**CAPITOLO 7: STALLO DEI PROCESSI**

In un ambiente con multiprogrammazione più processi possono competere per ottenere un numero finito di risorse. Se una risorsa non è correntemente disponibile, il processo richiedente passa allo stato d’attesa. Se le risorse richieste sono trattenute da altri processi, a loro volta nello stato d’attesa, il processo potrebbe non cambiare più il suo stato. Situazioni di questo tipo sono chiamate di stallo (**deadlock).**

**7.1** **MODELLO DEL SISTEMA**

Un sistema è composto da un numero finito di risorse da distribuire tra i vari processi in competizione. Le risorse sono suddivise in tipi differenti, ciascuno formato da un certo numero di istanze identiche. Cicli di CPU, spazio di memoria, file e dispositivi di I/O sono tutti tipi di risorsa. Se un sistema ha due unità d’elaborazione, tale tipo di risorsa ha due istanze. Se un processo richiede un’istanza relativa a un tipo di risorsa, l’assegnazione di qualsiasi istanza di quel tipo può soddisfare la richiesta. Se ciò non si verifica significa che le istanze non sono identiche e le classi di risorse non sono state definite correttamente.

Prima di adoperare una risorsa, un sistema deve richiederla e dopo averla usata deve rilasciarla. Un processo può richiedere tutte le risorse necessarie per eseguire il compito assegnatogli, anche se il numero delle risorse richieste non può superare il numero totale delle risorse presenti nel sistema. Nelle ordinarie condizioni di funzionamento un processo per servirsi di una risorsa deve rispettare la seguente sequenza:

* **Richiesta.** Se la richiesta non si può soddisfare immediatamente, il processo deve attendere
* **Uso.** Il processo può adoperare la risorsa
* **Rilascio.** Una volta terminato l’uso della risorsa, deve rilasciarla

La richiesta e il rilascio avvengono attraverso chiamate di sistema quali **request()** e **release()**.

La richiesta e il rilascio di altre risorse si possono eseguire per mezzo delle operazioni **wait( )** e **signal( )** su semafori. Quindi ogni volta che si usa una risorsa, il SO controlla che il processo utente me abbia fatto già richiesta e che questa gli sia stata assegnata. Una tabella del sistema memorizza tutti gli stati di ogni risorsa e se questa viene assegnata, indica il processo relativo.

Se un processo richiede una risorsa già assegnata un altro processo, il processo richiedente può essere accodato agli altri processi che attendono tale risorsa.

Un gruppo di processi entra in stallo quando tutti i processi del gruppo attendono un evento che può essere causato solo da un altro processo che si trova nello stesso stato di attesa. Le risorse possono essere sia fisiche (CPU, stampanti..) , sia logiche (file, semafori).

**7.2** **CARATTERIZZAZIONE DELLE SITUAZIONI DI STALLO**

In una situazione di stallo, i processi non terminano mai l’esecuzione e le risorse del sistema vengono bloccate impedendo l’esecuzione di altri processi.

**7.2.1** **Condizioni necessarie**

So può avere una situazione di stallo solo se si verificano contemporaneamente queste 4 condizioni:

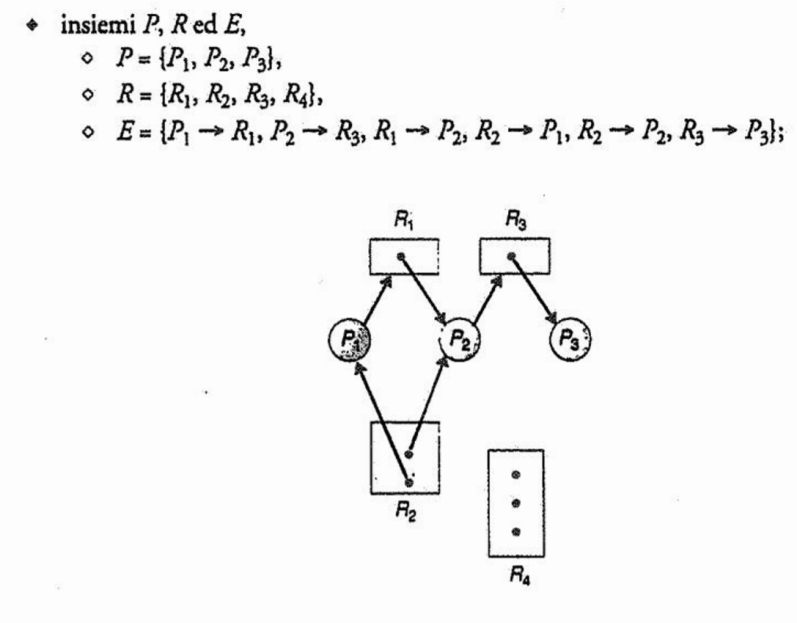
* **Mutua esclusione.** Almeno una risorsa deve essere non condivisibile, vale a dire che è utilizzabile da un solo processo alla volta. Se un altro processo richiede tale risorsa si deve ritardare il processo finché la risorsa non diventi disponibile.
* **Possesso e attesa.** Un processo in possesso di almeno una risorsa attende di acquisire risorse già in possesso di altri processi.
* **Impossibilità di prelazione.** Non esiste un diritto di prelazione sulle risorse vale a dire che una risorsa può essere rilasciata volontariamente solo dal processo che la sta utilizzando.
* **Attesa circolare.** Deve esistere un insieme di processi (P0, P1,…Pn-1) tale che P0 attende una risorsa da P1, P1 una di P2, …, Pn-1 una risorsa di P0.

Occorre sottolineare che tutte e quattro le condizioni devono essere vere, queste condizioni non sono completamente indipendenti; tuttavia è utile considerare separatamente ciascuna condizione.

**7.2.2 Grafo di assegnazione delle risorse**

Le situazioni di stallo si possono descrivere con maggior precisione avvalendosi di una rappresentazione detta **grafo di assegnazione delle risorse**.

Si tratta di un insieme di vertici V e un insieme di archi E. un arco diretto dal processo Pi alla risorsa Rj significa che il processo ha richiesto la risorsa. Un arco dalla risorsa al processo significa che la risorsa è assegnata al processo. Un arco dal processo alla risorsa è detto arco di richiesta, un arco dalla risorsa al processo è detto arco di assegnazione. Quando il processo Pi richiede un’istanza del tipo di risorsa *Rj,* si inserisce un arco di richiesta nel grafo di assegnazione delle risorse. Se questa richiesta può essere esaudita, si trasforma *immediatamente* l’arco di richiesta in un arco di assegnazione, che al rilascio della risorsa viene cancellato.



**Istanze delle risorse:**

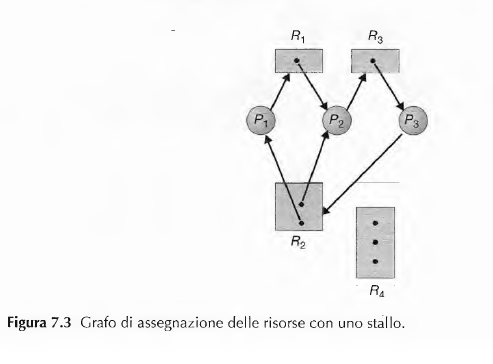
* un’istanza del tipo di risorsa R1
* due istanze del tipo di risorsa R2
* un’istanza del tipo di risorsa R3
* tre istanze del tipo di risorsa R4

**Stati dei processi**

* il processo *P1* possiede un’istanza del tipo di risorsa *R2* e attende un’istanza del tipo di risorsa *R1*
* il processo *P2* possiede un’istanza dei tipi di risorsa *R1* ed *R2* e attende un’istanza del tipo di risorsa *R3*
* il processo *P3* possiede un’istanza del tipo di risorsa *R3*

Data la definizione di grafo di assegnazione delle risorse, e facile mostrare che, se il grafo non contiene cicli, nessun processo del sistema subisce uno stallo; se il grafo contiene un ciclo, può sopraggiungere uno stallo. Se ciascun tipo di risorsa ha esattamente un’istanza, l'esistenza di un ciclo implica la presenza di uno stallo; se il ciclo riguarda solo un insieme di tipi di risorsa, ciascuno del quale ha solo un’istanza, si è verificato uno stallo. Se ogni tipo di risorsa ha più istanze, un ciclo non implica necessariamente uno stallo. In questo caso l'esistenza di un ciclo nel grafo è una condizione necessaria ma non sufficiente per l'esistenza di uno stallo.

**esempio)** I processi *P1, P2 e P3* sono in stallo: il processo *P2* attende la risorsa *R*3, posseduta dal processo *P3;* il processo *P*3, invece, attende che il processo *P1* o *P2* rilasci la risorsa *R*2; inoltre, il processo *P1* attende che il processo *P2* rilasci la risorsa *R1.*



Si consideri ora il grafo di assegnazione delle risorse della Figura 7.4. Anche in questo esempio

c’è un ciclo:

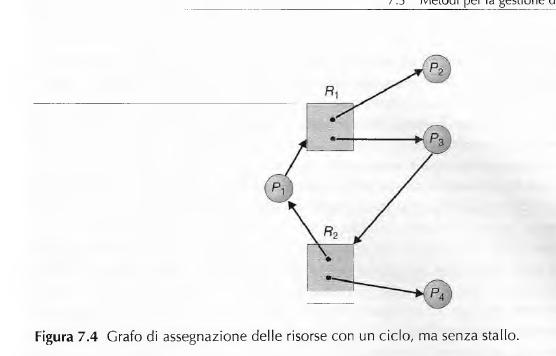
***P1***—> *R1* —> ***P3*** —>*R2* —> ***P1***

In questo caso, però, non si ha alcuno stallo: il processo *P4* può rilasciare la propria istanza del tipo di risorsa *R2 ,* che si può assegnare al processo P3, rompendo il ciclo.

Per concludere, l’assenza di cicli nel grafo di assegnazione delle risorse implica *l'assenza*

di situazioni di stallo nel sistema. Viceversa, la presenza di un ciclo non è sufficiente a implicare

la presenza di uno stallo nel sistema.



**7.3 METODI PER LA GESTIONE DELLE SITUAZIONI DI STALLO**

La situazione di stallo si può affrontare in 3 modi:

* Si può usare un protocollo per prevenire le situazioni di stallo
* Si può permettere al SO di entrare in stallo individuarlo e ripristinarlo
* Si può ignorare che il problema di stallo fingendo che non esistono.

Quest’ultima soluzione è quella adottata dalla maggior parte dei sistemi operativi (compresi Unix e Windows).

Per assicurare che non si verifichi mai uno stallo, il sistema può servirsi di metodi di prevenzione o di metodi per evitare tale situazione. **Prevenire le situazioni di stallo** significa far uso di metodi atti ad assicurare che non si verifichi almeno una delle condizioni necessarie (7.2).Per **evitare le situazioni di stallo** Occorre che il sistema operativo abbia in anticipo informazioni aggiuntive riguardanti le risorse che un processo richiederà e userà durante le sue attività. Con queste informazioni aggiuntive si può decidere sulla richiesta di risorse da parte di un processo si può soddisfare o si debba invece sospendere. In tale processo di decisione il sistema tiene conto delle risorse correntemente disponibili, di quelle correntemente assegnato a ciascun processo, e delle future richieste e futuri rilasci di un processo. Se un sistema non impiega né un algoritmo per evitare né un algoritmo per prevenire gli stalli, il SO può Servirsi di un algoritmo che esamini lo stato del sistema al fine di stabilire se si è verificato uno stallo e in tal caso ricorrere a un secondo algoritmo per il ripristino del sistema.

Se un sistema non garantisce che la situazione di stallo non possono mai verificarsi e non fornisce alcun meccanismo per la loro individuazione e per il ripristino del sistema, situazioni di stallo possono avvenire senza che ci sia la possibilità di capire cosa è successo. In questo caso la presenza di situazioni di stallo non rilevate può causare un degrado delle prestazioni del sistema fino al blocco totale del sistema che dovrà essere riavviato manualmente.