… ma come determinare la lunghezza del prossimo CPU burst?!

Si puo' ***stimare*** tenendo traccia dei ***precedenti***

1. t_n = actual lenght of n^{th} CPU burst
2. τ_{n+1} = predicted value for the next CPU burst
3. $\alpha, 0 \le \alpha \le 1$
4. Define:

$$\tau_{n+1} = \alpha\, t_n + (1-\alpha)\tau_n.$$

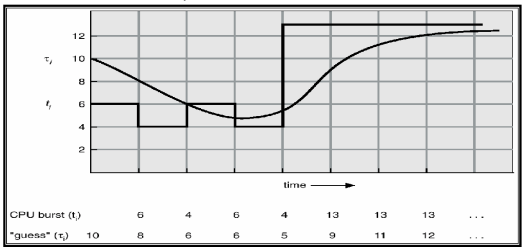
… α rappresenta il peso che vogliamo dare alla storia recente

- α ***=0***
  - $\tau_{n+1} = \tau_n$
  - La storia recente non conta mai, e quindi il valore rimane costante
- α ***=1***
  - $\tau_{n+1} = t_n$
  - Solo il valore dell'ultimo CPU burst conta
- α ***=1/2***
  - $\tau_{n+1} = (t_n + \tau_n)/2$
  - Storia e ultimo valore hanno lo stesso peso

Questione aperta

$$\tau_{n=1} = \alpha\, t_n + (1-\alpha)\tau_n.$$

Provare a espandere la formula per comprendere il significato del peso dato ai valori passati e il comportamento asintotico



3. Con priorita'

- Un ***numero*** (intero) ***di priorita'*** e' assegnato ad ogni processo
- La CPU e' allocata al processo con la piu' alta priorita' (di solito il processo con l'intero piu' piccolo e' quello a piu' alta priorita'! Es. UNIX)

… parametri che possono indurre il livello di priorita' di un processo…

limiti di tempo
requisiti di memoria
numero di file aperti
rapporto tra I/O e CPU burst

INTERNI

importanza del processo
tipologia di utente
fondi per l'utilizzo del PC

ESTERNI

… e alcune considerazioni

SJF puo' essere considerata una politica dove la priorita' e' il tempo del prossimo CPU burst

Piu' in generale, una politica con priorita' puo' essere sia preemptive che non-preemptive

Un problema : la starvation

Processi a bassa priorita' potrebbero non essere eseguiti mai!

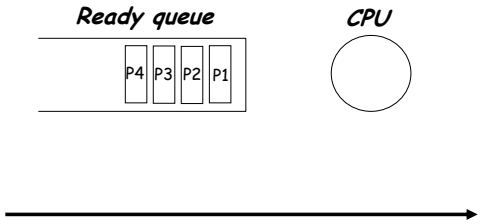
E una possibile soluzione : ***aging***

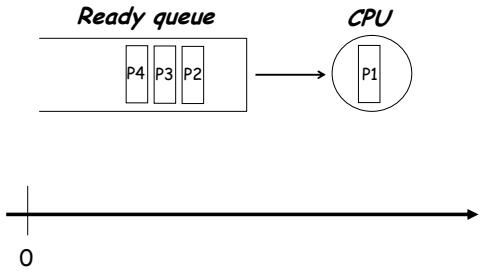
Col passare del tempo la priorita' di un processo cresce

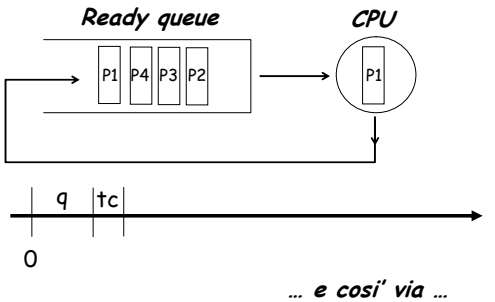
4. Round Robin (RR)

Ad ogni processo viene assegnata una piccola unita' di tempo di CPU (***quanto***), usualmente dell'ordine di alcune decine di millisecondi.

Quando questo tempo scade, il processo e' ***preempted*** e viene inserito ***alla fine della ready queue***



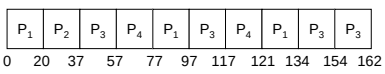




… e cosi' via …

Esempio: q = 20 ; tc = 0

Processo	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24



Usualmente un piu' alto turnaround time medio, ma un piu' breve tempo di risposta

Alcune considerazioni

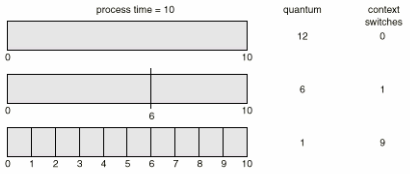
Se ci sono ***n*** processi nella ready queue e il quanto e' ***q***, allora ogni processo utilizza una frazione ***1/n*** della CPU in segmenti di durata ***q***

Nessun processo attende piu' di ***(n-1)q*** unita' di tempo per guadagnare la prima volta la CPU

Lunghezza del quanto:

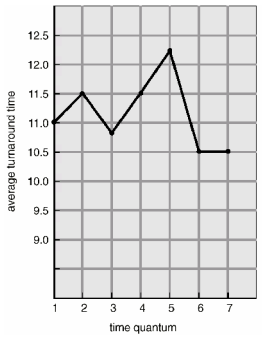
q molto ***grande*** ⇒ FCFS
q molto ***piccolo*** ⇒ confrontabile con l'overhead introdotto dal context switching

Lunghezza del quanto e numero di context-switching



E' consigliabile che l'80% dei CPU burst siano piu' piccoli del quanto

Il Turnaround Time dipende dal quanto

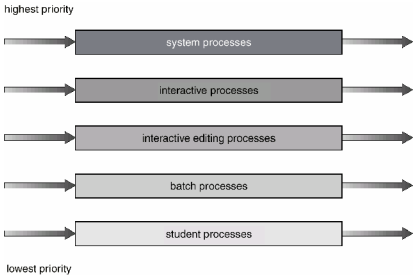


process	time
P_1	6
P_2	3
P_3	1
P_4	7

5. Code multilivello

- La ready queue e' partizionata in ***code separate*** e ad ogni coda viene assegnata una ***priorita'***
- Si servono prima tutti i processi in code ad alta priorita' e poi via via gli altri (preemptive o non-preemptive) → ***starvation***

Una rappresentazione grafica

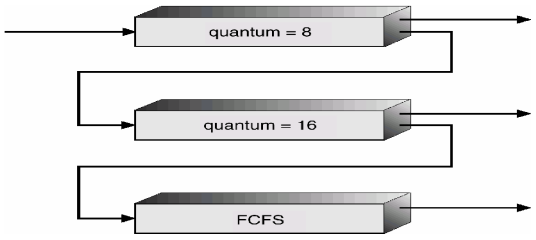


Ogni coda al suo interno ha la ***propria politica di scheduling***

Soluzioni alla starvation: CPU slicing

Ad ogni coda viene assegnata una certa ***percentuale di tempo di CPU*** che dividera' tra i suoi processi

Soluzioni alla starvation: code con feedback



Una speciale tecnica di ***aging*** :
un processo si puo' muovere tra le code

Esempio

- Tre code:
  - Q_0 – quanto: 8 milliseconds
  - Q_1 – quanto: 16 milliseconds
  - Q_2 – FCFS
- Scheduling
  - Un nuovo job entra nella coda Q_0 . Quando guadagna la CPU ci sta per 8 millisecondi. Se non finisce va della coda Q_1 .
  - In Q_1 il job aspetta il suo turno e quindi riceve la CPU per altri 16 milliseconds. Se ancora non ha completato va nella coda Q_2.

Parametri delle code con feedback

- Numero di code
- Algoritmo di scheduling di ogni coda
- Metodo usato per la migrazione dei processi tra code (anche eventualmente verso l'alto!)
- Coda in cui allocare un processo appena entrato