***Trường Đại Học Đà Lạt***

***Khoa Công Nghệ Thông Tin***

**ĐỀ THI HỌC PHẦN: HỆ ĐIỀU HÀNH**

**LỚP CTK40, HỌC KỲ II, NĂM 2020-2021**

**THỜI GIAN: 90 PHÚT**

**Họ và tên**: Nhữ Văn Hữu

**MSSV**: 1910127

**Lớp**: CTK43

**Caâu 1: So sánh cấu trúc hệ thống tập tin FAT và cấu trúc tập tin Ext của hệ điều hành Unix?**

|  |  |  |
| --- | --- | --- |
|  | FAT | EXT của hệ điều hành Unix |
| Giống nhau | Đều dùng để quản lý 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. | Quản lý các I-Node. |
| 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 tập tin. | Mỗi tập tin tương ứng với một I-Node có kích thước 64 bytes. Phần đánh địa chỉ nội dung ô nhớ sử dụng: 10 phần tử đầu dùng để lưu địa chỉ ô nhớ của tập tin và ba con trỏ ảo (Single, double và Triple indirect). |

**Caâu 2:**

1. **Hãy tính thời gian chờ được xử lý trung bình của các tiến trình theo thuật toán Round Robin với Q=3**

|  |  |  |
| --- | --- | --- |
| Tiến trình | Thời điểm vào | Thời gian xử lý |
| P1 | 0 | 10 |
| P2 | 1 | 7 |
| P3 | 3 | 6 |
| P4 | 5 | 8 |

Thứ tự cấp phát CPU:

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

0 3 6 9 12 15 18 21 24 27 28 29 31

Thời gian chờ xử lý của P1: 0+3+9+7 = 19;

Thời gian chờ xử lý của P2: 2+9+9 = 20;

Thời gian chờ xử lý của P3: 6+9 = 15;

Thời gian chờ xử lý của P4: 7+9+2 = 18;

Thời gian chờ xử lý trung bình của các tiến trình: (19+20+15+18)/4 = 18;

1. **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=3) 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.**

Thứ tự cấp phát CPU:

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

0 3 9 17 20 23 26 29 30 31

Thời gian chờ xử lý của P1: 0+17+3+1 = 21;

Thời gian chờ xử lý của P2: 16+3+3 = 22;

Thời gian chờ xử lý của P3: 0;

Thời gian chờ xử lý của P4: 4;

Thời gian chờ xử lý trung bình của các tiến trình là: (21+22+0+4)/4 = 11.75;

**Câu 3:** **Sử dụng semaphore tạo ra hai tiến trình sau sao cho nb<na<nb+10:**

**Tiến trình 1: Tiến trình 2:**

While(TRUE) While(TRUE)

{ {

na=na+1; nb=nb+1;

} }

Typedef int Semaphore;

Semaphore Mutex = 1;

Int na = 0;

Int nb =0;

Void TienTrinh1 (void){

While (True){

Down(&Mutex);

If (na < nb+10 -1){

na = na+1;

}

Up(&Mutex);

}

}

Void TienTrinh2 (void){

While (True){

Down(&Mutex);

If (nb < na -1){

nb = nb+1;

}

Up(&Mutex);

}

}

**Câu 4:**  **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**

Trong bài toán “ bữa ăn tối của các nhà Hiền triết” nếu bỏ câu lệnh Test(LEFT) và Test(RIGHT) trong hàm DatNia(..) thì sẽ dẫn đến việc lỗi tín hiệu đánh thức làm cho tất cả các tiến trình sẽ bị block trong khi không có tiến trình nào đang được thực thi dẫn đến lỗi chương trình. Do:

* Hàm Test(int i) được tạo ra với mục đích kiểm tra xem nhà hiền triết I có đang trong trạng thái đói và hai nhà hiền triết kế bên nhà hiền triết I có đang ăn hay không, nếu nhà hiền triết I đang đói và hai nhà hiền triết bên trái và bên phải nhà hiền triết I không đang ăn thì nhà hiền triết I được ăn và sẽ gọi hàm UP(&S[i]).
* Khi bỏ qua việc gọi hàm Test(LEFT) và Test(RIGHT) khi một nhà hiền triết I đặt nĩa xuống sẽ gây lỗi: Ví dụ, trong trường hợp nhà hiền triết số 0 đang ăn thì nhà hiền triết bên phải nhà hiền triết 0 là nhà hiền triết 1 đói và muốn ăn, nhưng do nhà hiền triết 0 đang ăn nên nhà hiền triết sẽ đói và bị block (do down(S[1]) với S[1] = 0). Khi nhà hiền triết 0 ăn xong thì và đặt nĩa xuống nhưng lại bỏ qua việc hai hàm Test(LEFT) và Test(RIGHT) thì nhà nhiền triết bên phải nhà hiền triết 0 là nhà hiền triết 1 sẽ không được gọi dậy và không được đặt trạng thái và ăn.

**Tóm lại**, nếu bỏ câu lệnh Test(LEFT) và Test(RIGHT) trong hàm DatNia(..) sẽ dẫn đến không có tín hiệu đánh thức các tiến trình đang đợi để vào vùng găng (đã bị block trong khi đợi) dẫn đến lỗi chương trình.

**Câu 5:**  **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**

**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.**

Thuật toán thay thế tràn LRU:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  |  |  |  | \* |  | \* | \* | \* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Theo thuật toán thay thế trang LRU có 4 lỗi xảy ra.

# ---HEÁT----