm Link count để quyết định có giải phóng khối đĩa chứa nội dung tệp hay không

Liên kết mềm - symbolic link

Ví dụ về lệnh ln -s tạo liên kết mềm

$ whoami

ntb

$ ln -s tepgoc.txt teplienket.txt

-rw-rw-rw- 1 nva sinhvien 15 Mar 28 16:31 tepgoc

lrwxrwxrwx 1 ntb sinhvien 15 Mar 28 16:31 teplienket=>tepgoc

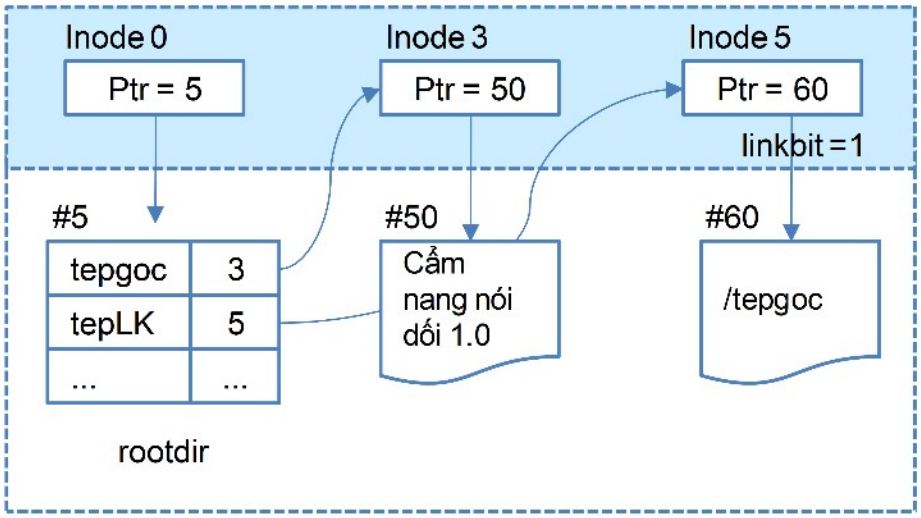
$ rm tepgoc

lrwxrwxrwx 1 ntb sinhvien 15 Mar 28 16:31 teplienket=>tepgoc

$ cat teplienket

Lỗi: find not found

Cài đặt tệp liên kết mềm



Tệp liên kết có nội dung là đường dẫn tới tệp gốc

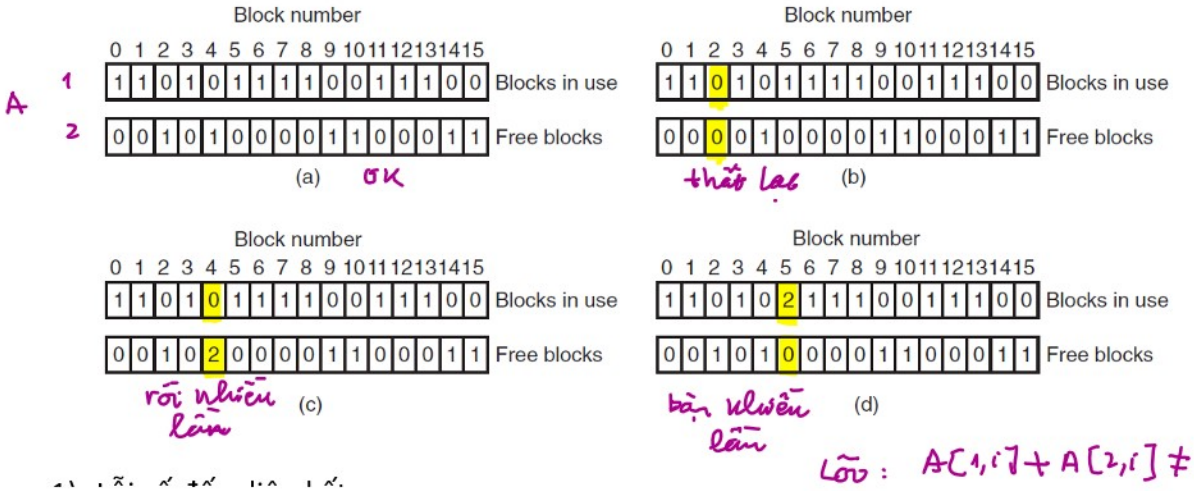
Tệp gốc và tệp liên kết mềm có 2 i-nodes độc lập, do vậy thuộc tính của 2 tệp này khác nhau.

Tệp liên kết mềm có bit thuộc tính l được bật và nội dung tệp chứa đường dẫn tới tệp gốc.

Phát hiện và xử lý lỗi với hệ thống tệp

Do hệ thống tệp được xây dựng từ các móc nối inode, khối đĩa qua nhiều bước nên có thể xảy ra sự cố không đảm bảo tính nhất quán. Các lỗi vi phạm tính nhất quán có thể xảy ra là:

1) Số đếm liên kết cứng trong một i-node không trùng khớp với số tệp đang sử dụng i-node đó
2) Lỗi về trạng thái khối đĩa: (a) Không có lỗi; (b) Lỗi thất lạc; (c) Trùng lặp khối rỗi; (d) Trùng lặp khối dữ liệu



rơi nhiều lần (c)

bận nhiều lần (d)

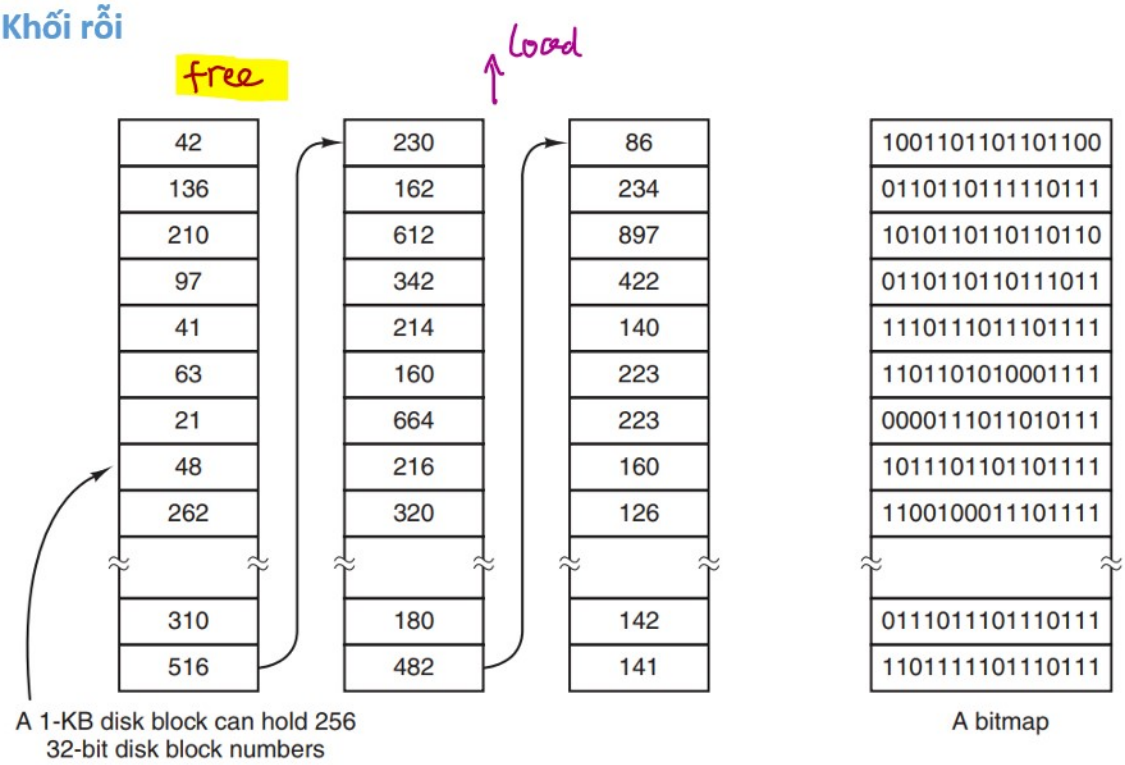
Lỗi: A[1,i] + A[2,i] ≠ 1

1) Lỗi số đếm liên kết:
   Giải pháp: Quét tất cả các tệp, đếm số tệp sử dụng chung 1 i-node, cập nhật lại vào trường linkcount trong i-node

2) Lỗi trạng thái khối đĩa
- Lỗi thất lạc khối đĩa: bổ sung vào danh sách các khối rỗi
- Lỗi lặp khối đĩa rỗi: xóa bớt khỏi danh sách khối rỗi
- Lỗi lặp khối đĩa dữ liệu: thuê thêm 1 khối rỗi, sao nội dung, cho mỗi tệp trỏ tới một khối riêng biệt
- Lỗi vừa rỗi, vừa bận: xóa khỏi danh sách khối rỗi

Quản lý khối rỗi, khối hỏng, quota

Khối rỗi



Sử dụng bản đồ bit

Trạng thái của mỗi khối đĩa được biểu diễn bởi 1 bit ở vị trí tương ứng.

Ưu điểm: kích thước lưu trên đĩa nhỏ

Nhược điểm: phải lưu toàn bộ bản đồ bit trong RAM phục vụ cho việc cấp phát

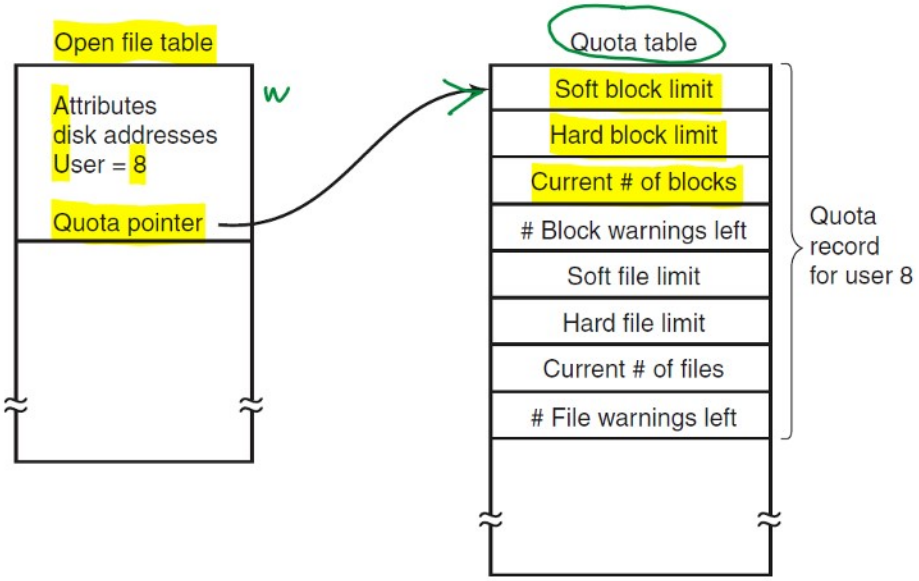
Sử dụng danh sách các khối rỗi
Tận dụng các khối đĩa rỗi móc nối với nhau. Mỗi khối rỗi chứa danh sách các khối rỗi có thể dùng cấp phát.
Nhược điểm: kích thước trên đĩa lớn hơn nhiều so với bản đồ bit
Ưu điểm: chỉ cần lưu 1 khối danh sách rỗi trong RAM phục vụ cho việc cấp phát.
Các khối móc nối cũng là các khối rỗi, do vậy không chiếm không gian nhớ cố định.

Khối hỏng

Các khối hỏng phát hiện khi format ở mức thấp sẽ được thay thế bởi các khối dự phòng nằm ở cuối mỗi rãnh, không cần tới sự can thiệp của hệ điều hành.

Các khối hỏng phát hiện sau khi format được hệ điều hành xếp vào một tệp ẩn, không truy cập được, và như vậy không bị dùng để cấp phát cho tệp.

Quota đĩa



Mỗi người dùng được quản lý qua một bản ghi quota và được tải vào bộ nhớ mỗi khi có tệp liên quan tới người dùng đó được mở.
Khi một file được mở, một con trỏ quota được bổ sung thêm vào trong bảng Open File, trỏ tới bản ghi quota của chủ sở hữu tệp. Việc tăng/giảm kích thước tệp sẽ làm tăng/giảm số lượng blocks đã cấp trong bản ghi quota.

Soft limit là giới hạn có thể được vi phạm trong một số lần cảnh báo nhất định.
Hard limit là giới hạn không được phép vượt qua.

Quản lý quyền truy cập tệp

Ma trận thưa biểu diễn quyền truy cập tệp

Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2
	Object							
1	Read	Read Write						
2			Read	Read Write Execute	Read Write		Write	
3						Read Write Execute	Write	Write

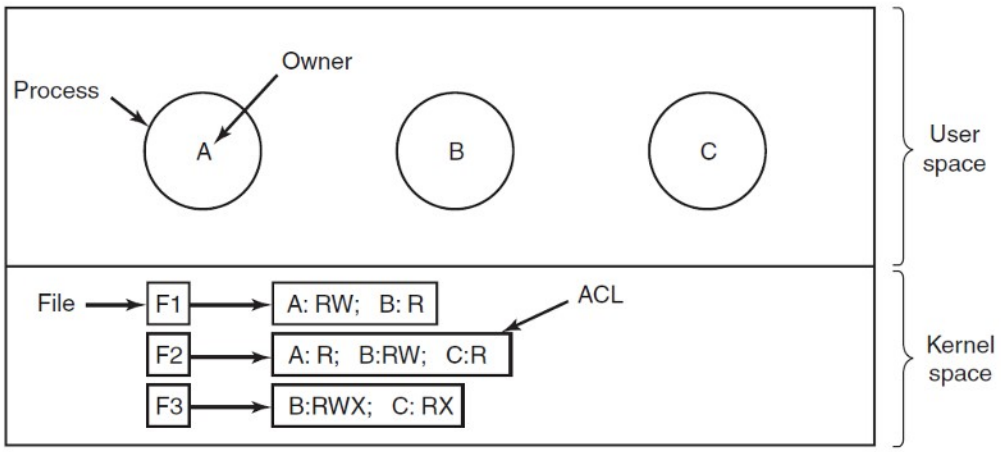
Object: thể hiện các tài nguyên máy tính
Domain: thể hiện người sử dụng (phạm vi thao tác của người dùng)

Nhược điểm: ma trận thưa chiếm nhiều không gian lưu trữ

Nén ma trận thưa theo cột - phương pháp Access Control List (ACL)

Đối với mỗi đối tượng, gắn kèm danh sách các miền và quyền thao tác đi kèm.
Miền có thể là nhóm hoặc người dùng cụ thể.

Ví dụ:



Danh sách ACL đối với file F1: người dùng A có quyền RW, người dùng B quyền R.

Khi một tiến trình của người dùng A định truy cập tệp F1, hệ thống sẽ tìm trong danh sách ACL để xem tiến trình có được phép truy cập không.

Nhận xét:

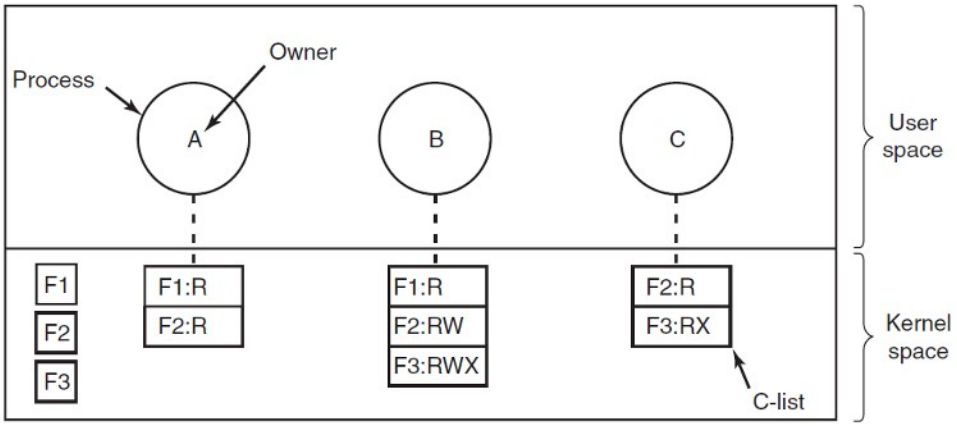
- Thao tác tìm kiếm quyền thực tế cũng mất 1 khoảng thời gian do phải tải danh sách quyền để kiểm tra và nếu ACL chứa quyền cho nhóm thì phải đối chiếu người dùng có thuộc nhóm đó không
- Tuy nhiên ACL định nghĩa nhóm và vai trò (role), cho phép dễ dàng tước quyền truy cập đối với 1 nhóm lựa chọn người dùng, chẳng hạn loại bỏ quyền với mọi đối tượng, trừ người dùng nva của nhóm $student$

  Tệp F1 - $nva, student: RW$; *,*: $(none)$
- Đây là phương pháp thông dụng, được dùng trong Windows, Unix ACL

Nén ma trận thưa theo hàng - phương pháp Capability-List (C-list)

Đối với mỗi miền (người dùng), gắn kèm danh sách các tệp/tài nguyên và quyền truy cập tương ứng (capability).

Ví dụ:



Tiến trình của người dùng A được gắn với danh sách các tệp mà nó được quyền truy cập tới.

Nhận xét:

- C-List cho phép truy cập trực tiếp tới tài nguyên theo con trỏ capability trong danh sách mà không cần tìm kiếm, đối chiếu quyền như ACL
- Mỗi capability là một con trỏ tới tài nguyên và do vậy có thể trỏ tới một C-list khác, cho phép phân chia sub-domain theo cơ chế phân quyền
- C-List không cho phép thu hồi quyền đối với một miền theo tiêu chí lựa chọn dễ dàng như ACL, vì phải tìm toàn bộ danh sách gắn với tất cả các miền trong hệ thống
- Việc xóa người dùng nhưng chưa xóa C-List hoặc ngược lại cũng gây sự cố trong hệ thống
- Đây là phương pháp được sử dụng trong HĐH mạng Novell Netware trước đây

Vấn đề lưu đệm (caching)

Các vùng nhớ thường xuyên truy cập trên đĩa cần phải được lưu đệm trong RAM để cải thiện hiệu suất, chẳng hạn bảng i-node.

Vùng nhớ đệm là một bộ đệm có kích thước hạn chế, do vậy chỉ lưu những gì thường xuyên dùng tới nhất.

Câu hỏi: có thể áp dụng các thuật toán hoán đổi trang nhớ ảo như LRU cho vùng nhớ đệm đĩa không?

- Có, nhưng cần phải giải quyết vấn đề đảm bảo tính nhất quán đối với các khối đĩa quan trọng, thường xuyên được dùng trong hệ thống tệp, chẳng hạn bảng i-node
    - Cần định kỳ cập nhật nội dung các khối này ra đĩa
    - write-through cache: cache đọc, nhưng không cache ghi