ÔN TẬP HỆ ĐIỀU HÀNH

# Câu 1: Trên một đĩa cứng có thể cài đặt tối đa bao nhiêu hệ điều hành? Giải thích lý do?

Trên một đĩa cứng MBR có thể cài đặt tối đa 4 hệ điều hành vì đĩa MBR chỉ có thể tạo ra tối đa 4 phân vùng chính. Còn đĩa cứng GPT thì tuỳ thuộc vào kích thước ổ đĩa.

# Câu 2: Hãy so sánh cấu trúc hệ thống tập tin FAT và cấu trúc hệ thống tập tin Unix

|  |  |  |
| --- | --- | --- |
|  | **FAT** | **UNIX** |
| Giống nhau | Đều dùng để quản lý hệ thống tập tin, thư mục | |
| Khác nhau | - Quản lý các cluster thay vì quản lý các sector.  - Quản lý tập tin bằng giá trị các entry của cluster.  - Nội dung gồm tên tập tin và số hiệu cluster đầu tiên của tập tin. Dùng để lưu trữ các thông tin địa chỉ các ô nhớ còn trống, các ô nhớ nào thuộc về mỗi tập tin. | - Quản lý các I-Node  - Mỗi File tương ứng với một I-Node có kích thước 64 bytes.  - Phần đánh địa chỉ nội dung ô nhớ được chia làm 2 phần: 10 phần từ đầu để lưu địa chỉ ô nhớ của tập tin, sử dụng 3 con trỏ Single, Double, Triple indirect |

# Câu 3: Hiện nay có mấy cách quản lý cấu trúc đĩa cứng, hãy trình bày các cấu trúc đó?

Chưa biết làm :))

# Câu 4:

## **a) Hãy tính thời gian chờ được xử lý trung bình của các tiến trình theo thuật toán Round Robin với Q = 4**

|  |  |  |
| --- | --- | --- |
| Tiến trình | Thời điểm vào | Thời gian xử lý |
| P1 | 0 | 10 |
| P2 | 1 | 5 |
| P3 | 4 | 3 |
| P4 | 5 | 8 |

Giải

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P1 | P4 | P2 | P1 | P4 |

0 4 8 11 15 19 20 22 26

Chú thích: [tiến trình]-[thời gian xử lý]

0 P1-10

4 P2-5, P3-3, P1-6

8 P3-3, P1-6, P4-8, P2-1

11 P1-6, P4-8, P2-1

15 P4-8, P2-1, P1-2

19 P2-1, P1-2, P4-4

20 P1-2, P4-4

22 P4-4

26 Xong

* Thời gian ở trong hệ thống:

P1 = 22 – 0 = 22

P2 = 20 – 1 = 19

P3 = 11 – 4 = 7

P4 = 26 – 5 = 21

* Tính thời gian chờ (thời gian trong hệ thống – thời gian xử lý):

P1 = 22 – 10 = 12

P2 = 19 – 5 = 14

P3 = 7 – 3 = 4

P4 = 21 – 8 = 13

**Trung bình: (12 + 14 + 4 + 13) / 4 = 10.75**

## **b) Hãy tính thời gian chờ được xử lý trung bình của các tiến trình theo thuật toán độ ưu tiên không độc quyền kết hợp Round Robin (Q = 4) đối với các tiến trình có cùng độ ưu tiên.**

**Biết rằng:**

* **Độ ưu tiên của P1, P2, P3, P4 lần lượt là 2, 3, 1, 2. (1 > 2 > 3)**
* **Sau mỗi lần xử lý độ ưu tiên của tiến trình giảm đi 1.**

**Giải**

# Câu 5: Trong bài toán “Bữa ăn tối của các nhà hiền triết” nếu bỏ câu lệnh Test(LEFT) và Test(RIGHT) trong hàm DatNia(…) thì vấn đề gì xảy ra? Hãy giải thích.

* 2 hàm Test kiểm tra xem 2 nhà hiền triết bên cạnh nhà hiền triết vừa ăn xong có đang trong trạng thái BLOCK không? Nếu có thì sẽ Up S[i] lên để 2 nhà hiền triết bên cạnh thoát khỏi trạng thái BLOCK, sau khi nhà hiền triết ăn xong đặt nĩa xuống thì 2 nhà hiền triết bên cạnh có thể cầm nĩa lên để ăn.
* Nếu bỏ 2 hàm Test thì một lúc sau tất cả các nhà hiền triết đều bị rơi vào trạng thái BLOCK.

# Câu 6: Sử dụng semaphore tạo ra hai tiến trình sau sao cho: nb < na < nb + 10

**Tiến trình 1:  
While(TRUE)**

**{**

**na = na + 1;**

**}**

**Tiến trình 2:**

**While(TRUE)**

**{**

**nb = nb + 1;**

**}**

**Giải**

Semaphore sp1 = 0;

Semaphore sp2 = 10;

Process A:

While(1)

{

DOWN(sp2);

na = na + 1;

UP(sp1);

}

Process B:

While(1)

{

DOWN(sp1);

nb = nb + 1;

UP(sp2);

}

# Câu 7: Trong một thống cài đặt bộ nhớ ảo theo kỹ thuật phân trang, giả sử một chương trình truy cập đến các trang sau: 1, 0, 4, 2, 1, 3, 1, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 3, 0, 1, 2, 4, 0, 1.

**a) Giả sử hệ thống có 5 khung trang lúc đầu đã chứa các trang 1, 0, 4, 2, 1. Có bao nhiêu lỗi trang xảy ra khi hệ thống sử dụng thuật toán thay thế trang LRU.**

**b) Giả sử hệ thống có 4 khung trang lúc đầu đã chứa các trang 1, 0, 4, 2. Có bao nhiêu lỗi trang xảy ra khi hệ thống sử dụng thuật toán thay thế trang LRU.**

**c) Giả sử hệ thống có 4 khung trang lúc đầu còn trống. Có bao nhiêu lỗi trang xảy ra khi hệ thống sử dụng thuật toán thay thế trang LRU.**

**Giải**

a) 5 khung trang lúc đầu đã chứa các trang 1, 0, 4, 2, 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 4 | 2 | 1 | 3 | 1 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 | 3 | 0 | 1 | 2 | 4 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 4 | 4 | 4 | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 2 | 2 | 2 | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Có … lỗi trang

b) 4 khung trang lúc đầu đã chứa các trang 1, 0, 4, 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 4 | 2 | 1 | 3 | 1 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 | 3 | 0 | 1 | 2 | 4 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

Có 4 lỗi trang

c) 4 khung trang lúc đầu còn trống

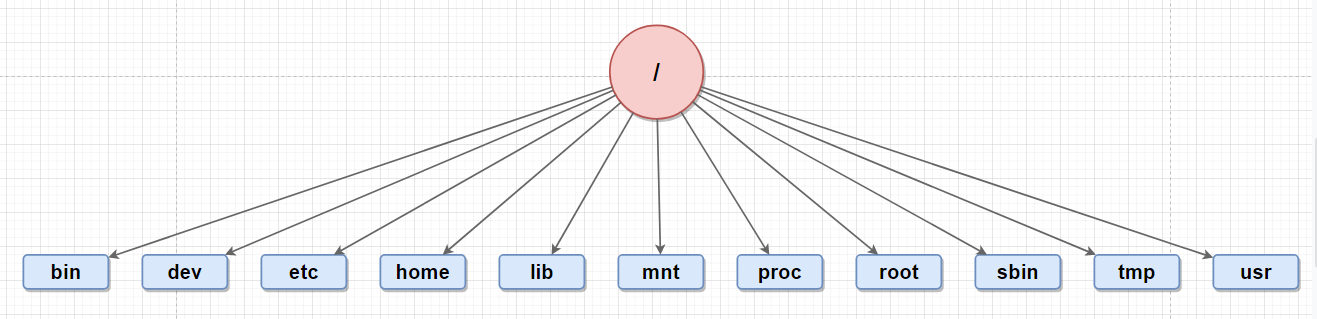
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 4 | 2 | 1 | 3 | 1 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 | 3 | 0 | 1 | 2 | 4 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 |
|  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  |  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

Có 8 lỗi trang

# Câu 8: Trình bày cấu trúc hệ thống tập tin của hệ điều hành Unix

Hệ thống tập tin Unix có cấu trúc phân cấp với thư mục cấp cao nhất được gọi là root (ký hiệu là /). Ngay dưới root là các thư mục con, hầu hết đều chứa các tập tin hệ thống.

Sơ đồ hệ thống tập tin:



* bin: thư mục chứa nhiều lệnh thực thi thường được sử dụng
* dev: chứa các tập tin cụ thể của thiết bị
* etc: chứa các tập tin cấu hình hệ thống
* home: chứa các thư mục và tập tin người dùng
* lib: chứa các tập tin thư viện
* mnt: chứa các tập tin thiết bị liên quan đến thiết bị được gắn kết
* proc: chứa các tập tin liên quan đến các quy trình hệ thống
* root: thư mục chính của người dùng root (khác với /)
* sbin: chứa các tập tin nhị phân của hệ thống
* tmp: bộ nhớ cho các tập tin tạm thời được xoá định kỳ trong hệ thống tập tin
* usr: cũng chứa các tập tin thực thi

# Câu 9: Một biến X được chia sẽ bởi 2 tiến trình cùng thực hiện đoạn chương trình sau:

**do {**

**X = X + 1;**

**if (X == 100) X = 0;**

**} while (TRUE);**

**Với giá trị X = 0 lúc ban đầu, có khi nào giá trị X vượt quá 100 không? Nếu có thì đưa ra giải pháp để đảm bảo giá trị X không bao giờ vượt quá 100.**

**Giải**

* Do X được chia sẽ bởi 2 tiến trình cùng thực hiện nên có 1 lý do nào đó khiến máy đột ngột dừng tiến trình 2 lại, và sau đó tại tiến trình 1, khi X = 99 thì tiến trình 2 bị replease đúng ngay đoạn X = X + 1, X bị cộng dồn với X = 100 sau lệnh X = X + 1 của tiến trình 1 dẫn tới X vượt quá 100.
* Giải pháp là cùng 1 lúc thì chỉ cho phép 1 trong 2 tiến trình được chạy bằng 1 biến Semaphore được khởi tạo.