CAP. 7 : DEADLOCK

Il problema del deadlock

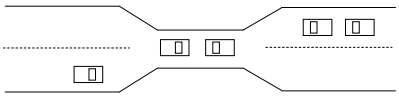
Esiste un ***insieme di processi*** bloccati, ognuno:

- ***in possesso di una o piu' risorse***

e

- ***in attesa di una risorsa posseduta da un altro processo dell'insieme***

L'attraversamento di un ponte…



- Il traffico sul ponte e' a ***senso unico alternato***
- Ogni ***tratto del ponte*** (es. 100 mt.) puo' essere visto come una ***risorsa***
- In caso di deadlock uno o piu' veicoli possono essere ***ricondotti indietro*** per sbloccare la situazione
- Comunque e' possibile che si verifichi ***starvation***

Di nuovo sui processi…

Definizione formale :

Un insieme di processi e' in deadlock se ogni processo nell'insieme sta attendendo un evento che puo' essere causato solo da un altro processo in quell'insieme (di solito il rilascio di una risorsa)

… un esempio…

Un sistema di calcolo ha ***due unita' a disco*** :

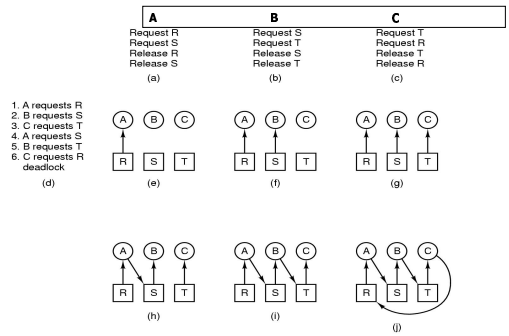
P1 ***mantiene il possesso*** di un disco ed ***ha bisogno*** dell'altro
P2 fa lo stesso

I semafori *A* e *B*, inizializzati a 1, sono di ***mutua esclusione*** sulle due unita'

P1	P2
wait (A)	*wait(B)*
wait (B)	*wait(A)*

P1 : wait(A)
P2 : wait(B) ⟵ ***Deadlock!***

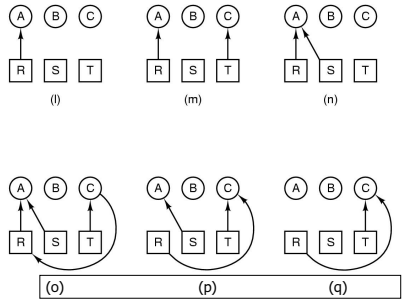
… uno piu' complicato…



… e come poteva essere evitato!

1. A requests R
2. C requests T
3. A requests S
4. C requests R
5. A releases R
6. A releases S

no deadlock



Un modello del sistema di calcolo

- *m* ***tipi di risorse*** $R_1, R_2, \ldots, R_m$
  Cicli di CPU, spazio di memoria, dispositivi di I/O, etc.
- Ogni tipo di risorsa R_i ha W_i ***istanze***
- *n* ***processi*** $P_1, P_2, \ldots, P_n$ - ogni processo, per utilizzare una risorsa, esegue :
  - *request*
  - use
  - *release*

  system calls

  Accodamento sulla risorsa occupata

Caratterizzazione del deadlock…

Un deadlock puo' avvenire ***solo se*** le seguenti condizioni valgono ***simultaneamente*** :

1. **Mutua esclusione**
2. **Possesso e attesa**
3. **Assenza di preemption**
4. **Attesa circolare**

⇒ ***Condizioni necessarie !!!***

… e in dettaglio

A. Mutua esclusione: solo un processo alla volta puo' utilizzare una (istanza di una) risorsa

B. Possesso e attesa: un processo che possiede almeno una risorsa sta in attesa di acquisire risorse aggiuntive possedute da altri processi

C. Assenza di preemption: una risorsa puo' essere rilasciata da un processo che la possiede solo in modo volontario, dopo che il processo ha completato il suo compito

D. Attesa circolare: esiste un insieme {P0, P1, …, Pn} di processi in attesa tale che P0 sta attendendo una risorsa posseduta da P1, P1 sta attendendo una risorsa posseduta da P2, …, Pn–1 sta attendendo una risorsa posseduta da Pn, e Pn sta attendendo una risorsa posseduta da P0

Rappresentazione del modello: grafo di allocazione risorse…

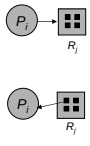
Un grafo e' costituito da un insieme di ***vertici V*** e un insieme di ***archi E***

In un ***grafo di allocazione risorse*** (struttura dinamica):

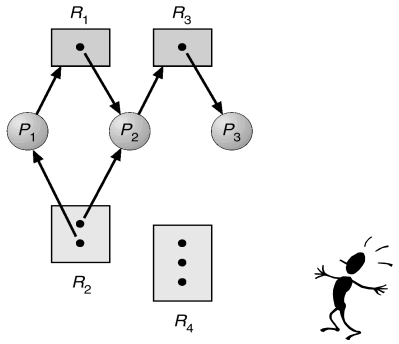
- ***V*** e' partizionato in due sottinsiemi di vertici:
  - $P = \{P_1, P_2, \ldots, P_n\}$, l'insieme dei ***processi***
  - $R = \{R_1, R_2, \ldots, R_m\}$, l'insieme dei ***tipi di risorse***
- ***E*** e' partizionato in due sottinsiemi di archi:
  - Archi di ***richiesta*** $P_1 \rightarrow R_j$
  - Archi di ***assegnamento*** $R_j \rightarrow P_i$

… la notazione…

- Processo
- Tipo di risorsa con istanze
- P_j richiede una istanza di tipo R_j
- P_j possiede una istanza di tipo R_j



… ed un esempio

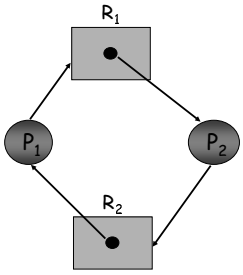


Condizioni per deadlock in un grafo di allocazione risorse

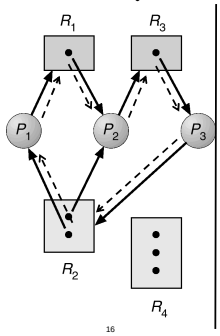
La presenza di un ***ciclo*** in un grafo di allocazione risorse con ***un'unica istanza*** per ogni tipo di risorsa ***implica*** deadlock

La presenza di un ***ciclo*** in un grafo di allocazione risorse con ***multiple istanze*** per ogni tipo di risorsa ***puo' implicare*** deadlock

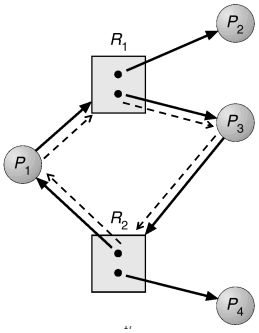
Grafo di allocazione risorse con deadlock (unica istanza)



Grafo di allocazione risorse con deadlock (multiple istanze)…



… e grafo con ciclo ma senza deadlock (multiple istanze)



Come trattare il deadlock…

- ***Garantire*** che il sistema ***non entrera' mai*** in uno stato di deadlock (prevenire o evitare)
- Permettere che il sistema entri in uno stato di deadlock e quando cio' accade ***ripristinare una situazione normale*** (rilevare e ripristinare)

… o come far finta di niente!

Ignorare il problema supponendo che il sistema non entri mai in uno stato di deadlock (es. UNIX)

IDEA DI BASE: ***l'overhead*** introdotto dal trattamento del deadlock e' ***troppo alto*** e il ***deadlock troppo raro*** per valere la pena di considerarlo

CONTROINDICAZIONE: un ***piccolo insieme di processi*** in deadlock puo' facilmente ***contagiare*** altri e portare tutto il sistema al ***blocco totale***, che in certi casi puo' essere ***piu' dannoso dell'overhead*** di gestione

Operazioni di trattamento del deadlock

Esaminiamo ora nel dettaglio le operazioni di trattamento del deadlock :

1. ***Prevenire***
   oppure
2. ***Evitare***
   oppure
3. ***Rilevare***
4. ***Ripristinare***

1. Prevenire

Tecniche per ***prevenire*** che (anche solo) ***una delle 4 condizioni necessarie accada***

A. ***Mutua esclusione*** – si puo' prevenire soltanto su risorse condivisibili quali file in sola lettura

B. ***Possesso e attesa*** – *Due possibilita':*
   - ✓ un processo puo' richiedere risorse solo se non possiede altre
   - ✓ un processo richiede prima di iniziare tutte le risorse di cui ha bisogno

e' possibile starvation!

C. Assenza di preemption –

1. se un processo che possiede qualche risorsa richiede un'altra risorsa che non puo' immediatamente essergli allocata, allora tutte le risorse correntemente possedute sono rilasciate
2. tali risorse vengono aggiunte alla lista di risorse per le quali il processo sta in attesa
3. il processo ripartira' solo quando potra' riacquisire le vecchie e le nuove risorse

- OPPURE -

La preemption viene applicata solo quando una delle risorse possedute dal processo viene richiesta da un altro processo

Di facile applicazione solo a risorse "leggere" che possono essere allocate/deallocate facilmente

D. Attesa circolare – si ordinano le risorse su numeri naturali (eventualmente per tipologie)

Dischi ad accesso veloce : 1,
Dischi ad accesso lento : 2 , etc.

- si formulano le richieste solo in ordine crescente

OPPURE

- si puo' chiedere una risorsa di numero inferiore solo se si rilascia tutto cio' che si possiede di numero superiore

Come sarebbe cambiato questo grafo se una delle 4 "prevenzioni" fosse stata applicata ???