

## ĐỀ 2011-2012

Câu 1: A, Trình bày ý tưởng của giải pháp semaphore điều độ tiến trình qua đoạn găng.

-Khởi gán bằng 1 giá trị không âm

-Üng với S(Semaphore) có 2 hàng đợi F(s) để lưu các tiến trình đang bị blocked trên S

-Chỉ có 2 thao tác Down và Up được tác động đến Semaphore S, Down giảm S xuống 1 đơn vị, Up tăng S lên 1 đơn vị

-Mỗi tiến trình trước khi vào đoạn găng gọi Down để kiểm tra và xác lập quyền vào đoạn găng. Khi tiến trình gọi Down(S) thì  $S=S-1$ , nếu  $S \geq 0$  thì tiến trình được xử lý tiếp tục và vào đoạn găng, nếu  $S < 0$  thì tiến trình phải vào hàng đợi để chờ cho đến khi  $S \geq 0$

B,

Câu 2: a, Trình bày các trạng thái của tiến trình. Tiến trình có thể rời bỏ trạng thái hiện trong trường hợp nào?

Các trạng thái của tiến trình

- + Mới tạo : tiến trình đang được tạo lập.
- + Trạng thái Ready (sẵn sàng): Tiến trình đang sẵn sàng, chờ cấp CPU để xử lý
- + Trạng thái Running (thực hiện): Tiến trình đang xử lý
- + Trạng thái Blocked (khoá): Tiến trình bị chặn không thể tiếp tục
- + Kết thúc : tiến trình hoàn tất xử lý.

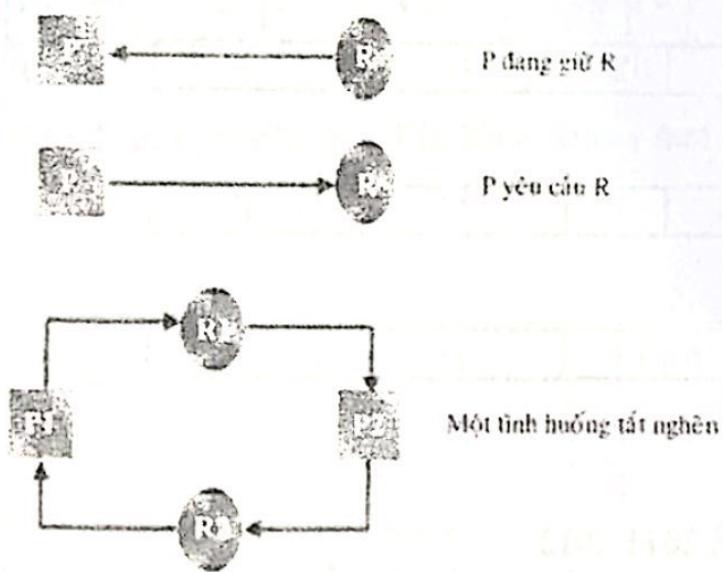
**Tiến trình có thể rời bỏ trạng thái thực hiện trong trường hợp**

Tiến trình là 1 bộ phận của chương trình đang thực hiện, là đơn vị làm việc cơ bản của hệ thống, trong hệ thống có thể tồn tại nhiều tiến trình cùng hoạt động, trong đó có cả tiến trình của HDH và tiến trình của CT người sử dụng, các tiến trình này có thể hoạt động đồng thời với nhau

Tiến trình có thể rời bỏ trạng thái thực hiện trong trường hợp hệ thống không cung cấp đầy đủ tài nguyên cho tiến trình và không duy trì đủ tài nguyên cho tiến trình trong suốt quá trình hoạt động của tiến trình

**b, Tắc nghẽn là hiện tượng:** Trong hệ thống xuất hiện một tập các tiến trình, mà mỗi tiến trình trong tập này đều chờ được cấp tài nguyên, mà tài nguyên đó đang được một tiến trình trong tập này chiếm giữ. Và sự đợi này có thể kéo dài vô hạn nếu không có sự tác động từ bên ngoài.

**Để tránh tắc nghẽn người ta sử dụng mô hình đồ thị về tài nguyên và tiến trình**



**VD:** hai tiến trình  $P_1$  và  $P_2$  sẽ rơi vào trạng thái tắc nghẽn, nếu không có sự can thiệp của hệ điều hành. Để phá bỏ tắc nghẽn này hệ điều hành có thể cho tạm dừng tiến trình  $P_1$  để thu hồi lại tài nguyên  $R_1$ , lấy  $R_1$  cấp cho tiến trình  $P_2$  để  $P_2$  hoạt động và kết thúc, sau đó thu hồi cả  $R_1$  và  $R_2$  từ tiến trình  $P_2$  để cấp cho  $P_1$  và tái kích hoạt  $P_1$  để  $P_1$  hoạt động trở lại. Như vậy sau một khoảng thời gian cả  $P_1$  và  $P_2$  đều ra khỏi tình trạng tắc nghẽn.

C,

Câu 3:a, Lỗi trang là hiện tượng hệ thống cần truy xuất đến một page của tiến trình mà trang này chưa được nạp vào bộ nhớ, hay không thuộc không gian địa chỉ của tiến trình. Ở đây ta chỉ xét lỗi trang của trường hợp: Page cần truy xuất chưa được nạp vào bộ nhớ chính.

#### Quy trình xử lý lỗi trang:

+ Hệ thống còn frame trống (a): Hệ điều hành sẽ thực hiện các bước sau:

- 1, Tìm vị trí của page cần truy xuất trên đĩa.
- 2, Nạp page vừa tìm thấy vào bộ nhớ chính.
- 3, Cập nhật lại bảng trang (PCT) tiến trình.
- 4, Tái kích hoạt tiến trình để tiến trình tiếp tục hoạt động.

+ Hệ thống không còn frame trống (b):

- 1, Tìm vị trí của page cần truy xuất trên đĩa.
- 2, Tìm một page không hoạt động hoặc không thực sự cần thiết tại thời điểm hiện tại để swap out nó ra đĩa, lấy frame trống đó để nạp page mà hệ thống vừa cần truy xuất. Page bị swap out sẽ được hệ điều hành swap in trở lại bộ nhớ tại một thời điểm thích hợp sau này.
- 3, Cập nhật PCT của tiến trình có page vừa bị swap out.
- 4, Nạp trang vừa tìm thấy ở trên (bước 1) vào frame trống ở trên (bước 2).
- 5, Cập nhật lại bảng trang (PCT) của tiến trình.
- 6, Tái kích hoạt tiến trình để tiến trình tiếp tục hoạt động.

B, LRU = 9 ; FIFO = 10