



**detection:**  
permette deadlock e rileva aggiorna Holt

casi e cause deadlock:  

- \* 1risorsa1classe->ciclo
- \*+risorse1classe->non completamente riducibile
- \*1risorsa1proc->esiste knot

**recovery:** risolve manualmente/automatica:  

- \*terminazione totale/parziale (costoso, inconsistenza)
- \*preemption
- \*checkpoint-rollback

detection and recovery

impedisce il deadlock

prevention: elimina condizioni deadlock con spooling, allocazione:  

- \*totale (memoria)
- risorse non note subito
- \*gerarchica (tempo)
- risorse priorità, allocare:  
 ↑priorità ok  
 ↓priorità rilascia priorità ≥

avoidance: ritarda assegnamenti risorse causano deadlock

definizioni

prevention/avoidance

nella versione multi valuta tutti i parametri sono condizionati (k)

alg. banchiere

ignora il problema

Ostrich algorithm

processi bloccati in reciproca attesa della risorsa dell'altro

Ricorda

\*mutua esclusione risorse seriali  
 \*risorse non preemptive  
 \*richieste bloccanti (hold and wait)  
 \*attesa circolare  
 necessarie e sufficienti

Condizioni

Classi di risorse

Azioni processo

Risorse preemptable

Deadlock

Gestione Risorse e Deadlock

Grafo (G) di Holt

Cos'è

Riducibilità

Knot

istanze(risorse) stessa classe sono equivalenti (soddisfano stessa richiesta)  
molteplicità= $\text{size}(\text{classe})$

processo richiede una istanza (non specifica) di classe (specificata)

assegnazione risorse:  

- \*statica, dalla creazione all'eliminazione
- \*dinamica, all'occorrenza processo già avviato richiede e rilascia

risorsa:  

- \*seriale, 1proc alla volta (W)
- \* non seriali, +proc alla volta (R)

richiesta:  

- \*singola, 1risorsa-1classe
- \*multipla, 1/+risorse-1/+classi
- \*bloccante, proc. sospeso fino +(rivalutazione a ogni rilascio)
- \*non bloccante, proc. continua se assegnazione fallisce

funzione gestione sottrae (poi restituisce) risorsa a processo (si sospende) prima del dovuto

condizione: stato risorsa indipendente utilizzo o salvabile e ripristinabile

Risorse non preemptable...

bipartito (2tipi di nodi) risorse/classi e processi

diretto (archi direzionali):

- \*risorsa->processo assegnata
- \*processo->risorsa richiesta insoddisfacibile

nodi stesso partito non si connettono

esistenza nodo proc. con solo archi entranti

riduzione: riassegnare sue risorse ad altri proc.

idea: prima o poi termina

$\{\max(M) \mid \forall n \in M, R(n)=M\}$

$R(n)=\text{insieme di raggiungibilità (nodi raggiungibili da } n\text{), } M \subseteq G$