

Politiche di scheduling

Index

- [Introduction](#)
 - [Funzione di selezione](#)
 - [Modalità di decisione](#)
 - [Scenario comune di esempio](#)
 - [FCFS \(First Come First Served\)](#)
 - [Round-Robin](#)
 - [Misura del quanto di tempo per la preemption](#)
 - [CPU-bound vs. I/O-bound](#)
 - [SPN \(Shortest Process Next\)](#)
 - [Come stimare il tempo di esecuzione?](#)
 - [SRT \(Shortest Remaining Time\)](#)
 - [HRRN \(Highest Response Ratio Next\)](#)
 - [Confronto tra le varie politiche](#)
-

Introduction

	FCFS	Round robin	SPN	SRT	HRRN	Feedback
Selection function	$\max[w]$	constant	$\min[s]$	$\min[s - e]$	$\max\left(\frac{w + s}{s}\right)$	(see text)
Decision mode	Non-preemptive	Preemptive (at time quantum)	Non-preemptive	Preemptive (at arrival)	Non-preemptive	Preemptive (at time quantum)
Throughput	Not emphasized	May be low if quantum is too small	High	High	High	Not emphasized
Response time	May be high, especially if there is a large variance in process execution times	Provides good response time for short processes	Provides good response time for short processes	Provides good response time	Provides good response time	Not emphasized
Overhead	Minimum	Minimum	Can be high	Can be high	Can be high	Can be high
Effect on processes	Penalizes short processes; penalizes I/O bound processes	Fair treatment	Penalizes long processes	Penalizes long processes	Good balance	May favor I/O bound processes
Starvation	No	No	Possible	Possible	No	Possible

Le colonne rappresentano gli algoritmi per la politica di scheduling, nei sistemi moderni vengono attuate più politiche insieme

Funzione di selezione

La funzione di selezione è quella che sceglie effettivamente il processo da mandare in esecuzione. Se è basata sulle caratteristiche dell'esecuzione, i parametri da cui dipende sono:

- $w \rightarrow$ tempo trascorso in attesa
- $e \rightarrow$ tempo trascorso in esecuzione
- $s \rightarrow$ tempo totale richiesto, incluso quello già servito (quindi va stimato o fornito come input insieme alla richiesta di creazione del processo)

Modalità di decisione

La modalità di decisione specifica in quali istanti di tempo la funzione di selezione viene invocata. Ci sono due possibilità:

- **preemptive**

Il sistema operativo può interrompere un processo indipendentemente da eventi terzi (lo può decidere per conto suo). In questo caso il processo diverrà ready. Questa può avvenire o per l'arrivo di nuovi processi (appena forkati) o per un

interrupt (può essere di I/O, un processo blocked diventa ready, o di clock, periodico per evitare che un processo monopolizzi il sistema)

- **non-preemptive**

Se un processo è in esecuzione, allora arriva o fino a terminazione o fino ad una richiesta di I/O (o comunque ad una richiesta bloccante)

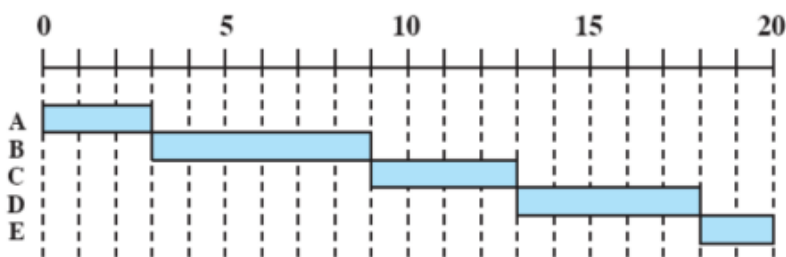
Scenario comune di esempio

5 processi batch

Processo	Tempo di arrivo	Tempo di esecuzione
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

FCFS (First Come First Served)

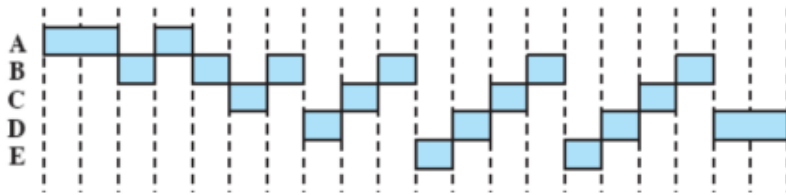
La politica FCFS è una politica non-preemptive. Quando un processo smette di essere eseguito, si passa al processo che ha aspettato di più nella coda ready finora



Il problema di questo tipo di politica è che un processo “corto” potrebbe dover attendere molto prima di essere eseguito (come è capitato per il processo E nell’esempio). Tende inoltre a favorire i processi che usano molto da CPU (CPU-bound), che infatti, una volta preso possesso della CPU, non viene rilasciata finché il processo non termina

Round-Robin

La politica Round-Robin è una politica preemptive, basandosi sul clock. In questa politica è infatti necessario fissare un'unità di tempo che determina il tempo di esecuzione di ogni processo

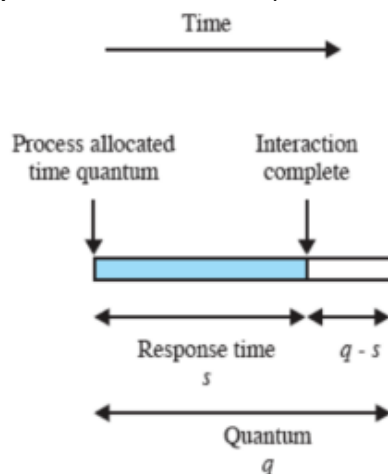


Nella pratica un'interruzione di clock viene generata ad intervalli periodici, e quando questa arriva il processo attualmente in esecuzione viene rimesso nella coda dei ready (ovviamente se il processo in esecuzione arriva ad un'istruzione I/O prima dell'interruzione allora viene spostato nella coda dei blocked) e il prossimo processo ready nella coda viene selezionato

T	Coda (prima)	Coda (dopo)	Esecuzione
0			A[3]
1			A[2]
2	B[6]	A[1]	B[6]
3	A[1]	B[5]	A[1]
4	B[5], C[4]	C[4]	B[5]
5	C[4]	B[4]	C[4]
6	B[4], D[5]	D[5], C[3]	B[4]
7	D[5], C[3]	C[3], B[3]	D[5]
8	C[3], B[3], E[2]	B[3], E[2], D[4]	C[3]
9	B[3], E[2], D[4]	E[2], D[4], C[2]	B[3]
10	E[2], D[4], C[2]	D[4], C[2], B[2]	E[2]

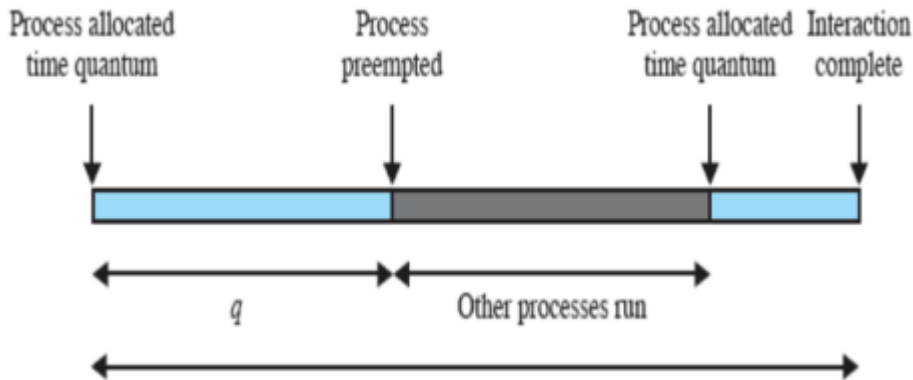
Misura del quanto di tempo per la preemption

Il quanto di tempo deve essere poco più grande del "tipico" tempo di interazione di un processo (tempo che ci mette un processo a rispondere, il round-robin è tipico di processi interattivi)



Se invece si scegliesse un quanto di tempo minore del tipico tempo di interazione il tempo di risposta per un processo aumenta notevolmente (vengono infatti mandati in

esecuzione tutti gli altri processi prima di calcolare la risposta)

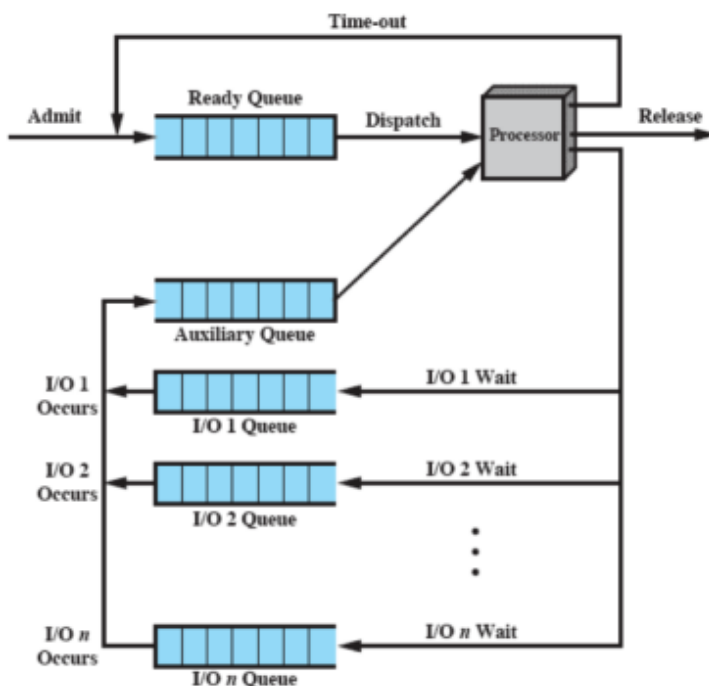


Ma se lo si fa troppo lungo, potrebbe durare più del tipico processo e il round-robin degenera in FCFS

CPU-bound vs. I/O-bound

I processi CPU-bound con il round-robin sono favoriti, infatti vuol dire che il proprio quanto di tempo viene usato per intero o quasi. Invece gli I/O bound ne usano solo una porzione, infatti non appena arriva una richiesta bloccante, il processo va nella coda dei blocked.

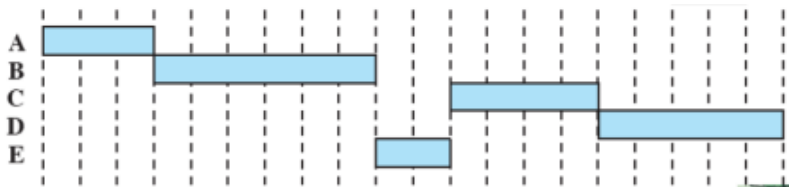
Risulta quindi essere non equo e non efficiente per l'I/O



Come soluzione è stato proposto il round-robin **virtuale**; infatti se un processo fa una richiesta bloccante, una volta che è stata esaudita, non va nella coda dei ready come accadrebbe solitamente, bensì viene direttamente messo in una coda prioritaria che viene scelta per prima dal dispatcher e vengono eseguiti per il quanto di tempo rimanente dalla precedente esecuzione prima di essere bloccato

SPN (Shortest Process Next)

La politica shortest process next è una politica non-preemprive. Questa manda in esecuzione il processo con tempo di esecuzione più breve (tempo di esecuzione stimato), permettendo ai processi più corti di scavalcare i più lunghi. Il tempo di esecuzione stimato, come precedentemente detto può essere calcolato dal sistema operativo oppure fornito dal processo stesso



Un problema di questa politica è il fatto che potrebbe andare incontro a starvation, infatti i processi più lunghi potrebbero non andare mai in esecuzione se continuano ad arrivare processi corti. Bisogna anche tener presente che se il tempo di esecuzione stimato si rivela inesatto, il sistema operativo può abortire il processo, dunque se il tempo previsto è maggiore del tempo reale, il processo può essere terminato dal sistema operativo

Come stimare il tempo di esecuzione?

In alcuni sistemi ci sono processi (sia batch che interattivi) che sono eseguiti svariate volte, quindi si usa il passato (T_i) per predire il futuro (S_i)

Questo può essere fatto con una media, ma vorrebbe dire per il dispatcher il doversi tenere in memoria tutti i tempi delle precedenti esecuzioni

$$S_{n+1} = \frac{1}{n} \sum_{i=1}^n T_i$$

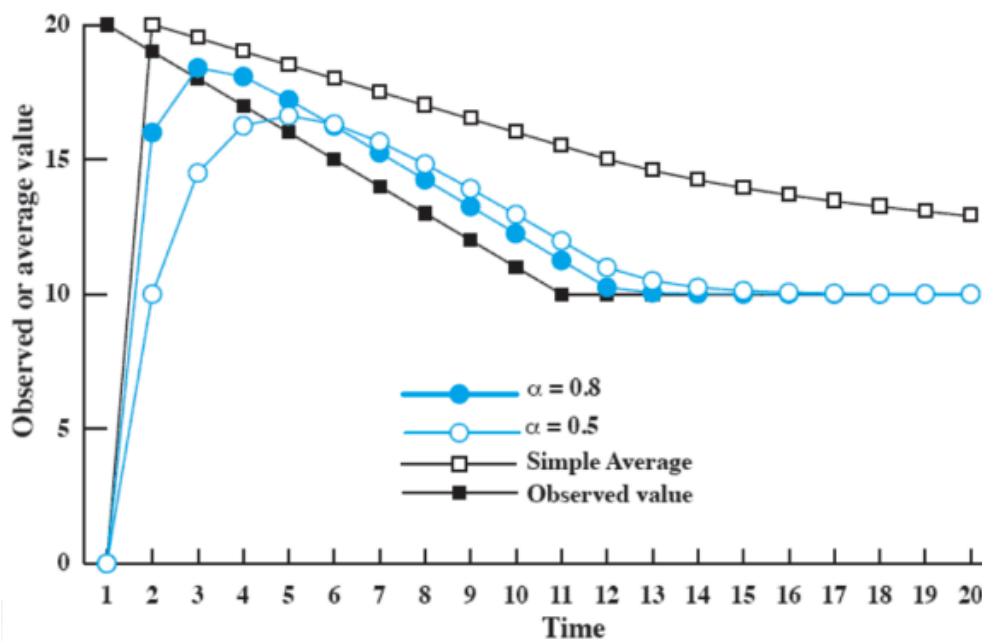
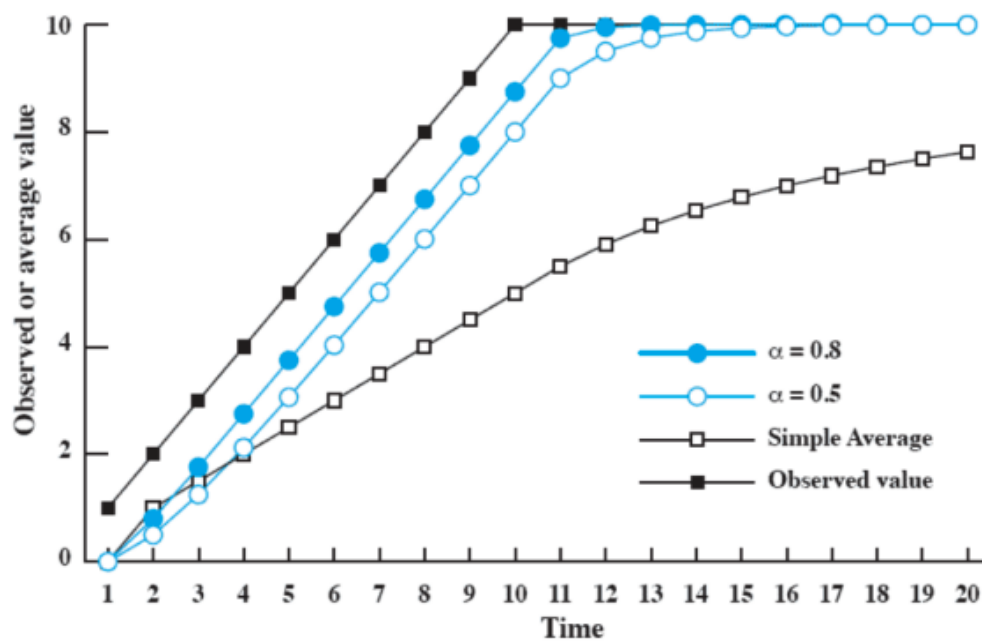
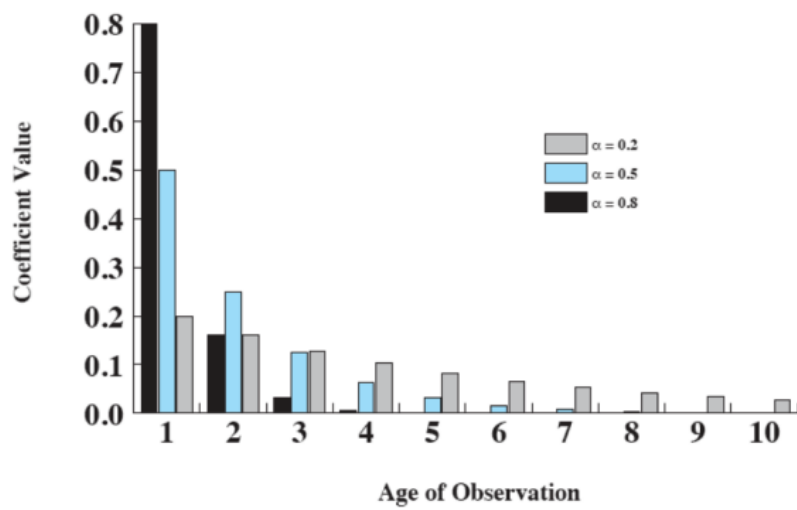
Per fortuna si può fare anche in un altro modo che necessita di ricordare solamente l'ultimo tempo di esecuzione e l'ultima stima fatta

$$S_{n+1} = \frac{1}{n} T_n + \frac{n+1}{n} S_n$$

Questa formula può essere generalizzata chiamata *exponential averaging*

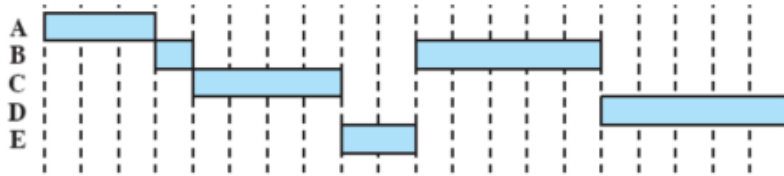
$$S_{n+1} = \alpha T_n + (1 - \alpha) S_n, \quad 0 < \alpha < 1$$

$$S_{n+1} = \alpha T_n + \dots + \alpha(1 - \alpha)^i T_{n-1} + \dots + (1 - \alpha)^n S_1$$



SRT (Shortest Remaining Time)

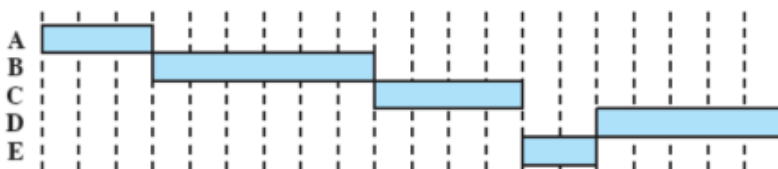
La politica shortest remaining time è simile alla SPN, ma preemptive. Questa infatti non utilizza un quanto di tempo, dunque un processo può essere bloccato solo quando ne arriva uno nuovo appena creato (o se fa un I/O bloccante). Una volta che un processo è stato interrotto viene rimesso nella coda dei ready; all'arrivo di un nuovo processo viene stimato il tempo rimanente richiesto per l'esecuzione, e viene eseguito il processo con quello più breve



HRRN (Highest Response Ratio Next)

La politica highest response ratio next è una politica non preemptive. Questa richiede la conoscenza del tempo di servizio e risolve il problema della starvation. Qui viene massimizzato il seguente rapporto (oltre al tempo che ci mette viene confrontato con il tempo in cui è stato fatto aspettare, rendendo la decisione più equa):

$$\frac{w + s}{s} = \frac{\text{tempo trascorso in attesa} + \text{tempo totale richiesto}}{\text{tempo totale richiesto}}$$



Confronto tra le varie politiche

