

Câu 1 : Tiến trình nào có thể bị ảnh hưởng bởi (các) tiến trình khác mà chúng đang thực thi trong hệ thống?

Tiến trình cộng tác

Câu 2 : Khi một vài tiến trình cùng truy xuất lên một dữ liệu và kết quả tính toán của dữ liệu đó phụ thuộc vào thứ tự thực thi của các chỉ thị, chúng ta gọi đó là tình huống gì?

race condition

Câu 3 : Nếu một tiến trình đang thực thi chỉ thị nằm trong vùng nguy cơ (critical section), thì không một tiến trình nào khác được thực thi chỉ thị nằm trong vùng nguy cơ của chúng. Chúng ta gọi điều kiện như vậy là?

mutual exclusion / loại trừ tương hỗ

Câu 4 : Công cụ nào sau đây được sử dụng cho đồng bộ hóa?

Semaphore

Câu 5 : Đồng bộ tiến trình có thể tiến hành ở mức độ nào?

có thể ở mức độ phần cứng hoặc phần mềm

Câu 6 : Truy xuất đồng thời vào dữ liệu chia sẻ có thể dẫn đến

data inconsistency

Câu 7 : Đoạn mã thuộc một tiến trình mà chúng có thể thay đổi giá trị các biến dùng chung, cập nhật các bảng, ghi vào các tập tin được gọi là

critical section

Câu 8 : Ba điều kiện nào sau đây là cần để giải quyết vấn đề critical section :

i) Aging ii) Mutual Exclusion iii) Deadlock iv) Progress v) Bounded Waiting

ii, iv và v

Câu 9 : Mutual exclusion được mô tả như thế nào :

nếu một tiến trình đang thực thi đoạn mã nguy cơ (critical section), thì không tiến trình nào khác được thực thi đoạn mã nguy cơ của chúng.

Câu 10 : Bounded waiting là một đảm bảo thời gian hữu hạn đối với :

từ khi một tiến trình muốn vào critical section cho đến khi nó được vào

Câu 11 :

```
Trong giải thuật Peterson
while(1)
{
    flag [i] = true;
    turn = i;
    while ( P );
    // critical section
    flag [ i ] = false;
    // remainder section
}
```

P được mô tả như thế nào?

flag[j] = true and turn = j

Câu 12 : Kỹ thuật đồng bộ sử dụng Semaphore giải quyết được vấn đề gì mà giải thuật Peterson chưa làm được?

Busy-waiting (Tốn tài nguyên để kiểm tra theo chu kỳ).

Câu 13 : Bài toán “Sản xuất – Tiêu thụ” (Producer – Consumer) đề cập đến vấn đề gì trong Hệ điều hành?

Bảo mật thông tin khi gửi và nhận thông điệp.

Câu 14 : Câu lệnh đơn nguyên (atomic) là gì?

Câu lệnh cần thực thi liên tục bởi CPU.

Câu 15 : “Entry / Exit Section” là đoạn mã gì?

Đoạn mã hệ điều hành thêm vào trước và sau đoạn mã “Critical”.

Câu 16 : Semaphore được hiện thực như thế nào?

Biến số nguyên hoặc nhị phân, kèm theo 2 thao tác wait() và signal().

Câu 17 : Khái niệm vùng tranh chấp “Critical Section” tồn tại trong một hệ thống như thế nào?

Vùng mà nhiều tiến trình muốn thao tác tại cùng một thời điểm.

Câu 18 :

Trong đó biến mutex là biến toàn cục dùng chung (shared variable)

<i>P1:</i>	<i>P2:</i>
<i>(các lệnh khác)</i>	<i>(các lệnh khác)</i>
<i>wait(mutex);</i>	<i>wait(mutex);</i>
<i>critical section</i>	<i>critical section</i>
<i>signal(mutex);</i>	<i>signal(mutex);</i>
<i>(các lệnh khác)</i>	<i>(các lệnh khác)</i>

Phát biểu nào sau đây là đúng với hệ thống nêu trên?

Với khởi tạo mutex = 1; chỉ có 1 tiến trình được vào critical section.

Câu 19 :

Cho đoạn mã của 2 tiến trình P1 và P2 như sau:

<i>P1:</i>	<i>P2:</i>
<i>(các lệnh khác)</i>	<i>(các lệnh khác)</i>
<i>signal(mutex);</i>	<i>wait(mutex);</i>
<i>func_1();</i>	<i>func_2();</i>
<i>(các lệnh khác)</i>	<i>(các lệnh khác)</i>

Trong đó biến mutex là biến toàn cục dùng chung (Shared variable). Chọn phát biểu đúng.

Để đảm bảo hàm func_1() chạy trước func_2(), khởi tạo mutex = 0.

Câu 20 : Một tiến trình Px thực hiện thao tác signal() trên một biến số Semaphore n thì có tác dụng gì?

n++ và sau đó nếu n <= 0 thì wake_up() tiến trình đang bị blocked.

Câu 21 : Một tiến trình Px thực hiện thao tác wait() trên một biến số Semaphore n thì có tác dụng gì?

n-- và sau đó nếu n < 0 thì block() tiến trình Px.