

[CHUONG 01: GENERAL]

[
]

Câu 1:

Một tập tin thực thi sở hữu bởi root (ví dụ như ping). Cách nào để lỗi hệ điều hành cho phép một user bình thường có thể thực hiện file thực thi này.

- A. Hệ điều hành không cần làm gì hết.
- ☒ B. Hệ điều hành sẽ thay đổi user ID của quá trình sang Effective ID trùng với ID của tập tin thực thi.
- C. Chỉ có root mới thực thi được tập tin đã cho.
- D. Hệ điều hành thay đổi user ID của tập tin sang user của người dùng bình thường, rồi thực thi tập tin này.

[
]

Câu 2:

Ngắt gì được sử dụng để phục vụ quá trình định thời của hệ điều hành ?

- A. Ngắt I/O
- B. Ngắt mềm (trap)
- ☒ C. Ngắt đồng hồ (Timer)
- D. Ngắt đồng hồ và ngắt I/O

[CHUONG 03: THREAD MANAGEMENT]

[
]

Câu 1:

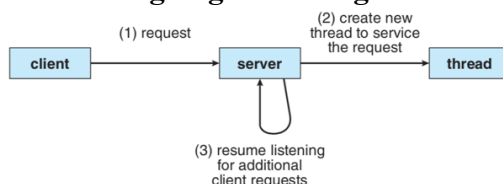
Tại sao các thread lại cần phải có lưu thanh ghi (register) và stack riêng (không giống như phần code, data, files) khi quản lý ?

- A. Lưu thanh ghi và stack riêng giúp thread tăng tốc được quá trình xử lý.
- B. Các thread thực thi trên các bộ xử lý hoặc lõi khác nhau nên cần lưu riêng.
- ☒ C. Các thread khác nhau thực thi độc lập và đây là 2 thành phần chính kiểm soát thực thi.
- D. Giúp cho hệ điều hành ghi nhớ được thread nào đang được gán trên bộ xử lý hoặc lõi nào.

[
]

Câu 2:

Apache (một máy chủ web rất phổ biến) sử dụng cơ chế one-thread-per-connection (một thread một kết nối) để phục vụ các yêu cầu đến giống như trong hình sau.



Giả thiết số lượng các yêu cầu kết nối đến máy chủ web dùng Apache lên đến vài trăm nghìn yêu cầu đồng thời. Hãy cho biết cơ chế one-thread-per-connection có còn hợp lý hay không ?

- A. Hợp lý. Số lượng thread lớn sẽ phân bổ cân bằng đến các bộ xử lý giúp đáp ứng hiệu quả.
- B. Hợp lý. Thời gian chuyển đổi ngữ cảnh giữa các thread là nhỏ nên không ảnh hưởng thời gian đáp ứng các yêu cầu.
- C. Không hợp lý. Số lượng thread tạo ra quá lớn, dẫn đến thời gian đáp ứng với yêu cầu mới sẽ sụt

giảm.

☒ D. Không hợp lý. Số lượng thread tạo ra quá lớn, giữ nhiều tài nguyên trong khi nhiều kết nối Web không thực sự có nhiều tương tác.

[
]

Câu 3:

Cơ chế ánh xạ thread người dùng (user thread) vào thread lõi (kernel thread) nào sau đây trong quản lý luồng (thread) để không bị tình trạng “một thread bị block thì tất cả các thread khác cũng bị ảnh hưởng ?

A. Many-to-one và one-to-one

☒ B. One-to-one và many-to-many

C. Many-to-one và many-to-many

D. Các cơ chế trong câu hỏi khác đều được.

[CHUONG 06: MAIN MEMORY MANAGEMENT]

[
]

Câu 1:

Xem xét một chương trình C như bên dưới.

```
void *counter( void *n )
{
    int s = 0;
    int *b = (int*)n;
    for( int i = *b; i < *(b+1); i++ )
        s += i;
    return &s;
}
```

Hãy cho biết hiện thực hàm counter () có vấn đề gì ?

A. Hàm không có vấn đề gì.

☒ B. Trả về con trỏ đến biến địa phương (s) là không ổn định.

C. Trả về một con trỏ đến tổng s là không cần thiết, mặc dù vẫn dùng được.

D. Giá trị ở vùng nhớ b+1 là không được dùng.

[
]

Câu 2:

Việc gán địa chỉ (address binding) được thực hiện bởi trình biên dịch là quá trình

A. Chuyển đổi địa chỉ khả tái định vị (relocatable address) thành địa chỉ tuyệt đối.

B. Chuyển đổi địa chỉ dạng ký hiệu (symbolic address) thành địa chỉ tuyệt đối.

☒ C. Chuyển đổi địa chỉ dạng ký hiệu (symbolic address) thành địa chỉ khả tái định vị (relocatable address).

D. Gán một tập tin thực thi vào một không gian bộ nhớ tuyệt đối.

[
]

Câu 3:

Trong việc gán địa chỉ (address binding), thanh ghi tái định vị (relocation register) được dùng để ánh xạ giữa địa chỉ logic (logical address) và địa chỉ ảo (virtual address) khi nào ?

- A. Lúc biên dịch.
- ☒ B. Lúc thực thi.
- C. Lúc liên kết.
- D. Lúc nạp chương trình.

[
]

Câu 4:

Những kỹ thuật nào sau đây có thể được sử dụng để nạp những chương trình lớn hơn khả năng của bộ nhớ, mà không cần sự can thiệp của người lập trình ?

- ☒ A. Nạp động (dynamic loading)
- ☐ B. Liên kết động (dynamic linking)
- ☐ C. Liên kết tĩnh (static linking)
- ☐ D. Tất cả các phương pháp trong câu khác.

[
]

Câu 5:

Với cơ chế standard swapping, để kích hoạt việc swap một ảnh của quá trình ra khỏi bộ nhớ thì hệ điều hành phải dựa trên sự chuyển trạng thái nào của quá trình ?

- A. running → terminated
- ☒ B. running → ready
- ☐ C. ready → block
- ☐ D. ready → running

[
]

Câu 6:

Cho trước một vùng nhớ [0,999] được dành cho những quá trình người dùng. Có những phần của vùng nhớ này đã được cấp trước cho các quá trình chạy rất lâu ở các đoạn địa chỉ [200,249], [400,599], [900,949]. Có một số quá trình mới đến hệ thống, được chi tiết trong bảng sau.

Quá trình	Bộ nhớ yêu cầu	Thời điểm bắt đầu	Thời điểm kết thúc
1	200	0	10
2	250	5	25
3	150	18	40
4	100	22	35

Bộ nhớ được quản lý bởi cơ chế phân hoạch thay đổi (variable-partition). Giả thiết bộ nhớ sẽ đáp ứng yêu cầu mới dựa trên nguyên tắc phù hợp đầu tiên (first-fit). Khi đó, những lỗ hổng trong bộ nhớ tại thời điểm 24 là

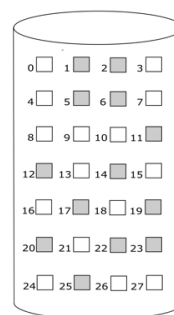
- A. [0,199], [350,399], [750,899], [950,999]
- B. [150,199], [250,399], [850,899], [950,999]
- C. [150,199], [350,299], [850,999]
- ☒ D. [150,199], [350,299], [850,899], [950,999]

[CHUONG 09: FILE IMPPLEMENTATION]

[
]

Câu 1: [L.O.2.2]

Một hệ thống tập tin được hiện thực theo cơ chế chỉ mục (indexed allocation) như hình vẽ bên. Những khối màu xám là đã được cấp phát cho các tập tin. Câu trả lời nào sau đây là đúng cho cấu trúc thư mục (directory structure) và những khối chỉ mục (index block) của hệ thống tập tin đó ?



A.

Tập tin	Khối chỉ mục
File-a.txt	11
File-b.txt	23

Khối	Nội dung
11	{5,6,1,14,25}
23	{2,22,12,17,19,20}

B.

Tập tin	Khối chỉ mục
File-a.txt	5
File-b.txt	12

C.

Khối	Nội dung
5	{1,6,11,12,14,25}
12	{2,17,19,20,22,23}

Tập tin	Khối chỉ mục
File-a.txt	5
File-b.txt	12
File-c.txt	22

Khối	Nội dung
5	{1,11,6,25}
12	{2,17,19,23}
22	{17,19,20}

D.

Tập tin	Khối chỉ mục
File-a.txt	11
File-b.txt	23
File-c.txt	22

Khối	Nội dung
11	{1,5,25}
23	{2,17,12}
22	{14,19,20}

[
]

Câu 2: [L.O.2.2]

Định nghĩa nào sau đây là đúng với một hệ thống tập tin dạng journaling?

A. Một sự kết hợp giữa cấp phát dạng liên kết (linked allocation) và cấp phát dạng chỉ mục (indexed allocation).

- B. Một ý tưởng để cải tiến dạng cấp phát liên tục (contiguous allocation) nhằm tránh phân mảnh ngoài.
 C. Một phương pháp để theo vết tất cả những khối dữ liệu chưa dùng trên đĩa.
☒ D. Một kỹ thuật để lưu lại (log) tất cả những giao dịch tập tin/thư mục giúp khả năng phục hồi.

[
]

Câu 3: [L.O.2.2]

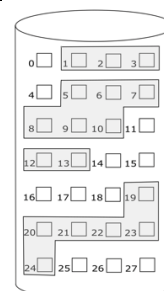
Xét một hệ thống tập tin FAT. Ý tưởng nào sau đây có thể được sử dụng để phục hồi một tập tin (những khối không bị hỏng) nếu một vài khối của tập tin đó bị hỏng (bad block)?

- A. Lưu thêm một cấu trúc thư mục để giúp phục hồi.
☒ B. Sử dụng danh sách liên kết kép (doubly linked list).
 C. Lưu một danh sách các khối cho tập tin đó trong cấu trúc thư mục.
 D. Các câu khác đều sai.

[
]

Câu 4: [L.O.2.2]

Cho trước một hình ảnh về cấp phát khối dữ liệu trên đĩa như hình vẽ bên. Cơ chế cấp phát nào là có thể phù hợp nhất đã được hiện thực cho hệ thống tập tin của đĩa ?



- A. Cấp phát liên tục (contiguous allocation).
 B. Cấp phát dạng liên kết (linked allocation).
☒ C. Cấp phát dạng chỉ mục (indexed allocation).
 D. Cấp phát dạng chỉ mục nhiều mức (multilevel indexed allocation).

[
]

Câu 5: [L.O.2.2]

Nội dung nào sau đây không phải là nhược điểm của dạng cấp phát liên tục khi hiện thực hệ thống tập tin ?

- A. Có tồn tại phân mảnh ngoài.
 B. Chi phí lớn khi dịch chuyển đầu đọc (seek) tới lui trong một tập tin.
 C. Có thể tồn tại phân mảnh trong nếu có hiện thực một cơ chế cấp phát sớm (pre-allocated mechanism).
 D. Những tập tin không thể mở rộng về mặt kích thước.

[
]

Câu 6: [L.O.2.2]

Hệ thống tập tin nào sau đây không phải thuộc dạng địa phương (local – chỉ quản lý đĩa cứng cục bộ)?

- A. UFS. ☒ B. NTFS. C. FAT. D. NFS.

[
]

Câu 7: [L.O.2.1]

Tổ chức tập tin và thư mục trên một phân hoạch đĩa được lưu trữ trong

- A. Bảng mount (mount table).
- B. Khối điều khiển tập tin (FCB).
- ☒ C. Cấu trúc thư mục.
- D. Khối điều khiển volume.

[
]

Câu 8: [L.O.2.1]

Điểm nào sau đây là lợi ích của việc dùng 2 bảng mở tập tin (open-file table) in lõi hệ điều hành khi hiện thực một hệ thống tập tin ?

- A. Để triển khai cơ chế điều khiển quyền truy xuất.
- B. Để di chuyển bảng mở tập tin cho từng tập tin (per-process open-file table) vào trong không gian của người dùng nhằm tiết kiệm bộ nhớ trong lõi.
- ☒ C. Để hỗ trợ truy xuất hiệu quả trong trường hợp có nhiều tác vụ mở cùng một tập tin.
- D. Các câu khác đều sai.