1. Kỹ thuật nào sử dụng để liên lạc trực tiếp giữa 2 tiến trình

* Đường ống (pipe)

1. ….

Với giải pháp Semaphore Dijkstra, tất cả các tiến trình được cài đặt như sau:

While( TRUE){

Down(s)

Critical-section();

Up(s)

Nonncritical-section();

}

Và e(s) khởi tạo =1,

Sử dụng để làm gì?

1. Chọn các giải pháp phần cứng để bảo vệ đoạn găng

* All (Biến cờ hiệu, cấm ngắt, kiểm tra luân phiên peterson, test&set)

1. Phương pháp nhanh nhất để chia sẻ dữ liệu giữa các tiến trình

* Vùng nhớ chia sẻ

1. Tiến trình chỉ có trạng thái running khi ở trong đooạn găng

* SAI (ở 2 trạng thái running, ready)

1. Chọn phát biểu đúng về nhóm giải pháp sleep-wakeup:

* Khi chưa đủ đk vào đoạn găng, tiến trình sang trạng thái WAITING
* Tiến trình đang sleep cần 1 tiến trình khác đánh thức

1. Giải pháp Test&Set có giải quyết triệt để bài toán độc quyền truy xuất không khi hệ thống sử dụng nhiều CPU

* Không

1. Khi
2. …
3. …
4. …
5. …
6. …
7. …
8. …
9. Chọn phát biều sai về busy-waiting:

* Tất cả các giải pháp trên đều phải thực hiện một vòng lặp để kiểm tra xem có được phép vào đoạn găng
* Nếu điều kiện chưa cho phép, tiến trình phải chờ tiếp tục trong vòng lặp kiểm tra này

1. Chọn phát biểu không chính xác về đoạn găng:

* Cả 3 phát biểu trên đều đúng

( Đoạn găng là đoạn mã lệnh dễ xảy ra tranh chấp tài nguyên khi tiến trình thực hiện nó

Đoạn găng là đoạn dữ liệu tranh chấp của chương trình

Đoạn găng là tập file độc quyền truy suất)

1. Đâu là giải pháp busy-waiting

* Biến cờ hiệu
* Peterson
* Kiểm tra luân phiên

1. Giải pháp nào có thể xày ra tình huống hai tiến trình trong đoạn găng cùng lúc:

* Biến cờ hiệu

1. Chọn phát biểu đúng về nhóm giải pháp busy-waiting:

* Các giải pháp busy-waiting làm mất thời gian của CPU vì luôn phải thực hiện kiểm tra khi chưa đủ điều kiện
* Khi chưa đủ điều kiện vào găng, tiến trình luôn ở trạng thái running