Tên: Trần Quốc Anh  
MSSV: 51900007  
Nhóm: 11

*GIẢI QUYẾT BÀI TOÁN “TRỢ GIẢNG ĐANG NGỦ”*

\_Các biến số: mutex TA entry

* Trường hợp 1: giữa các SV với nhau

mutex = 1 TA = 1 entry = 0

Sinh viên

wait(mutex)  
*//kiểm tra trạng thái của trợ giảng*  
entry++  
*//lượt vào được cộng vào*if (entry==1)  
 wait(TA)  
signal(mutex);

*//trợ giảng giải đáp thắc mắc cho sinh viên*

wait(mutex)  
entry--  
if (entry==0)  
 signal(TA)  
signal(mutex)

Sinh viên

wait(mutex)  
*//kiểm tra trạng thái của trợ giảng*  
entry++  
*//lượt vào được cộng vào*if (entry==1)  
 wait(TA)  
signal(mutex);

*//trợ giảng giải đáp thắc mắc cho sinh viên*

wait(mutex)  
entry--  
if (entry==0)  
 signal(TA)  
signal(mutex)

\_Các biến số: entry(lượt vào) = 0, TA = 1 và mutex = 1.

\_Đầu tiên, SV kiểm tra trạng thái của TA, nếu TA đang ngủ (TA không tư vấn cho sinh viên nào) thì bước vào phòng (đi vào vùng tranh chấp) bằng lệnh wait(mutex), biến mutex lúc này = 0. Sau đó, SV kiểm tra mình là người thứ mấy.  
\_Nếu chỉ có 1 SV duy nhất thì, SV kiểm tra trạng thái của TA, nếu TA đang ngủ thì đánh thức bằng lệnh wait(TA), lúc này biến TA = 0 là SV đang nói chuyện với TA.

\_Lúc này sinh viên hạ cờ báo hiệu đã vào được vùng tranh chấp signal(mutex), TA giải đáp thắc mắc cho SV và mutex lúc này bằng 1.

\_Nếu có một sinh viên khác đi vào, SV đó phải kiểm tra trạng thái bên trong phòng wait(mutex), lúc này mutex = 0, lượt vào được cộng thêm, lúc này entry = 2.

\_Kiểm tra thấy entry không bằng 1 nên sẽ không thể giao tiếp được với TA, lúc này SV đó ra khỏi vùng tranh chấp signal(mutex) lúc này mutex = 1.

\_SV sau khi được giải đáp xong bắt đầu bước ra khỏi phòng, lại đi vào vùng tranh chấp, và báo hiệu wait(mutex) (mutex = 0), sau khi bước ra khỏi phòng thì entry trừ đi 1. Sau đó kiểm tra xem có còn SV nào không, nếu không thì trợ giảng sẽ ngủ (signal(mutex)), nhưng vẫn còn SV.

\_Lúc này, TA giải đáp thắc mắc cho SV kia, sau khi xong thì lượt vào lại bị trừ entry lúc này bằng 0, kiểm tra thấy entry = 0, thì trợ giảng sẽ ngủ vì không giao tiếp với bất kì sinh viên nào nữa.

* Trường hợp 2: giữa SV với TA

mutex = 1 TA = 1 entry = 0

Trợ giảng

wait(TA)  
*//kiểm tra trạng thái của trợ giảng  
//trợ giảng tiếp sinh viên*

signal(TA)

Sinh viên

wait(mutex)  
*//kiểm tra trạng thái của trợ giảng*  
entry++  
*//lượt vào được cộng vào*if (entry==1)  
 wait(TA)  
signal(mutex);

*//trợ giảng giải đáp thắc mắc cho sinh viên*

wait(mutex)  
entry--  
if (entry==0)  
 signal(TA)  
signal(mutex)

\_Các biến số ban đầu: TA=1, mutex = 1, entry = 0

\_Kiểm tra tình trạng, trợ giảng đang tiếp một sinh viên, bằng lệnh wait(TA), lúc này TA = 0.

\_Lúc này có một sinh viên đi vào vùng tranh chấp (đi vào phòng), wait(mutex) thì mutex sẽ bằng 0 (báo hiệu sinh viên đi vào), entry đồng thời cũng được tăng lên là 1.

\_Kiểm tra entry bằng 1 thì sẽ kiểm tra tình trạng của trợ giảng (wait(TA)), lúc này TA không phải là số dương và là -1, nên SV ấy phải đợi (đóng băng), không được tiếp chuyện với trợ giảng.

\_Sau khi trợ giảng vấn đáp với SV, thì hạ cờ bằng lệnh signal(TA) thì TA sẽ thành 0, lúc này sẽ giải thoát đóng băng cho các SV đến sau.

\_Sau khi SV hỏi trợ giảng, SV ra khỏi vùng tranh chấp, và SV tiếp theo vào, nếu không còn sinh viên nào (entry = 0), trợ giảng sẽ rơi vào trạng thái ngủ.

* \_Hai trường hợp trên của bài toán, đều gây “bận chờ đợi”, nhưng cả hai trường hợp đã đáp ứng dc pp process và

\_Đối với trợ giảng và sinh viên vi phạm tính loại trừ lẫn nhau (vì khi TA = 0 đã đánh thức toàn bộ các tiến trình SV đi vào vùng tranh chấp).

\_Đối với trường hợp giữa các sinh viên sẽ gây tranh chấp rất nhiều, gây tắc nghẽn hệ thống.

\_Nếu đưa hệ thống trở lại bình thường thì khi hoạt động cũng có thể gây tắc nghẽn.

\_Vì vậy không thể tránh việc tranh chấp với cả việc “bận chờ đợi”.  
\_

Ảnh có chứa vẽ

Mô tả được tạo tự động