

★ Operazione di assegnamento gestita da due processi (istruzioni non atomiche)

Prenotazione dello stesso posto (aereo) da due punti diversi

Esempi

## Problema principale: INTERFERENZA. Si manifesta quando due processi (thread) accedono ad una risorsa condivisa

Soluzione: rendere atomiche (indivisibili) un insieme di operazioni raggruppandole in una parte di codice ad alto livello chiamata SEZIONE CRITICA, in modo da garantire il comportamento corretto del sistema indipendentemente dalla velocità relativa dei processi

Def. SEZIONE CRITICA ○ La soluzione viene realizzata attraverso appositi algoritmi che garantiscono

Mutua esclusione  
Progresso  
Attesa limitata

Es. Algoritmo di Petersen

CLASSICO ESEMPIO DI PROBLEMATICITA': PRODUTTORE CONSUMATORE: P è un processo che produce degli oggetti (scrive valori in un buffer). C, è un processo che consuma degli oggetti (legge dei valori dal B. e li elabora)

PROBLEMA: Si deve garantire mutua esclusione

SOLUZIONE: si utilizza una chiave (LOCK). Chi usa la risorsa deve prima prendere la chiave, finito di usare la risorsa deve riporre la chiave.

La soluzione si può implementare sia in maniera HW che SW

A livello SW la mutua esclusione è un problema di più basso livello rispetto agli altri, generalmente vengono messe a disposizione del programmatore delle funzioni per indicare che opportune risorse devono essere utilizzate in regime di mutua esclusione

PROBLEMA: Si deve garantire sincronizzazione: Se P è veloce e C è lento, P non deve sovrascrivere il valore. Viceversa se P è lento e C è veloce, C non deve prelevare due volte lo stesso valore

Soluzione: si usa un flag. Quando P scrive un nuovo dato alza il flag (indica un valore nuovo). Quando C legge abbassa il flag (indica che il B. è vuoto)