

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
Khoa học - Kỹ thuật Máy tính



HỆ ĐIỀU HÀNH

Bài thực hành số 10

File System

GVHD: Hoàng Lê Hải Thanh

SV thực hiện: Trương Hoài Nam – 1813175

TP. HỒ CHÍ MINH, THÁNG 06/2020



Mục lục

| | |
|---|----------|
| 1 EXERCISE | 2 |
| 1.1 Suppose you have a 10GB drive ($2^{30} * 10$ bytes). How many entries would a FAT need if the block size is 1KB (2^{10}), 4KB, and 8KB?? Please explain your caculation. | 2 |
| 1.2 Consider a system where free space is kept in a free-space list | 2 |

1 EXERCISE

1.1 Suppose you have a 10GB drive ($2^{30} * 10$ bytes). How many entries would a FAT need if the block size is 1KB (2^{10}), 4KB, and 8KB?? Please explain your calculation.

* Block size: 1 KB = 2^{10} bytes

$$\text{Số entries} = \frac{\text{size of drive}}{\text{block size}} = \frac{2^{30} * 10}{2^{10}} = 2^{20} * 10 = 10\,485\,760$$

* Block size: 4 KB = 2^{12} bytes

$$\text{Số entries} = \frac{\text{size of drive}}{\text{block size}} = \frac{2^{30} * 10}{2^{12}} = 2^{18} * 10 = 2\,621\,440$$

* Block size: 8 KB = 2^{13} bytes

$$\text{Số entries} = \frac{\text{size of drive}}{\text{block size}} = \frac{2^{30} * 10}{2^{13}} = 2^{17} * 10 = 1\,310\,720$$

1.2 Consider a system where free space is kept in a free-space list

a. Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list? Explain your answer.

- Nếu con trỏ trỏ đến free-space bị mất, vẫn có thể tái tạo lại free-space.

- Muốn tái tạo lại free-space, hệ điều hành phải quét hệ thống các tập tin bắt đầu từ gốc. Khi mỗi thư mục và tệp được quét, hệ điều hành sẽ đánh dấu các khối đĩa liên quan được sử dụng, từ đó HDH giữ riêng danh sách các khối đĩa bị hỏng (không thể đọc được, không thể ghi được). Thư mục chứa những tệp như trên vẫn còn tồn tại ở một số hệ thống.

b. Consider a file system similar to the one used by UNIX with indexed allocation. How many disk I/O operations might be required to read the contents of a small local file at /a/b/c? Assume that none of the disk blocks is currently being cached.

- Vì trong UNIX, các thư mục chứa các tệp theo dạng danh sách tên tệp/thư mục và con trỏ tới FCB (inodes).

Ở trường hợp tốt nhất, đọc từng thư mục hoặc tệp cần ba khối đĩa:

- + Khối đĩa để đọc inode cho tệp hoặc thư mục.
- + Khối đĩa để đọc khối chỉ mục được chỉ đến bởi nút.
- + Khối đĩa để đọc khối dữ liệu đầu tiên của tệp hoặc thư mục chính nó.

Giả sử ta biết địa chỉ của nút gốc khi bắt đầu, có 12 lần đọc là bắt buộc:

1. Đọc inode của root.
2. Đọc khối chỉ mục của root.
3. Đọc thư mục root.

4. Đọc inode của a.
5. Đọc một khối chỉ mục của a.
6. Đọc thư mục a.
7. Đọc inode của b.
8. Đọc khối chỉ mục của b.
9. Đọc thư mục b.
10. Đọc inode của c.
11. Đọc khối chỉ mục của c.
12. Đọc khối đầu tiên của tập tin c.

Nếu với double-indirect, một trường hợp xấu hơn là yêu cầu 4 lần đọc để có được khối thích hợp của một thư mục/tập tin vào bộ nhớ: đọc inode, đọc gián tiếp khối double, đọc khối gián tiếp, đọc khối dữ liệu. Do đó, sẽ mất tổng cộng 16 lần đọc.

c. Suggest a scheme to ensure that the pointer is never lost as a result of memory failure.

- Đó là lưu giữ con trỏ vào danh sách các free-space ở nhiều nơi khác nhau trong ổ đĩa.