

Passaggio tra sistemi concorrenti e sistemi distribuiti

Proprietà.

- **MUTEX** = al più un processo può entrare in sezione critica alla volta
- **NO DEADLOCK** = un processo tra N rimane bloccato e non accede alla sezione critica mentre esiste P che accede alla sezione critica = un insieme N prova a entrare in sez critica e di questi un sottoinsieme rimane in trying mentre altri riescono ad entrare ed uscire.

Ogni Processo esegue questo ciclo/codice:

```
<No in sez critica>                = CODICE UTENTE  
<trying section> // prova ad entrare = PROTOCOLLO  
<sezione critica> //utilizzo della risorsa condivisa (dato/device fisico/file  
etc.)  
<exit protocol>  
goto -1
```

Ma bisogna garantire le due proprietà sopra elencate in questo ciclo.
Il PROTOCOLLO è adattato al tipo di sistema di calcolo e si basa sulle ASSUNZIONI.

I MODELLO di SISTEMA:

Cpu - Memoria - Scheduler

P1 - ...- Pn = I Processi

I processi tramite lo scheduler ottengono la cpu secondo una certa politica.

Assunzioni:

Riguardo l'Accesso alla memoria:

1. A1 = i processi leggono e scrivono variabili condivise (var in memoria che può essere letta e scritta da ogni processo)
2. A2 = lettura e scrittura sono azioni atomiche (quando faccio read/write io processo la porto a termine)

Tipo di Scheduler

3. A3 = Scheduler è fair (non esiste processo che non entra in azione per tempo infinita, entro un certo tempo finito ogni processo entra nella

cpu e progredisce , in un esecuzione infinita tutti i processi eseguono infinite operazioni).

Teorema di Dijkstra.

1. **NCS**
2. $y[i] = \text{TRUE}$
3. $x[i] = \text{FALSE}$
4. while $k \neq i$ DO
5. if not $y[k]$ then $k = i$
6. $x[i] = \text{TRUE}$
7. for $j = 1$ to n DO
8. if $i \neq j$ and $x[j]$ then GOTO 3
9. $y[i] = x[i] = \text{FALSE}$

i è il mio identificatore

Fino a riga 7 è la mia trying section

Riga 8 = sezione critica

Riga9 = exit protoc