

Sistemi Operativi

AAF - Secondo anno - 3CFU

A.A. 2022/2023

Corso di Laurea in Matematica

I Processi - Parte 2

Annalisa Massini

Dipartimento di Informatica
Sapienza Università di Roma

Argomenti trattati

- 1 Stati di un processo
 - Processi a cinque stati
 - Processi sospesi

- 2 Descrizione del processo
 - Processi e risorse
 - Strutture di controllo del SO

Stati di un processo

Processi a cinque stati

Modello dei processi a 5 Stati

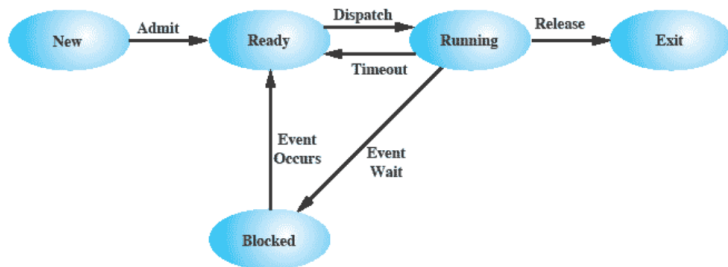
- Consideriamo il modello con la gestione a coda
- La coda è del tipo *first-in-first-out* (o FIFO)
- Il processore gestisce i processi in coda in *round-robin*, cioè ogni processo in coda riceve una certa quantità di tempo, poi torna in coda
- Il problema è che tra i processi *Not Running* ci sono:
 - processi *ready to execute*
 - processi *blocked* (in attesa di operazioni di I/O)

Modello dei processi a 5 Stati

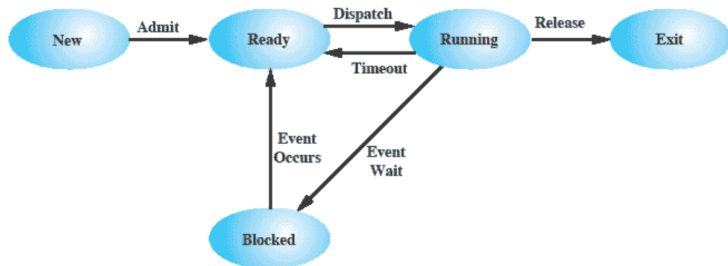
- Il dispatcher deve quindi cercare il processo non bloccato che si trova da più tempo in coda
- Usando una singola coda, il *dispatcher* non riesce a individuare il processo più vecchio in coda efficientemente
- Si può passare dal modello di processo a due stati al modello a cinque stati

Modello dei processi a 5 Stati

- Si risolve il problema di individuare il prossimo processo da mandare in esecuzione suddividendo lo stato *Not Running* nei due stati **Ready** e **Blocked**
- Si introducono gli stati *New* ed *Exit*

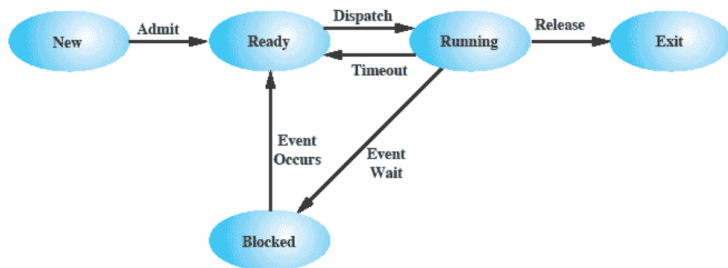


Modello dei Processi a 5 Stati



- Un processo appena creato, viene messo nella coda **Ready**
- Il SO sceglie il processo da far girare dalla coda *Ready*

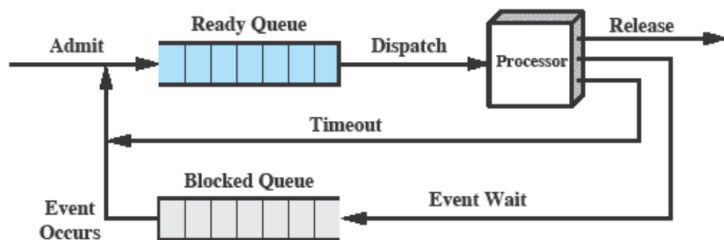
Modello dei Processi a 5 Stati



- Si può passare da ready a running
- Si può passare da running a ready oppure blocked
- Si può anche passare da ready o blocked ad exit (se un processo ne *uccide* un altro)
- *Waiting* è spesso usato in alternativa a *blocked*

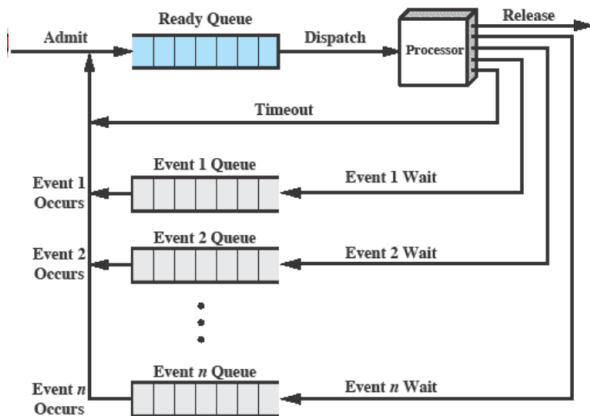
Gestione a Due Code

- Il processo *running*, cioè in esecuzione, e può:
 - terminare, oppure
 - essere posto in una delle code *Ready* o *Blocked*



Molteplici Code Bloccanti

- Per una gestione più efficiente delle centinaia/migliaia di processi si usano multiple code *Blocked* basate su tipi di eventi



Stati di un processo

Processi sospesi

Processi Sospesi

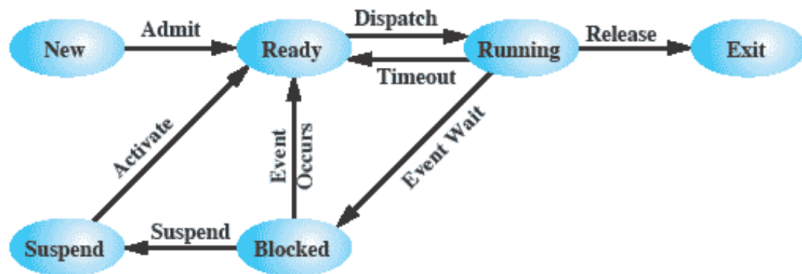
- Oltre agli stati *Ready*, *Running* e *Blocked* può essere utile usare altri stati
- Ogni processo deve essere in memoria principale (*non consideriamo per ora la memoria virtuale*)
- Il processore è più veloce dell'I/O, quindi tutti i processi attualmente in memoria potrebbero essere in attesa di I/O

Processi Sospesi

- Per non lasciare il processore inoperoso i processi vengono spostati su disco, **swap**, così da liberare memoria
- Quando il processo è *swappato* su disco lo stato da *Blocked* diventa **Suspend**
- Lo spazio liberato in memoria principale può essere usato per un altro processo:
 - un processo appena creato *oppure*
 - un processo precedentemente sospeso
 - *meglio un processo sospeso* per non sovraccaricare il sistema

Stato Suspended

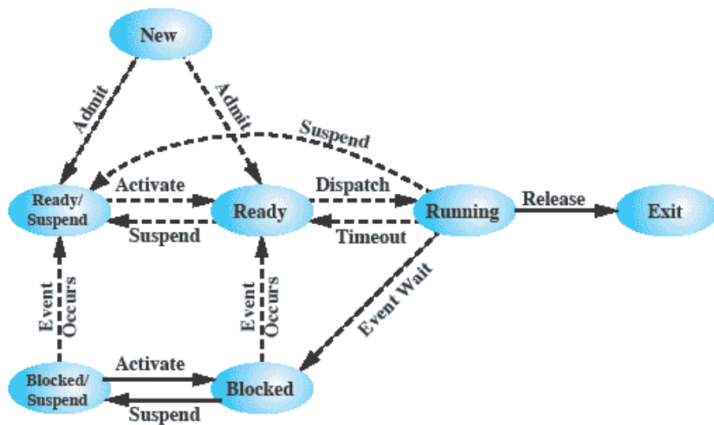
- Dallo stato *Blocked* si passa quindi allo stato **Suspend**
- E dallo stato **Suspend** si passa allo stato *Ready*



Processi Sospesi

- Non conviene riportare in memoria principale processi che non sono ancora pronti per l'esecuzione, cioè *bloccati*
- Ma se un processo era in attesa di un evento, quando l'evento si è verificato diventa pronto per l'esecuzione
- Due nuovi stati
 - **Blocked/Suspend** (swappato - in attesa dell'evento)
 - **Ready/Suspend** (swappato - pronto per l'esecuzione)

Due stati Suspend



- Si può andare ad Exit da un qualsiasi stato diverso da new
- Ad esempio se un processo ne *uccide* un altro

Motivi per Sospendere un Processo

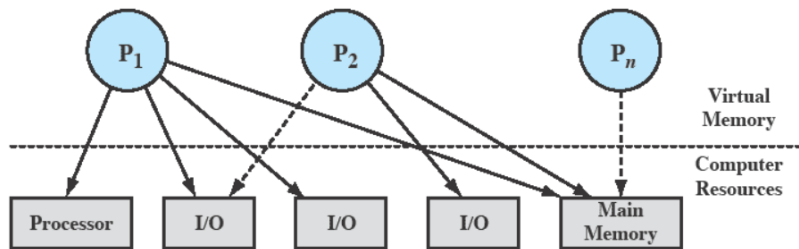
Motivo	Commento
Swapping	Il SO ha bisogno di liberare abbastanza memoria per caricare un processo ready
Interno al SO	Il SO sospetta che il processo stia causando problemi
Richiesta utente interattiva	Ad esempio: debugging o motivi legati all'uso di risorse
Periodicità	Il processo viene eseguito periodicamente (p.e. monitoraggio di sistema o accounting) e può venire sospeso in attesa della prossima esecuzione
Richiesta del padre	Il padre potrebbe sospendere l'esecuzione di un figlio per esaminarlo, modificarlo o per coordinare l'attività tra più figli

Descrizione del processo

Processi e risorse

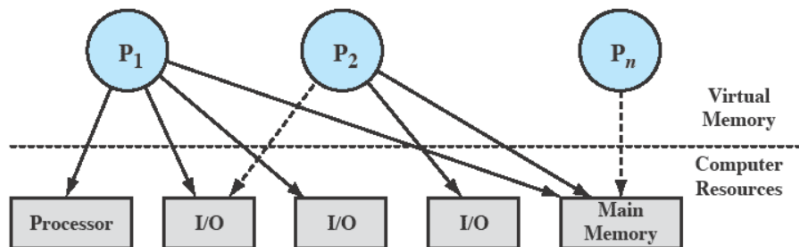
Processi e Risorse

- Il compito del sistema operativo è fondamentalemente la *gestione dell'uso delle risorse di sistema da parte dei processi, processore in primis*
- In un sistema multiprogrammato si ha un insieme di processi che competono per l'utilizzo delle risorse comuni



Processi e Risorse

- P_1 è running, quindi almeno in parte è in memoria principale e usa processore e dispositivi di I/O
- P_2 è in attesa dell'I/O utilizzato da P_1
- P_n è stato swappato ed è sospeso



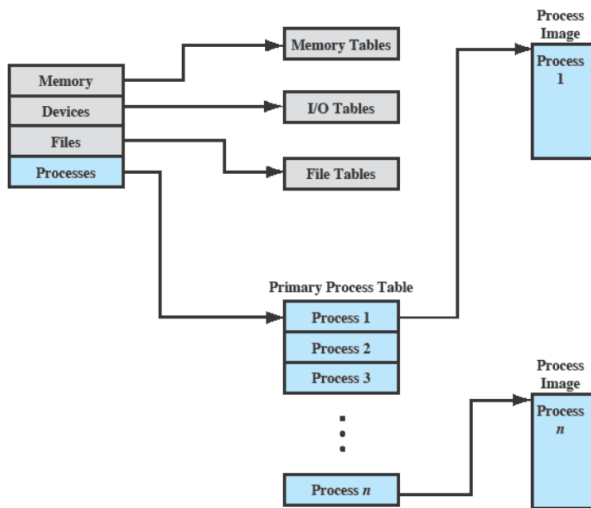
Descrizione del processo

Strutture di controllo del SO

Strutture di Controllo del SO

- Per gestire sia i processi che le risorse, il SO deve conoscere lo stato di ogni processo e di ogni risorsa
- Il SO costruisce e mantiene una o più tabelle per ognuna delle entità da gestire
- I quattro tipi di tabelle mantenute dal SO sono:
 - memoria
 - I/O
 - file
 - processi

Tabelle di controllo del SO



- Ci sono molti riferimenti incrociati

Tablelle di Memoria

- Le **tablelle di memoria** sono usate per gestire sia la memoria principale che quella secondaria
 - quella secondaria serve per la memoria virtuale (lo vedremo)
- Le **tablelle di memoria** devono comprendere le seguenti informazioni:
 - allocazione di memoria principale da parte dei processi
 - allocazione di memoria secondaria da parte dei processi
 - attributi di protezione per l'accesso a zone di memoria condivisa
 - informazioni per gestire la memoria virtuale

Tabelle per l'I/O

- Le **tabelle per l'I/O** sono usate dal SO per gestire i dispositivi e i canali di I/O
- Il SO deve sapere:
 - se il dispositivo è disponibile o già assegnato
 - lo stato dell'operazione di I/O
 - la locazione in memoria principale usata come sorgente o destinazione del trasferimento di I/O

Table dei File

- Le **tabelle dei file** forniscono informazioni su:
 - esistenza di files
 - locazioni in memoria secondaria
 - stato corrente
 - altri attributi
- Sono memorizzate parte su disco e parte in RAM