DEADLOCK

Un insieme di processi è in deadlock quando ogni processo è in attesa di un evento che può essere causato da un processo dello stesso insieme.

Condizioni necessarie:
1) Mutua esclusione: Almeno una risorsa deve essere non condivisibile
2) Hold and Wait: Deve esistere un processo che detiene una risorsa e che attende di acquisirne un'altra, detenuta da un altro
3) No preemption: Le risorse non possono essere rilasciate se non "volontariamente" dal processo che le usa
4) Attesa circolare: Deve esistere un insieme di processi che attendono ciclicamente il liberarsi di una risorsa, evono essere vere contemporaneamente Se una non si verifica, non si ha deadlock

Prevenzione statica: Impedire che si verifichi una delle 4 condizioni che devono essere vere contemporaneamente perché si verifichi un deadlock:
1)Mutua esclusione: è irrinunciabile per alcuni tipi di risorse 2)Hold and wait : Un processo alloca all'inizio tutte le risorse che deve
utilizzare, Un processo può ottenere una risorsa solo se non ne ha altre
– Problemi: Basso utilizzo delle risorse, possibilità di starvation
3)No preemption: • Un processo che richiede una risorsa non disponibile
deve cedere tutte le altre risorse che detiene, In alternativa, può cedere risorse che detiene su richiesta di un altro processo
– Problemi: Fattibile solo per risorse il cui stato può essere
facilmente "ristabilito" (CPU, registri, semafori, file)
4)Attesa circolare: Assegnare una priorità (ordinamento globale) ad ogni
risorse, Un processo può richiedere risorse solo in ordine
crescente di priorità

Prevenzione dinamica: Prevenzione in base alle richieste, Analisi dinamica del grafo delle risorse per evitare situazioni cicliche
• Requisito: Conoscenza del caso peggiore (bisogna conoscere il massimo numero di istanze di una risorsa richieste per processo

Stato di una risorsa calcolato come: numero di istanze allocate numero di istanze disponibili
Stato sicuro (safe):se esiste una sequenza safe, ovvero se usando le risorse disponibili, può allocare risorse ad ogni processo, in qualche ordine, in modo che ciascun di essi possa terminare la sua esecuzione

Algoritmo con RAG: Funziona solo se ho un'istanza per ogni risorsa Il RAG viene aumentato con archi di reclamo, All'inizio, ogni processo deve dire quali risorse
vorrebbe usare durante la sua esecuzione, Una richiesta viene soddisfatta se l'allocazione della risorsa non crea un ciclo nel RAG, Serve un algoritmo per la rilevazione cicli, Complessità O(n2) (dove n = n° di processi)

Algortimo del banchiere: Il banchiere non deve mai distribuire tutto il denaro che ha in cassa perché altrimenti non potrebbe più soddisfare successivi clienti, Ogni processo dichiara la sua massima richiesta, Ogni volta che un processo richiede una risorsa, si determina se soddisfarla ci lascia in uno stato safe
• Costituito da:
– Algoritmo di allocazione
– Algoritmo di verifica dello stato

Rilevazione e ripristino con RAG:Funziona solo con una risorsa per tipo, Analizzare periodicamente il RAG, verificare se esistono deadlock (detection) ed iniziare il ripristino (recovery) .Vantaggi = Conoscenza anticipata delle richieste non necessaria, Utilizzo migliore. Svantaggio = Costo del recovery

Algoritmo di rilevazione: Basato sull'esplorazione di ogni possibile sequenza di allocazione per i processi che non hanno ancora terminato, Se la sequenza va a buon fine (safe), non c'è deadlock, Strutture dati simili ad algoritmo del banchiere:

MEMORIA

Il binding di dati e istruzioni a indirizzi di memoria: può
avvenire in tre momenti distinti
– Al tempo di compilazione (compile time) :Se è noto a priori in quale parte della memoria risiederà il processo, è possibile generare codice assoluto ,Se la locazione di partenza cambia, necessario (ri)compilare
– Al tempo di caricamento (load time) : Necessario generare codice rilocabile (=riposizionabile), Indirizzi relativi, Se cambia l'indirizzo di riferimento, devo (ri)caricare
– Al tempo di esecuzione (run time): Binding posticipato se il processo può essere spostato durante l'esecuzione in posizioni diverse della memoria Richiesto supporto hardware (per efficienza)

Spazi di indirizzamento: Lo spazio di indirizzamento logico è legato a uno
spazio di indirizzamento fisico. indirizzo logico: generato dalla CPU, detto anche indirizzo virtuale indirizzo fisico: visto dalla memoria

Memory Management Unit (MMU): Dispositivo hardware che mappa indirizzi virtuali
(logici) indirizzi fisici – Il valore del registro di rilocazione è aggiunto ad ogni
indirizzo generato da un processo, e inviato alla
memoria

Allocazione contigua: Processi allocati in memoria in posizioni
contigue all'interno di una partizione ,La memoria è suddivisa in partizioni:
– Partizioni fisse
– Partizioni variabili