Câu 1: Bế tắc là gì?

Trình bày các điều kiện xảy ra bế tắc.

Khái niệm bế tắc : Một 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:

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)

Các tiến trình đang nắm giữ tài nguyên được cấp trước đó có thể gửi yêu cầu đòi cấp tài nguyên mới

Không thể lấy lại các tài nguyên đã được cấp trước đó cho tiến trình. Chúng phải được chính tiến trình đó giải phóng

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: Trình bày mô hình hóa sự bế tắc.

Holt đã đưa ra cách mô hình hoá bốn điều kiện trên bằng các sơ đồ nút (năm 1972)

Nút tiến trình được khoanh tròn

Nút tài nguyên được đóng khung vuông

**S**

**R**

H1 H2

Mũi tên được nối từ nút tài nguyên tới nút tiến trình nghĩa là tài nguyên đó do tiến trình đó yêu cầu, và đã được phân cho tiến trình tiến trình hiện đang sở hữu nó (hình 1)

Mũi tên được nối từ tiến trình tới tài nguyên, nghĩa là tiến trình đó hiện đang bị dừng để chờ nhận được tài nguyên đó (hình 2)

Câu 3:

a,

Diagram

Description automatically generated

b,

Hệ thống này có bế tắc gồm những tiến trình là C, D

Câu 4:

L = [∅]

Đặt E vào danh sách OPEN rồi đi tới U

L = [E,U]

L = [E,U,D]

L = [E,U,D,V]

L = [E,U,D,V,C]

L = [E,U,D,V,C,U] → xuất hiện 2 U → vòng lặp khép kín → END

⇒ Tiến trình bế tắc ở C,D

Câu 5:

L = [∅]

Đặt X vào danh sách OPEN rồi đi tới A

L = [X,A] (A sở hữu X)

L = [X,A,Y] (A xin cấp Y)

L = [X,A,Y,C] (C sơ hữu Y)

L = [X,A,Y,C,Z] (C xin cấp Z)

L = [X,A,Y,C,Z,D] (D sở hữu Z)

L = [X,A,Y,C,Z,D,Y] (D xin cấp Y)

🡪 Xuất h iện 2 Y → Vòng lặp khép kín → END

⇒ Tiến trình bế tắc ở C,D

Câu 6: Trình bày các biện pháp giải quyết bế tắc sau: giải phóng tài nguyên, quay trở lại và hủy bỏ tiến trình

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.

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 các công việc đã làm sau thời điểm đó sẽ bị mất.

Tài nguyên đó có thể được phân phối cho một 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.

Hủy bỏ tiến trình:

Biện pháp thô bạo nhất, nhưng cũng đơn giản nhất để giải quyết bế tắc là huỷ bỏ một hoặc nhiều tiến trình.

Có thể huỷ bỏ một 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 huỷ bỏ các tiến trình khác cho tới khi vòng kín bị phá vỡ.

Các vấn đề Phòng tránh bế tắc, Ngăn chặn bế tắc... sinh viên tự nghiên cứu trong cuốn Modern Operating Systems.