



Gestione dei Processi

Informatica B

Che cosa è un processo per il SO?

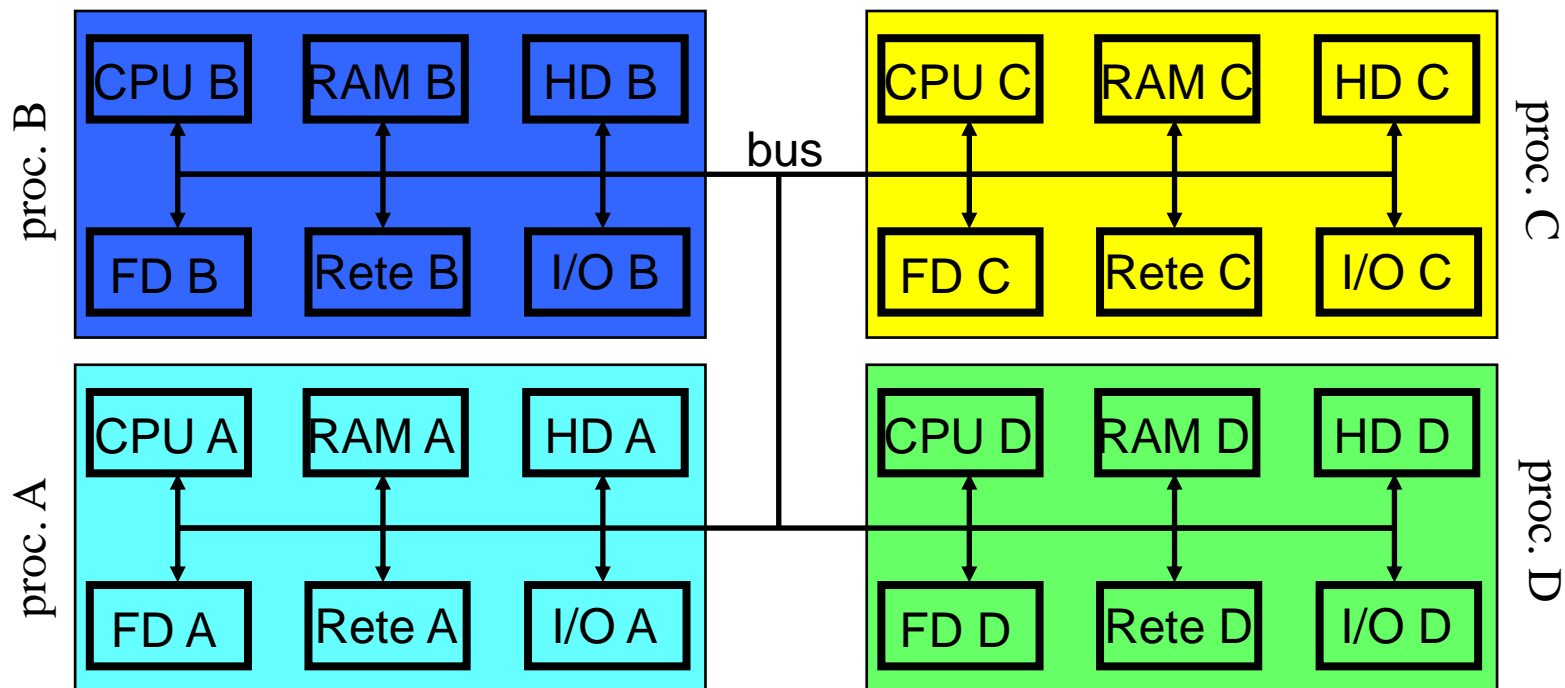
- ❑ Processo \neq programma !
- ❑ Rappresenta un'istanza di un programma composta da:
 - ▶ codice eseguibile (il programma stesso)
 - ▶ dati del programma
 - ▶ informazioni relative al suo funzionamento (stato)
- ❑ Lo stesso programma può essere associato a più processi:
 - ▶ Un programma può essere scomposto in varie parti e ognuna di esse può essere associata ad un diverso processo
 - ▶ Lo stesso programma può essere associato a diversi processi quando diverse copie del medesimo processo sono mandate in esecuzione

Lo stato di un processo

- ❑ Lo stato del processo può essere distinto fra stato interno e stato esterno.
- ❑ Lo stato interno indica:
 - ▶ la prossima istruzione del programma che deve essere eseguita;
 - ▶ i valori delle variabili e dei registri utilizzati dal processo.
- ❑ Lo stato esterno indica se il processo è:
 - ▶ in attesa di un evento, ad es. la lettura da disco o l'inserimento di dati da tastiera;
 - ▶ in esecuzione;
 - ▶ pronto per l'esecuzione, e quindi attende di accedere all'utilizzo della CPU.

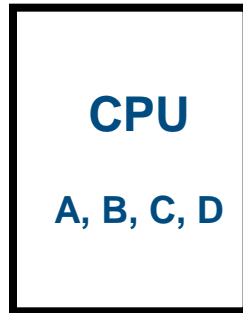
Il sistema operativo e le macchine virtuali

- ❑ Il sistema operativo esegue più processi contemporaneamente
- ❑ Rende quindi visibile ad ogni processo una macchina virtuale ad esso interamente dedicata e quindi con risorse proprie

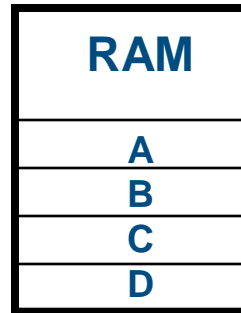


Il sistema operativo e la macchina reale

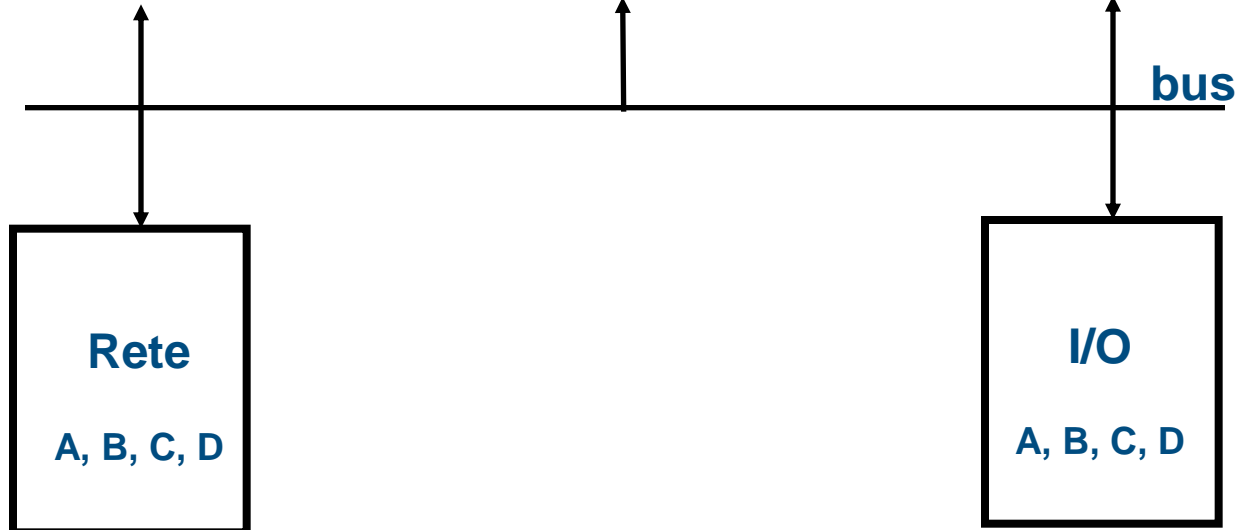
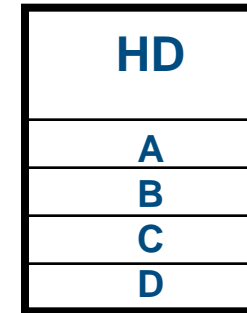
utilizzo a rotazione



suddivisione in blocchi



suddivisione in blocchi



utilizzo a rotazione

utilizzo a rotazione

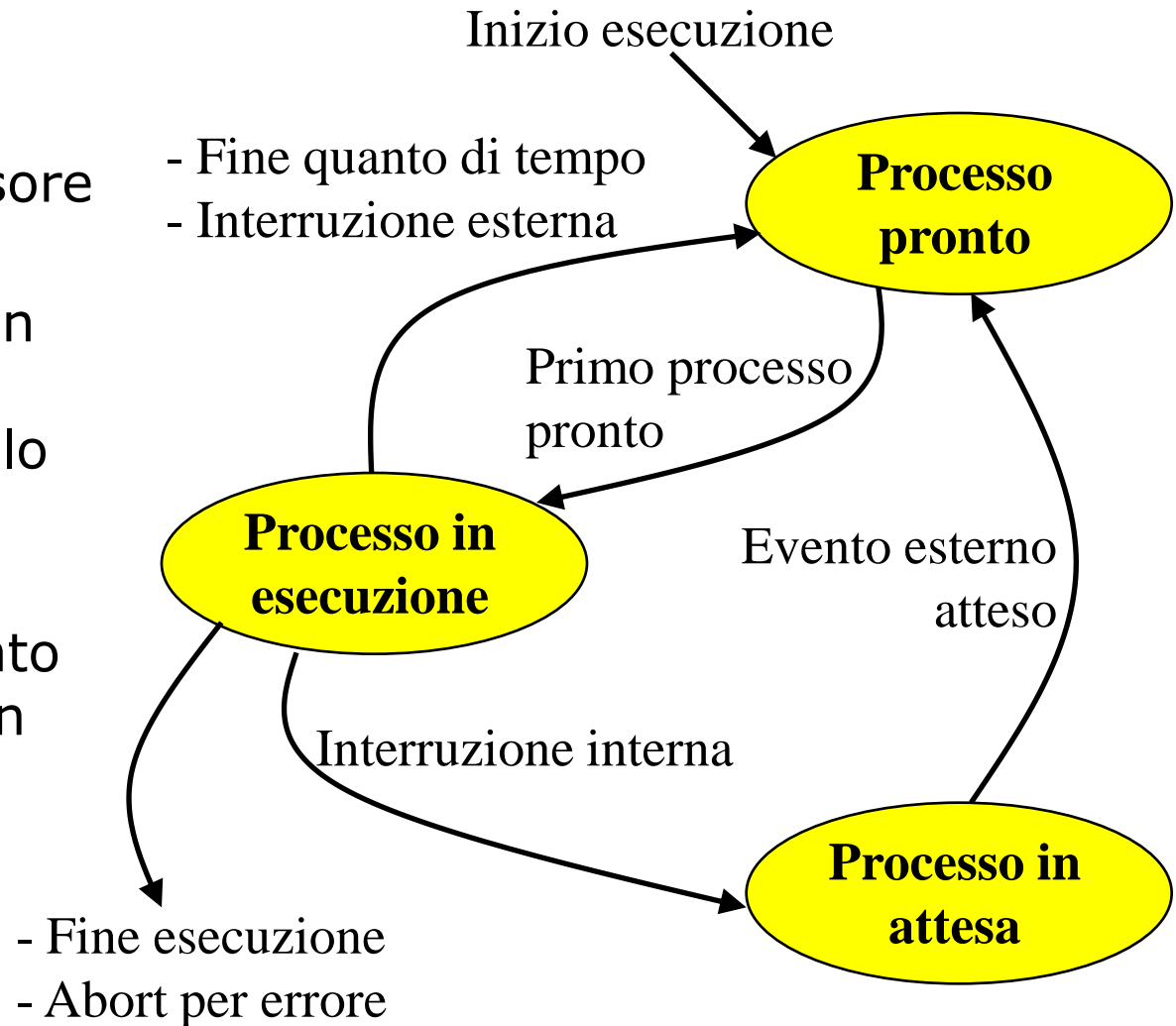
- ❑ Anche il sistema operativo è implementato tramite processi;
- ❑ Il sistema operativo è garante che i conflitti tra i processi siano controllati e gestiti correttamente;
- ❑ Il sistema operativo viene eseguito in modalità privilegiata (kernel mode o supervisor), così da poter controllare gli altri processi eseguiti in modalità user.

Chiamate al supervisor

- ❑ I processi utente per eseguire operazioni privilegiate (accesso a file, accesso ad altre risorse, operazioni di I/O, ecc.) invocano il supervisor tramite chiamate di sistema
- ❑ Perché usare la modalità privilegiata (supervisor)?
 - ▶ Le operazioni di I/O sono operazioni riservate:
 - un processo A non deve poter andare a scrivere messaggi su un terminale non associato allo stesso processo A;
 - un processo A non deve poter leggere caratteri immessi da un terminale non associato allo stesso processo A.
 - ▶ Un processo non deve poter sconfinare al di fuori del proprio spazio di memoria:
 - per non accedere allo spazio di memoria associato ad un altro processo, modificando codice e dati di quest'ultimo;
 - per non occupare tutta la memoria disponibile nel sistema, bloccandolo e rendendolo così inutilizzabile da altri processi.
 - ▶ La condivisione di risorse (dischi, CPU, ecc.) deve essere tale da cautelare i dati di ogni utente;

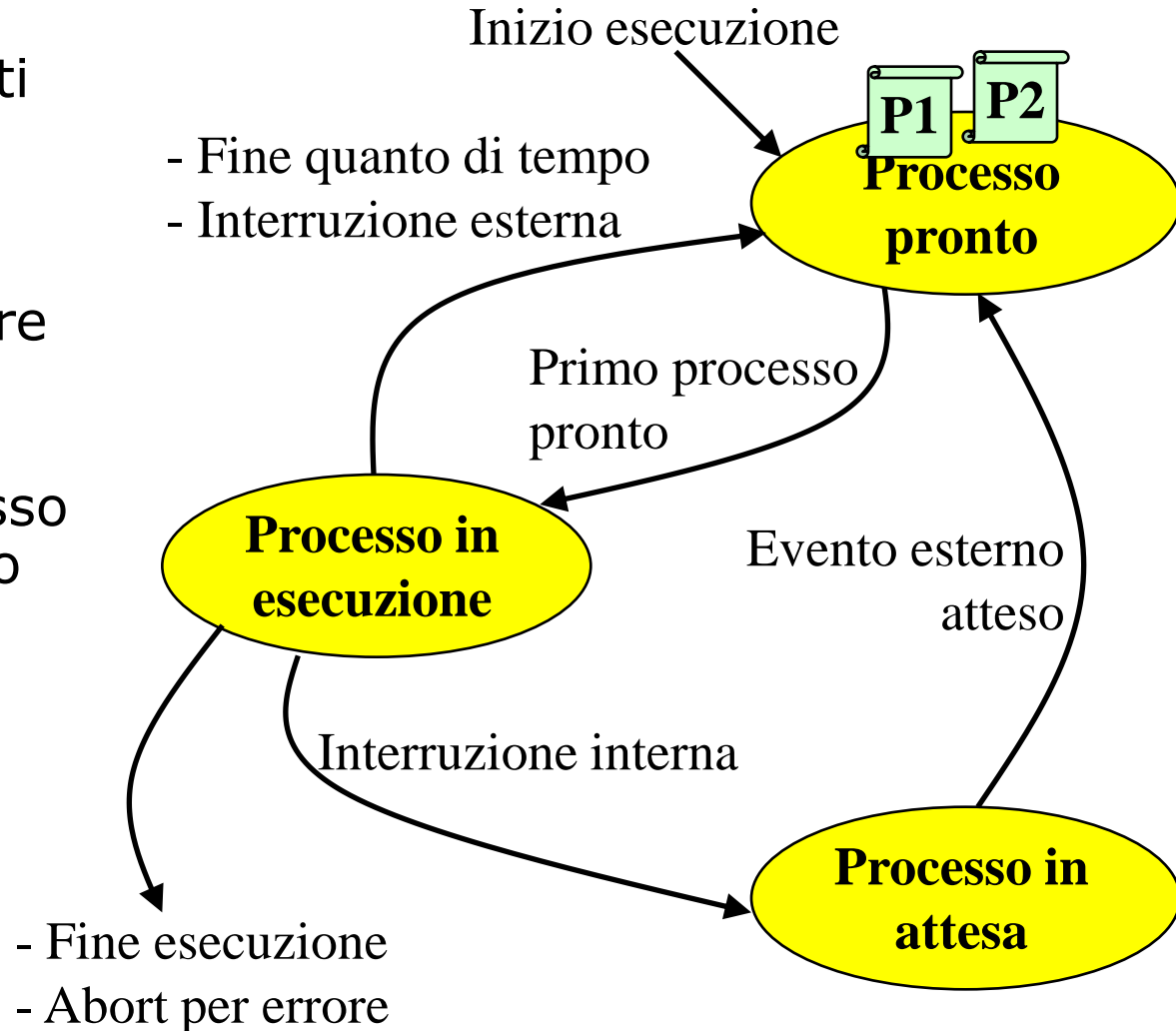
Stato di un processo (1)

- ❑ *In esecuzione*: assegnato al processore ed eseguito da esso
- ❑ *Pronto*: può andare in esecuzione, se il gestore dei processi lo decide
- ❑ *In attesa*: attende il verificarsi di un evento esterno per andare in stato di *pronto*



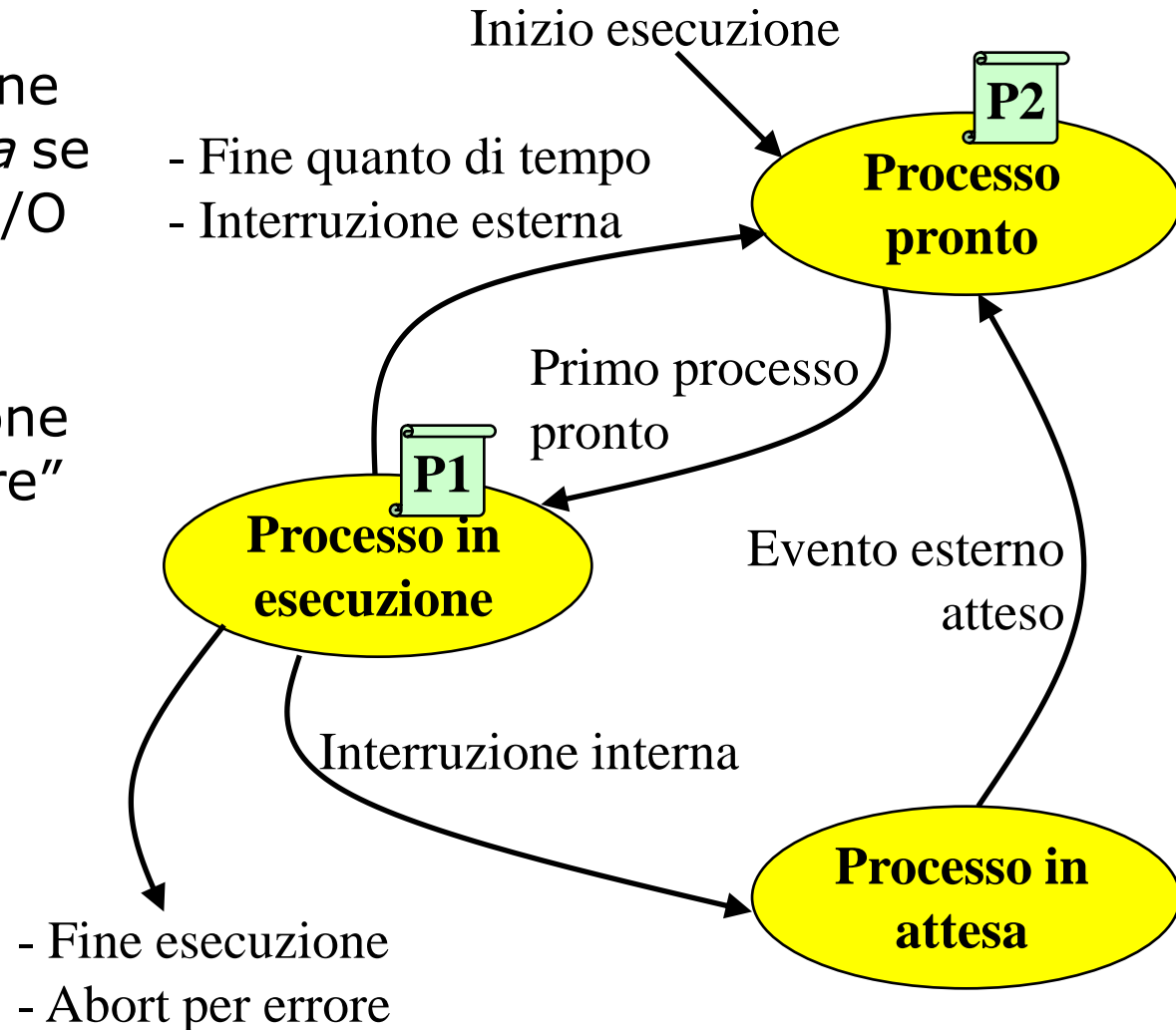
Stato di un processo (2)

- ❑ I processi appena creati sono messi in stato di *pronto*
- ❑ Il nucleo decide quale processo pronto mettere in stato di esecuzione
- ❑ Il nucleo assegna il processore a un processo per un quanto di tempo
 - ▶ Coda dei processi pronti
 - ▶ Round-robin
 - ▶ Priorità dei processi



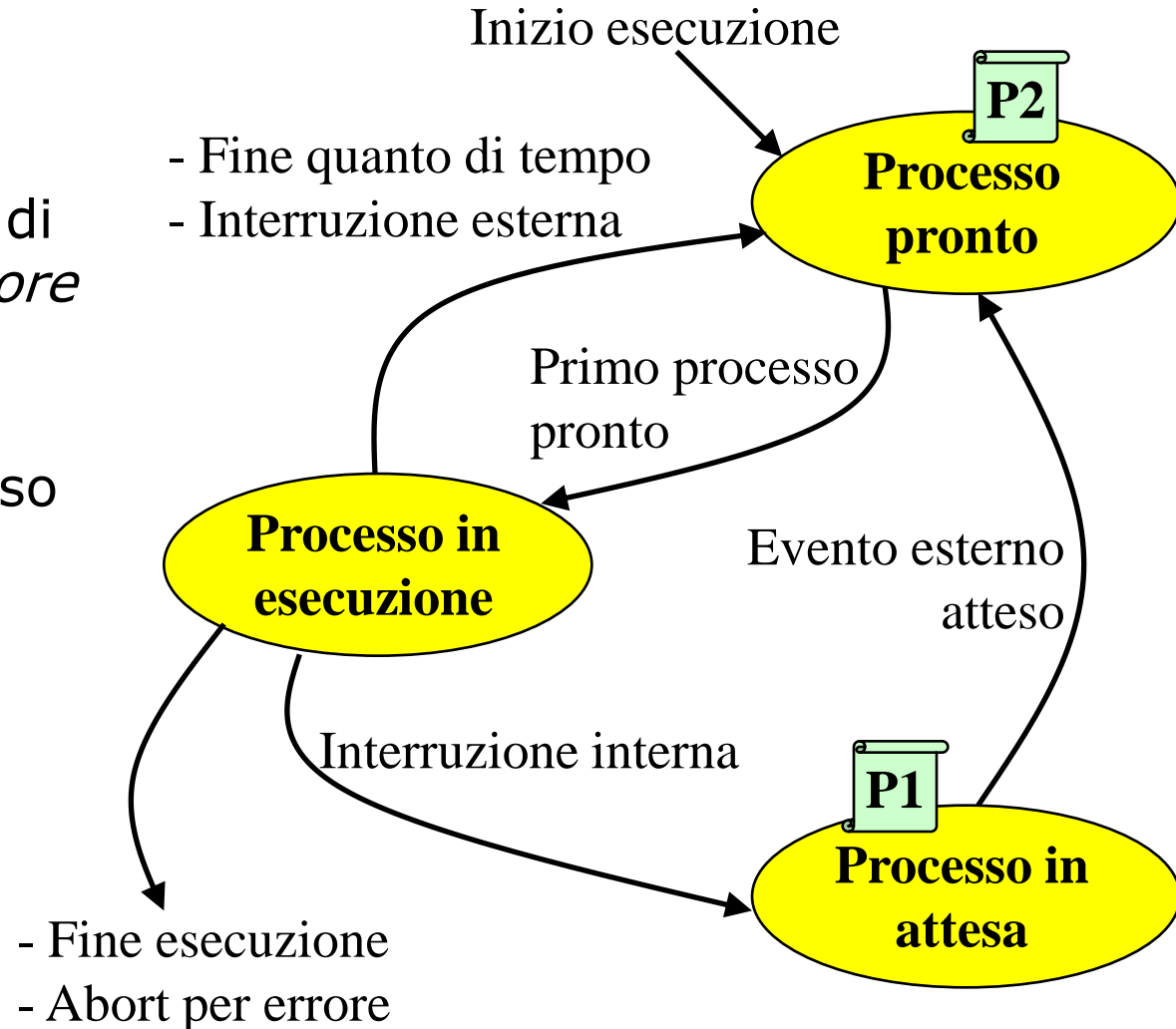
Stato di un processo (3)

- Il processo in esecuzione passa in stato di *attesa* se richiede operazioni di I/O (*interruzione interna*)
- Corrisponde alla esecuzione dell'istruzione "chiamata a supervisore" (*SuperVisor Call, SVC*)



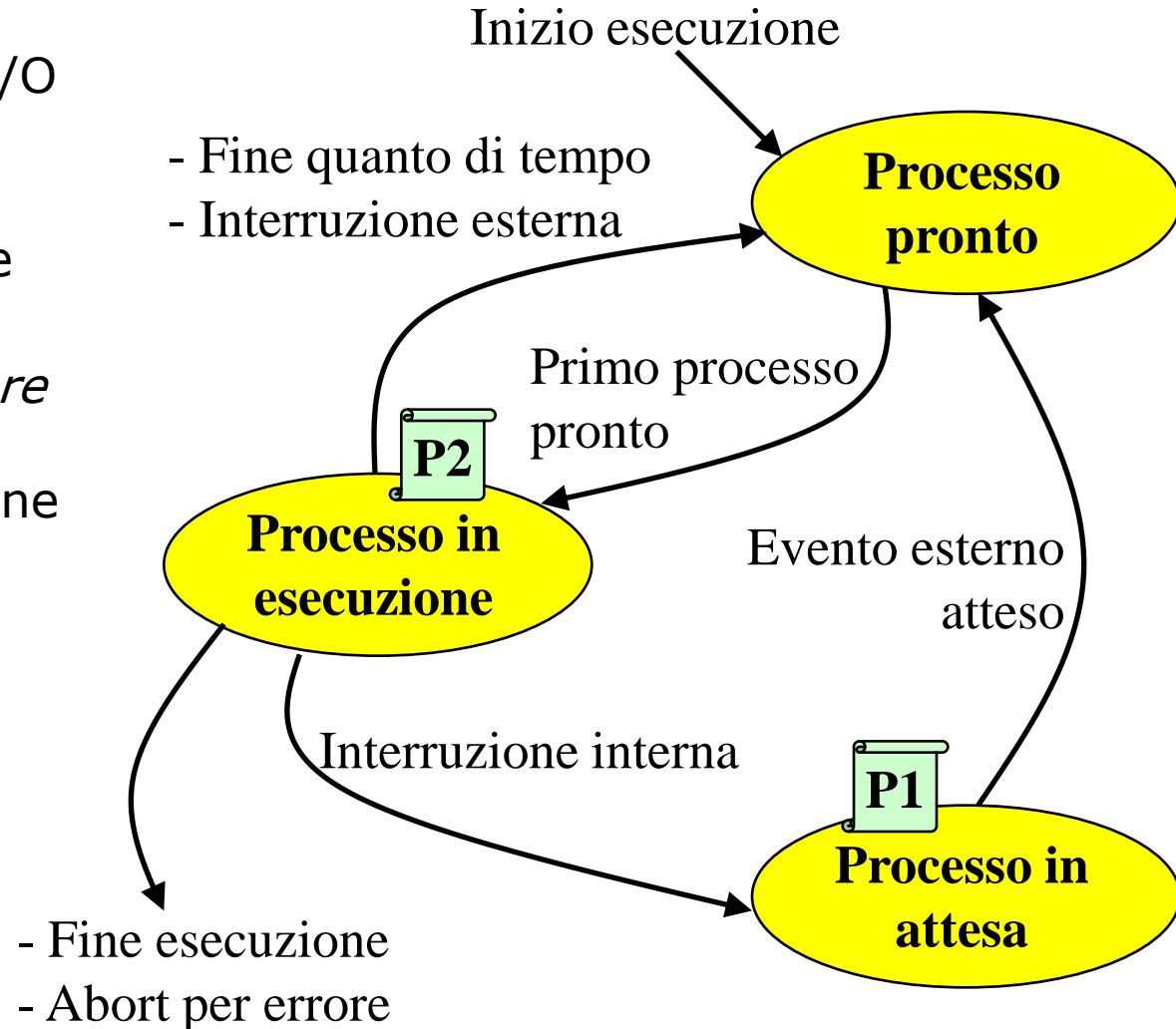
Stato di un processo (4)

- ❑ *Cambiamento di contesto:*
 - ▶ Salvare il *contesto* di P1 nel suo *descrittore di processo*
- ❑ Il processore è ora libero, un altro processo passerà in *esecuzione*



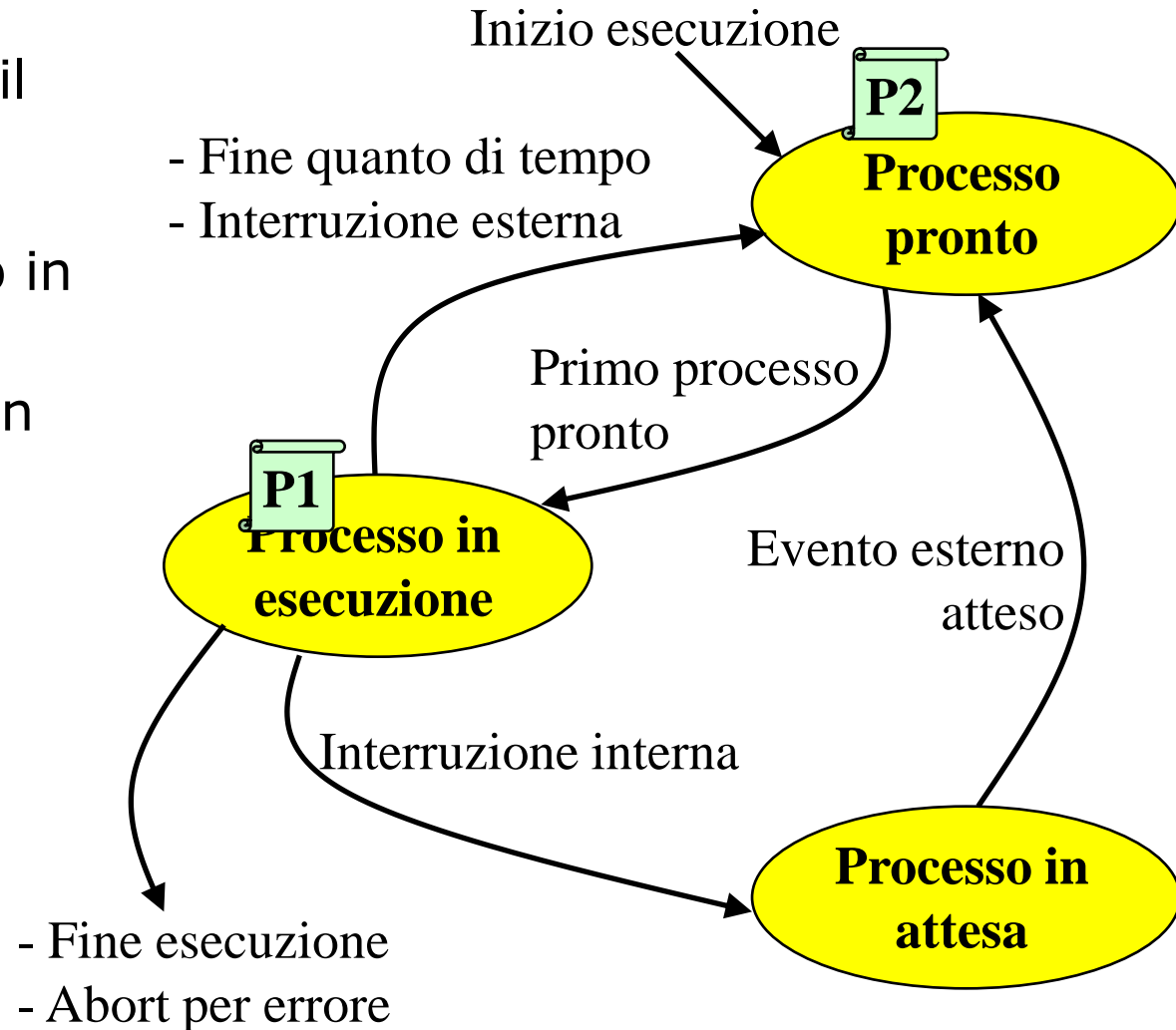
Stati di un processo (5)

- ❑ Quando l'operazione di I/O è finita viene generata un'*interruzione esterna*
- ❑ Il processo in esecuzione viene interrotto
- ❑ Il nucleo esegue il *gestore delle interruzioni* che esegue le azioni opportune
- ❑ P1 può tornare *pronto*
- ❑ Il nucleo sceglie quale processo mandare in esecuzione



Stato di un processo (6)

- ❑ *Pre-emption*: quando il quanto di tempo è scaduto, il nucleo interrompe il processo in esecuzione
- ❑ Si cerca di garantire un uso equo della CPU a tutti i processi



- Il quanto di tempo è gestito da una particolare interruzione, generata dall'orologio di sistema:
 - ▶ a una frequenza definita, il dispositivo che realizza l'orologio di sistema genera un'interruzione. La routine di risposta relativa incrementa una variabile opportuna che contiene il tempo di esecuzione del processo corrente
 - ▶ se il quanto di tempo non è scaduto la routine termina, se non ci sono interruzioni annidate, il processo prosegue nell'esecuzione
 - ▶ se invece il quanto di tempo è scaduto viene invocata una particolare funzione del nucleo (preempt) che cambia lo stato del processo da esecuzione a pronto, salva il contesto del processo e attiva una particolare funzione del nucleo (change) che esegue una commutazione di contesto e manda in esecuzione un processo pronto.

- Siano P e Q due processi lanciati su un sistema monoprocesso. P contiene una `scanf`, mentre Q non comporta alcuna chiamata al supervisor. Dire se ciascuna delle seguenti affermazioni é vera o falsa. Giustificare le risposte.
 - ▶ Il processo P potrebbe terminare senza mai essere mai essere nello stato "in attesa"

Falso.

Dal momento che contiene una `scanf` dovrà necessariamente effettuata una supervisor call e il suo stato diverrà "in attesa"

- Siano P e Q due processi lanciati su un sistema monoprocesso. P contiene una `scanf`, mentre Q non comporta alcuna chiamata al supervisor. Dire se ciascuna delle seguenti affermazioni é vera o falsa. Giustificare le risposte.
 - ▶ Se il processo Q viene lanciato prima di P allora Q termina sicuramente prima di P

Falso.

Non è possibile sapere quale processo terminerà prima a priori dal momento che ad ogni processo è garantito un quanto di tempo alla volta.

- Siano P e Q due processi lanciati su un sistema monoprocesso. P contiene una `scanf`, mentre Q non comporta alcuna chiamata al supervisor. Dire se ciascuna delle seguenti affermazioni é vera o falsa. Giustificare le risposte.
 - ▶ Una volta lanciato Q rimarrà sempre nello stato "in esecuzione"

Falso.

Non è possibile affermarlo con certezza: se Q dovesse terminare prima dello scadere del quanto di tempo allora rimmarrà sempre nello stato "in esecuzione", viceversa sarà posto nello stato di "pronto"