BÀI TẬP TUẦN 4 - SEMAPHORE

Huỳnh Tiến Dũng 21020007

1. Trường hợp 1: Semaphore nhị phân có biến đếm có thể có giá trị > 1

Đáp án: P0 sẽ in ra 3 lần ‘0’.
Giải thích:

- Do ban đầu S0 = 1. Nên P0 sẽ luôn chạy được wait(S0) và sau đó print ‘0’ ở vòng while đầu tiên. Sau vòng while đầu tiên thì S2 = 1 (do được signal(S2)). Ban đầu S1 = 1 nên cả P1 và P2 sẽ được chạy và mỗi process sẽ gọi signal(S0) một lần, tăng biến đếm của semaphore S0 lên 2 lần nữa. Khi đó P0 sẽ chạy được thêm 2 vòng while nữa và in ra thêm 2 số 0 nữa. Tổng cộng là 3 số 0.

2. Trường hợp 2: Semaphore nhị phân nghĩa là chỉ nhận giá trị 1 hoặc 0 (hoạt động như 1 mutex)

Đáp án: P0 sẽ in ra 2 hoặc 3 lần ‘0’ tùy vào thứ tự chạy của các process.
Giải thích:

- 2 lần ‘0’: Nếu P1 được chạy trước (do S1 = 1) và lệnh signal(S0) được chạy trước, khi mà S0 = 1 thì sau đó S0 vẫn = 1 (do đang xét trường hợp 2; nếu xét trường hợp 1 như trên thì tại đây S0 sẽ = 2). Sau đó P0 mới chạy, lệnh wait(S0); sẽ giảm S0 = 0 và P2 chạy signal(S0) và S0 = 1 và P0 có thể chạy thêm 1 vòng while nữa. Khi đó chỉ có 2 số 0 được in ra.
- 3 lần ‘0’: Nếu thứ tự sau xảy ra (khi đó các lệnh signal(S0) chạy khi S0 = 0 nên không bị mất giá trị):
    - P0 chạy wait(S0) -> P1 chạy signal(S0) -> P0 chạy wait(S0) của vòng lặp thứ 2 -> P2 chạy signal(S0) -> P0 chạy wait(S0) của vòng lặp thứ 3.
    - P0 chạy wait(S0) -> P2 chạy signal(S0) -> P0 chạy wait(S0) của vòng lặp thứ 2 -> P1 chạy signal(S0) -> P0 chạy wait(S0) của vòng lặp thứ 3.