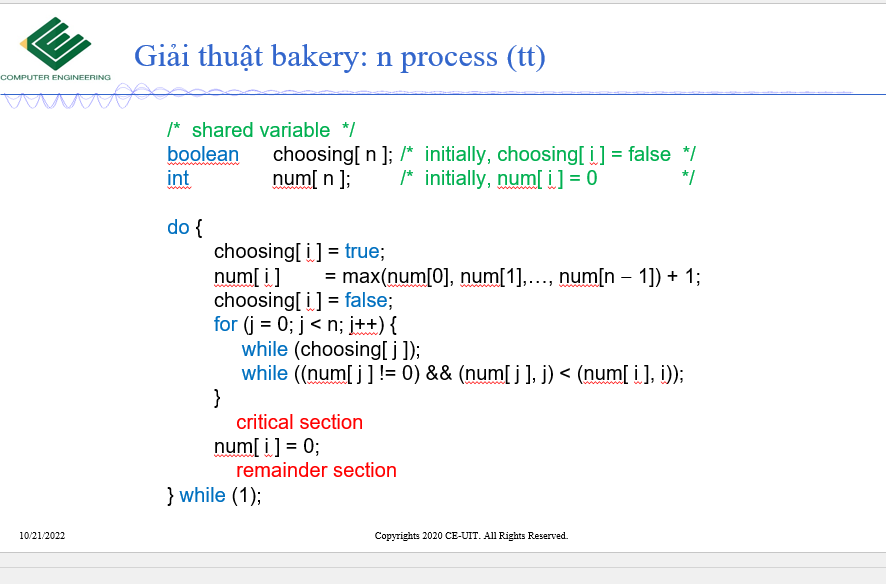
|  |
| --- |
| Họ tên: Lê Thị Thu Hiền  MSSV: 21522059  Môn : IT007 |

* Vấn đề Critical Section: phải bảo đảm sự loại trừ tương hỗ (mutual exclusion, mutex), tức là khi một process đang thực thi trong vùng tranh chấp, không có process nào khác đồng thời thực thi các lệnh trong vùng tranh chấp.
* Progress: Một tiến trình tạm dừng bên ngoài vùng tranh chấp không được ngăn cản các tiến trình khác vào vùng tranh chấp
* Chờ đợi giới hạn (Bounded waiting): Mỗi process chỉ phải chờ để được vào vùng tranh chấp trong một khoảng thời gian có hạn định nào đó. Không xảy ra tình trạng đói tài nguyên (starvation).

**Câu 1: Giải thích thuật toán có thỏa 3 điều kiện tranh chấp .**

Thuật toán thỏa mãn 3 điều kiện tranh chấp.



mỗi khi có một process nào vào thực thi trong vùng tranh chấp thì nó sẽ được gán cho một giá trị với giá trị đó luôn là nhỏ nhất so với giá trị của các process khác. Những process khác có giá trị được gán lớn hơn và xếp trong hàng đợi (thỏa điều kiện 1) .

Sau khi hoàn thành thì process sẽ được gán giá trị là 0 và đi ra khỏi vùng tranh chấp ( kết thúc hoặc ra hàng đợi) (thỏa mãn điều kiện 2) và sẽ không gây cản trở cho các process khác vào .

Việc đánh số cho các process dùng để xác định Process tới sau sẽ nhận number lớn hơn 1 so với Max của các number có sẵn trong hàng đợi,và vì process nào có number nhỏ nhất sẽ được vào vùng trah chấp trước, mỗi process đều sẽ chỉ có thời gian đợi vào vùng tranh chấp nhất định.(thỏa mãn điều kiện 3)

**Câu 2: Giải thích thuật toán có thỏa 3 điều kiện tranh chấp .**

Thuật toán thỏa mãn 3 điều kiện tranh chấp.

Graphical user interface, application

Description automatically generated

Cấu trúc dữ liệu dùng chung (khởi tạo là false)

bool waiting[ n ];

bool lock;

Mutual exclusion: Pi chỉ có thể vào vùng tranh chấp nếu và chỉ nếu hoặc waiting[ i ] = false, hoặc key = false

key = false chỉ khi TestAndSet (hay Swap) được thực thi

Process đầu tiên thực thi TestAndSet mới có key == false; các process khác đều phải đợi

waiting[ i ] = false chỉ khi process khác rời khỏi CS

Chỉ có một waiting[ i ] có giá trị false

Progress: hàm trả về false làm cho tiến trình bỏ qua vòng while, đồng thời gán lock=true để khóa các tiến trình khác không vào vùng tranh chấp. Nếu tiến trình nào đó mjoons vào vùng tranh chấp thì do lock=true nên vòng while bắt tiến trình đó phải chờ ( do hàm TestAndSet trả về true )

Bounded waiting: khi process i rời khỏi CS, process j = (i + 1) / n sẽ được set waiting[j] thành false và được tiến vào CS, nghĩa là mỗi process đều đợi tới lượt theo vòng tròn, tức là sẽ có một thời gian giới hạn nhất định cho việc chờ