

Câu 1:

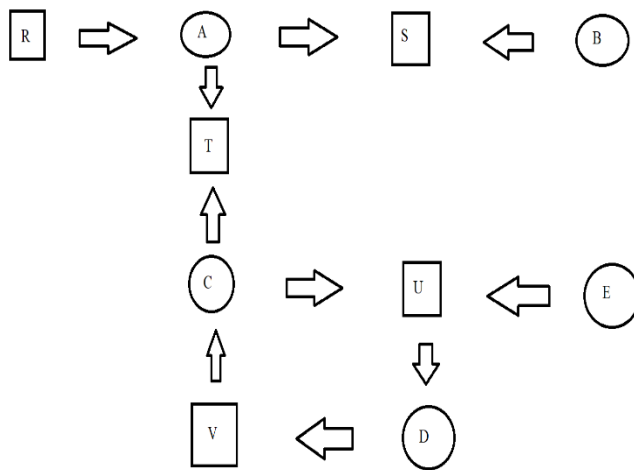
+Sự bế tắc là 1 tập hợp các tiến trình bị coi là bế tắc nếu mỗi tiến trình trong tập hợp phải chờ một sự kiện, mà sự kiện đó lại chỉ có thể do một tiến trình khác trong tập hợp tạo ra.

+Các điều kiện xảy ra bế tắc:

- 1.Điều kiện loại trừ tương hỗ : Mỗi tài nguyên hoặc được sở hữu bởi một tiến trình duy nhất, hoặc đang rảnh rỗi. (1 tài nguyên chỉ phục vụ 1 tiến trình)
- 2.Điều kiện nắm giữ và chờ : Tiến trình giữ tài nguyên trong khi chờ đợi, chẳng hạn chờ đợi để được cấp thêm tài nguyên khác
- 3.Điều kiện không giải phóng : Tài nguyên do tiến trình giữ không thể phân phối lại cho tiến trình khác trừ khi tiến trình đang giữ tự nguyện giải phóng tài nguyên
- 4.Điều kiện chờ vòng tròn: Phải có một hàng đợi vòng tròn gồm hai hoặc nhiều tiến trình, mỗi tiến trình lại đang chờ một tài nguyên được sở hữu bởi chính thành viên tiếp theo trong hàng đợi

Câu 2:

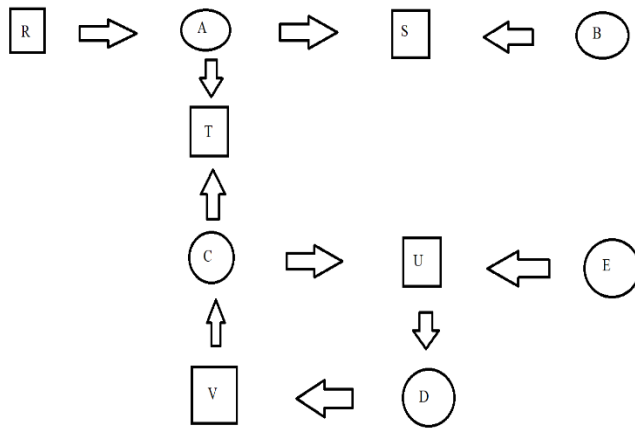
Mô hình hóa sự bế tắc



→ Hệ thống bị bế tắc tại C và D

Câu 3:

Sơ đồ phân phối tài nguyên của hệ thống



➔ Hệ thống này bị bế tắc. Hệ thống bị bế tắc tại C và D

Câu 4:

Giải thuật phát hiện vòng kín:

1. Khởi tạo L (là một danh sách rỗng), và đảm bảo rằng tất cả các mũi tên đều chưa bị đánh dấu.
2. Đặt nút hiện hành vào cuối danh sách L và kiểm tra xem liệu nút này có xuất hiện trong danh sách hai lần không. Nếu có thì tức là đã tạo thành vòng kín, giải thuật kết thúc.
3. Kiểm tra nút hiện hành xem có mũi tên nào đi ra mà lại chưa được đánh dấu không. Nếu có thì chuyển sang bước 4, nếu không có thì chuyển sang bước 5.
4. Chọn ngẫu nhiên một mũi tên (trong số các mũi tên chưa được đánh dấu) rồi tiến hành đánh dấu nó. Đi theo hướng mũi tên đó để đến một nút mới, rồi quay trở lại bước 2.
5. Ta đã đi đến điểm cuối. Xoá bỏ nó và quay trở lại nút trước đó, tiếp tục áp dụng bước 2. Nếu nút này chính là nút ban đầu, tức là không phát hiện ra vòng kín nào theo nhánh này, quá trình kiểm tra hoàn thành.

Áp dụng cho câu 3:

$L = \{ \}$

$L = \{ A \}$

$L = \{ A, S \}$ S không có mũi tên đi ra

$L = \{ A, T \}$ T không có mũi tên đi ra

$L = \{ \}$

$L = \{ B \}$

$L = \{ B, S \}$ S không có mũi tên đi ra

$L = \{ \}$

$L = \{ C \}$

$L = \{C, U\}$

$L = \{C, U, D\}$

$L = \{C, U, D, V\}$ tạo vòng kín \Rightarrow bế tắc

$L = \{C, T\}$ T không có mũi tên đi ra

Câu 5:

$L = [A, Y]$

Từ Y đi C, $L = [A, Y, C]$

Từ C đi Z, $L = [A, Y, C, Z]$

Từ Z đến D, $L = [A, Y, C, Z, D]$

Từ D đến Y, $L = [A, Y, C, Z, D, Y]$ \rightarrow phát hiện vòng lặp khép kín \Rightarrow kết thúc

Xét C đến T, $L = [A, Y, C, T]$: kết thúc vì không còn mũi tên đi ra

Câu 6:

+ **Biện pháp giải phóng tài nguyên:** Trong một số trường hợp, ta có thể tạm lấy tài nguyên của một tiến trình nào đó và giao cho tiến trình khác

Biện pháp này phụ thuộc rất nhiều vào thuộc tính của tài nguyên đó, và không dễ thực hiện

+ **Biện pháp quay trở lại:** Tiến trình đang sở hữu tài nguyên sẽ bị reset về thời điểm trước khi nó giành được tài nguyên. Tất cả công việc đã làm sau thời điểm đó sẽ bị mất.

Tài nguyên đó có thể được phân phối cho 1 tiến trình khác đang bế tắc.

Khi tiến trình được khởi động trở lại, nếu nó muốn giành lấy tài nguyên, nó sẽ phải chờ cho tới khi tài nguyên đó sẵn sàng.

Cần có cơ chế lưu trữ trạng thái của tiến trình trong quá khứ, để có thể quay trở lại.

+ **Biện pháp hủy bỏ tiến trình:** Biện pháp thô bạo nhất, nhưng cũng đơn giản nhất để giải quyết bế tắc là hủy bỏ 1 hay nhiều tiến trình.

Có thể hủy bỏ 1 tiến trình trong vòng kín. Nếu may mắn thì các tiến trình còn lại có thể tiếp tục hoạt động. Còn nếu chưa được thì có thể tiếp tục hủy bỏ các tiến trình khác cho tới khi vòng kín bị phá vỡ.