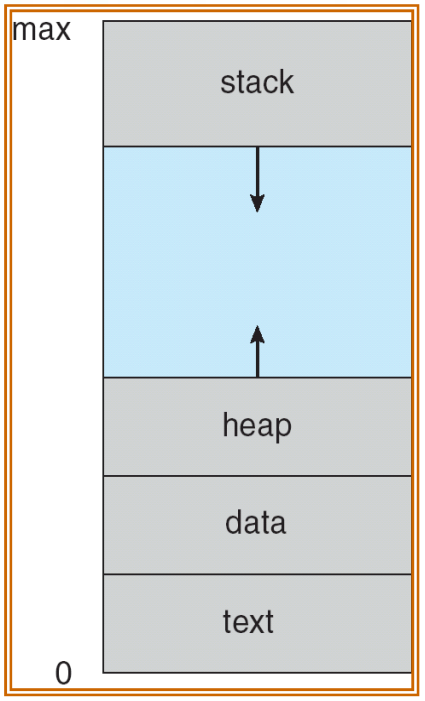
Processi

Processi: aspetti teorici

Stefano Quer
Dipartimento di Automatica e Informatica
Politecnico di Torino

Programmi e processi

- Programma
  - Entità passiva
  - Sequenza di linee di codice
- Processo
  - Programma in esecuzione
  - Entità attiva
    - Codice sorgente e program counter
    - Area dati (variabili globali)
    - Stack (parametri e variabili locali) e stack pointer
    - Heap (variabili dinamiche allocate durante l'esecuzione del processo)

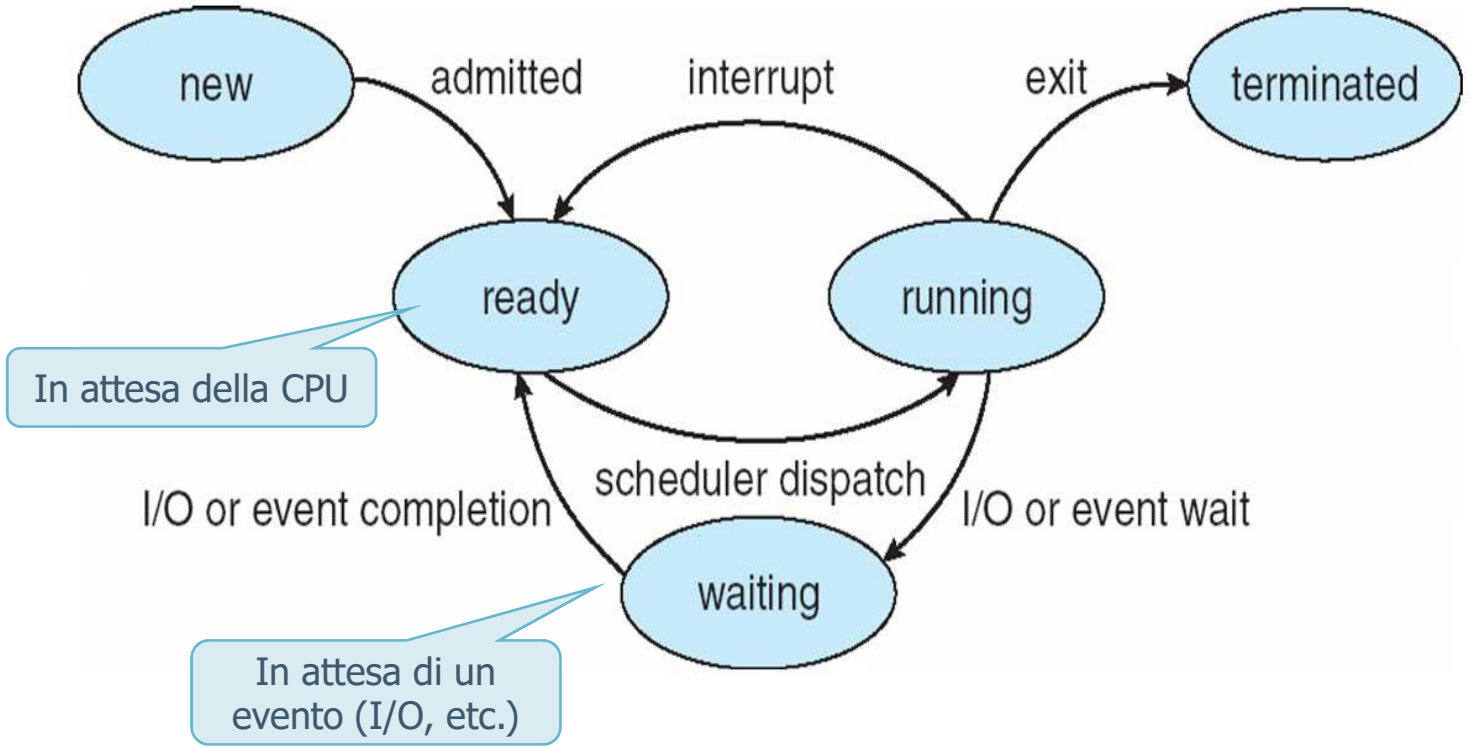


Stato di un processo

- Durante la sua esecuzione un processo cambia di stato
  - **New**: il processo viene creato e sottomesso al SO
  - **Running**: in esecuzione
  - **Ready**: Logicamente pronto ad essere eseguito, in attesa della risorsa processore
  - **Waiting**: in attesa della disponibilità di risorse da parte del sistema oppure di un qualche evento
  - **Terminated**: Il processo termina e rilascia le risorse utilizzate

Diagramma a stati

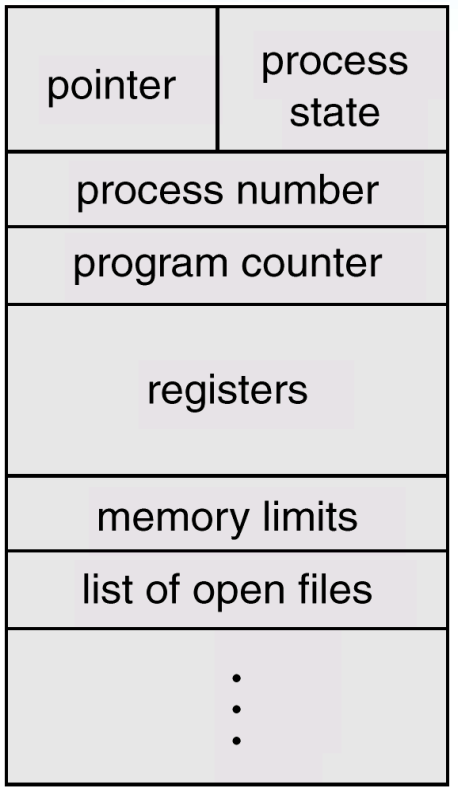
- L'evoluzione temporale di un processo è descritta da un diagramma a stati



Process Control Block (PCB)

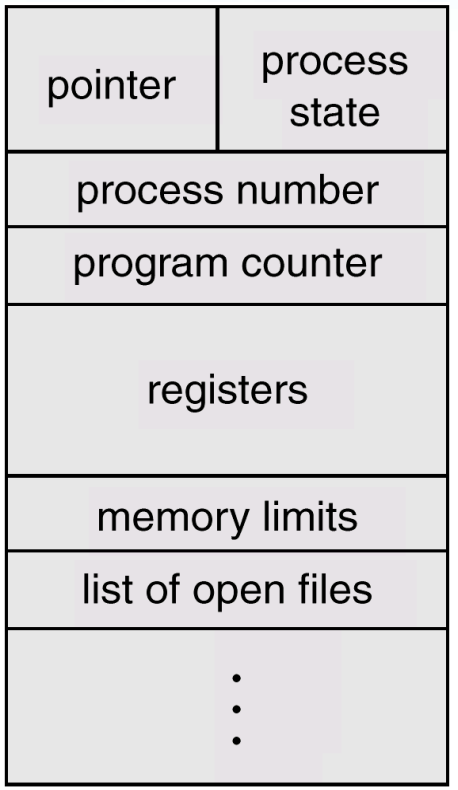
Tabella dei processi

- Il SO tiene traccia di ogni processo associando a esso un insieme di dati
- Tali dati includono
  - Stato del processo
    - New, Ready, Running, Waiting, Terminated
  - Program counter
    - Indirizzo della successiva istruzione da eseguire



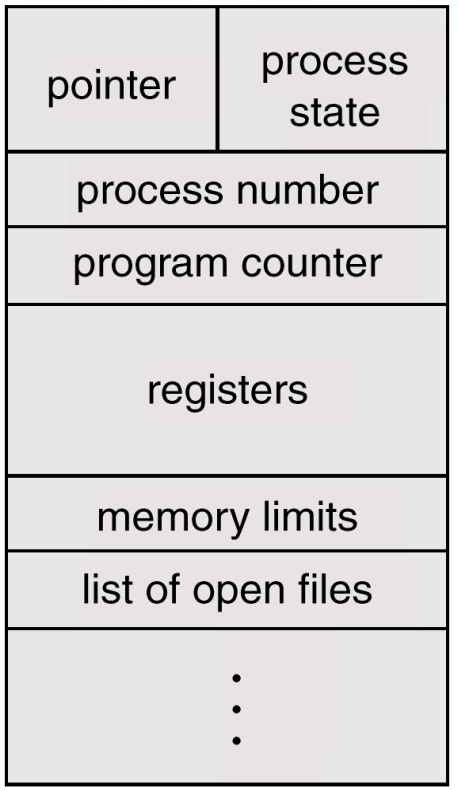
Process Control Block (PCB)

- Registri della CPU
  - In numero e tipo dipendente dall'hardware
- Informazioni utili per lo scheduling della CPU
  - Priorità, puntatori alle code di scheduling, etc.
- Informazioni utili per la gestione della memoria
  - Registro base, registro limite, tabelle pagine o segmenti, etc.



Process Control Block (PCB)

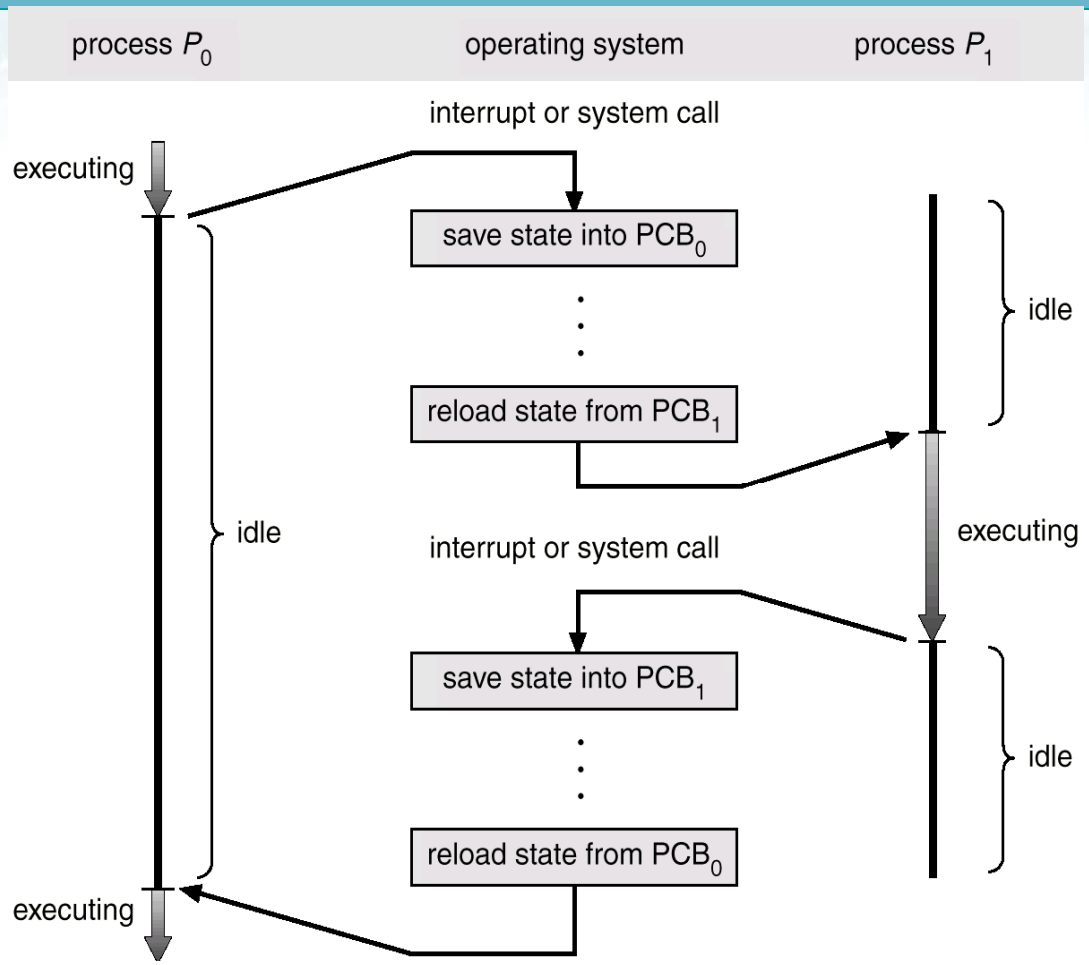
- Informazioni amministrative varie
  - Tempo di utilizzo CPU, limiti, etc.
- Informazioni sullo stato delle operazioni di I/O
  - Lista dispositivi di I/O, lista dei file aperti, etc.



Context switching

- Quando la CPU viene assegnata a un altro processo, il kernel deve
  - Salvare lo stato del processo running
  - Caricare un nuovo processo ripristinandone lo stato salvato precedentemente
- Il tempo dedicato al **context switching** è overhead, cioè lavoro non utile direttamente ad alcun processo
- Il tempo che un sistema usa per il context switching dipende da svariati fattori
  - Hardware a disposizione, SO, numero di processi, politica di scheduling, etc.

Context switching



Scheduling dei processi

- L'obiettivo della multiprogrammazione è quello di massimizzare l'utilizzo della CPU da parte dei processi
- I processi possono essere classificati in
  - I/O-bound
    - Passano più tempo effettuando I/O che calcoli
    - Richiedono molti servizi corti da parte della CPU
  - CPU-bound
    - Passano più tempo effettuando calcoli che I/O
    - Richiedono pochi servizi molto lunghi da parte della CPU

Scheduling dei processi

- Per massimizzare l'utilizzo della CPU, e soddisfare eventuali vincoli sul tempo di risposta, ogni SO gestisce i processi mediante uno **scheduler**
  - Compito di uno scheduler è quello di selezionare tra i processi disponibili il successivo processo da eseguire
  - Esempi
    - Dopo una fork procede padre o figlio
    - Quando un processo termina chi parte?
    - Quando un processo fa I/O chi parte?

Scheduling dei processi

- Esistono diversi tipi di scheduler
  - Scheduler a lungo termine (long-term scheduler)
    - Interviene molto meno frequentemente
    - Rischedula a tempi dell'ordine dei secondi/minuti
    - Seleziona quale processo inserire nella ready list e quali sono pronti all'esecuzione in memoria centrale
    - In pratica controlla il grado di multiprogrammazione

Schedula i processi su disco

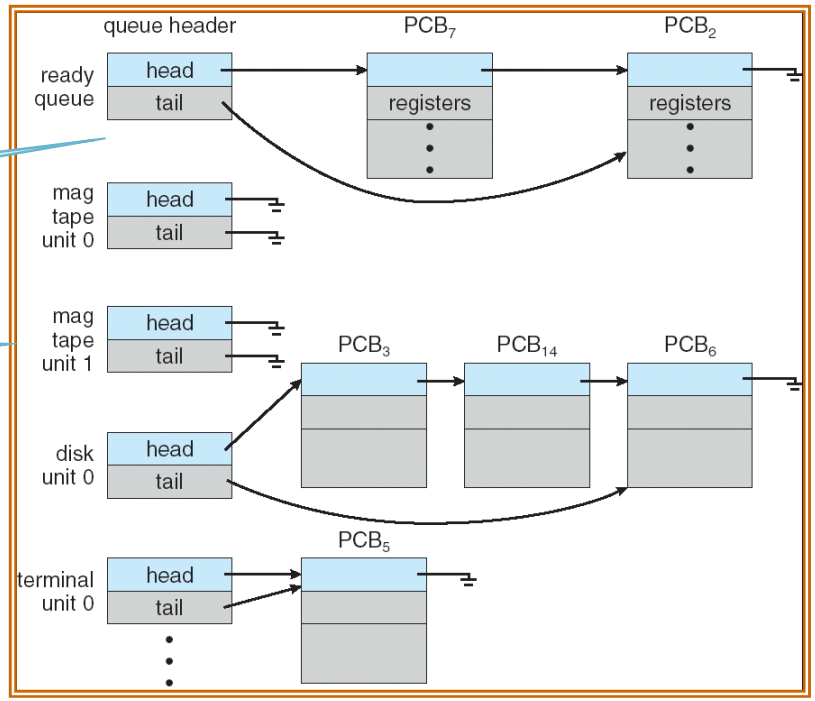
  - Scheduler a breve termine (short-term scheduler)
    - Seleziona prevalentemente processi per la CPU
    - Interviene molto frequentemente
    - Rischedula a tempi dell'ordine dei millisecondi
    - Deve essere molto veloce

Schedula i processi in RAM

Code di scheduling

Analisi statica dei processi

- Lo scheduler gestisce i processi in attesa di un dispositivo mediante **code** (**di processi**)
  - Esistono diverse code una per dispositivo
  - Ogni coda è una lista concatenata



Coda dei processi ready

Coda dei processi in attesa di I/O

Per massimizzare l'efficienza ogni dispositivo ha la sua coda

Analisi dinamica dei processi

Diagramma di accodamento

- Il diagramma di accodamento specifica la gestione dei processi (transizioni) nelle varie code
  - Ogni rettangolo rappresenta una coda

Il processo termina il propri compiti

Il processo si posiziona inizialmente nella ready queue

Il processo assegnato alla CPU ritorna in stato di attesa (I/O, interrupt, fork, etc.)