

## **Lezione 3 – Politiche di schedulazione**

Sistemi Operativi I

Modulo 3 - Gestione del processore

Unità didattica 3 - Schedulazione

**Vincenzo Piuri**

---

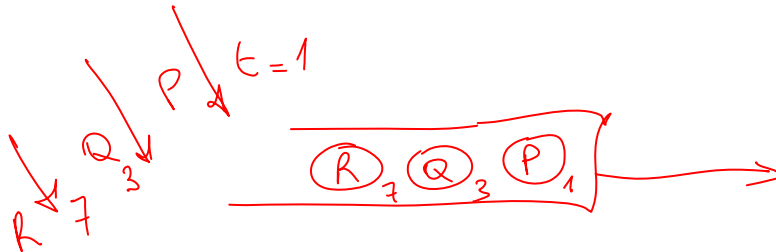
Università degli Studi di Milano - SSRI - CDL ONLINE

### **Sommario**

- Politiche e algoritmi di schedulazione
  - First Come, First Served
  - Shortest Job First
  - Priorità
  - Round Robin
  - Coda a più livelli
  - Coda a più livelli con retroazione
- Schedulazione in sistemi multiprocessore

## First-Come, First-Served: FCFS (1)

Primo Arrivato, Primo Servito



Non pre-emptive

## First-Come, First-Served: FCFS (2)

- Processi CPU-bound possono monopolizzare processore
- Tempo di attesa può essere alto

Processo	$P_1$	$P_2$	$P_3$
Tempo di elaborazione	24	3	3

Ordine di arrivo:  $P_1, P_2, P_3$

	$P_1$	$P_2$	$P_3$
0			
24			
27			
30			

Tempo di attesa medio: 17

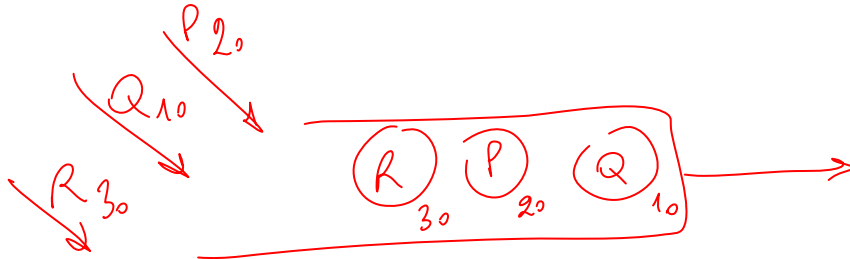
Ordine di arrivo:  $P_2, P_3, P_1$

	$P_2$	$P_3$	$P_1$
0			
3			
6			
30			

Tempo di attesa medio: 3

## Shortest-Job-First: SJF (1)

### Processo Più Breve Prima

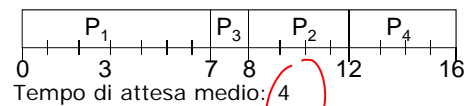


- **Pre-emptive:** processo che diventa pronto interrompe processo in esecuzione richiedendo schedulazione
- **Non pre-emptive:** processo che diventa pronto non interrompe processo in esecuzione richiedendo schedulazione

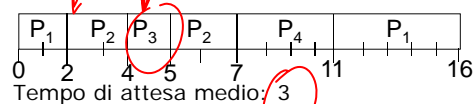
## Shortest-Job-First: SJF (2)

Processo	$P_1$	$P_2$	$P_3$	$P_4$
Tempo di arrivo	0	2	4	5
Tempo di elaborazione	7	4	1	4

- Non pre-emptive



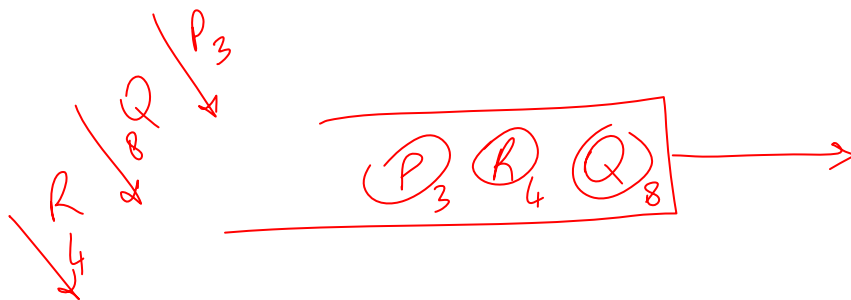
- Pre-emptive



### Shortest-Job-First: SJF (3)

- Garantisce il tempo minimo di attesa, se si conosce il tempo di processore richiesto dai processi
- Predizione del tempo di processore richiesto da un processo:
  - simile ai tempi precedenti del processo
  - media esponenziale  $\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n$

### Priorità (1)



- Rappresentazione della priorità
  - Logica diretta: Indice di priorità alto → Priorità alta
  - Logica inversa: Indice di priorità basso → Priorità alta

### Priorità (2)

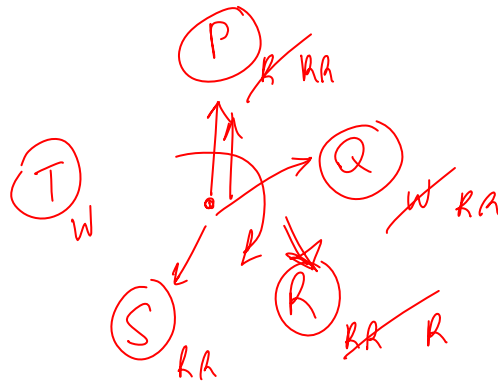
- **Pre-emptive:** processo che diventa pronto interrompe processo in esecuzione richiedendo schedulazione
- **Non pre-emptive:** processo che diventa pronto non interrompe processo in esecuzione richiedendo schedulazione

### Priorità (3)

- Problema:  
processi a bassa priorità potrebbero subire un blocco indefinito (*starvation*)
- Soluzione:  
progressivo invecchiamento della priorità (*aging*)  
con periodico ripristino al valore iniziale oppure ringiovanimento fino al valore iniziale

## Round-Robin RR (1)

### Rotazione



- Tipico dei sistemi time sharing
- Simile a FCFS con aggiunta di pre-emption

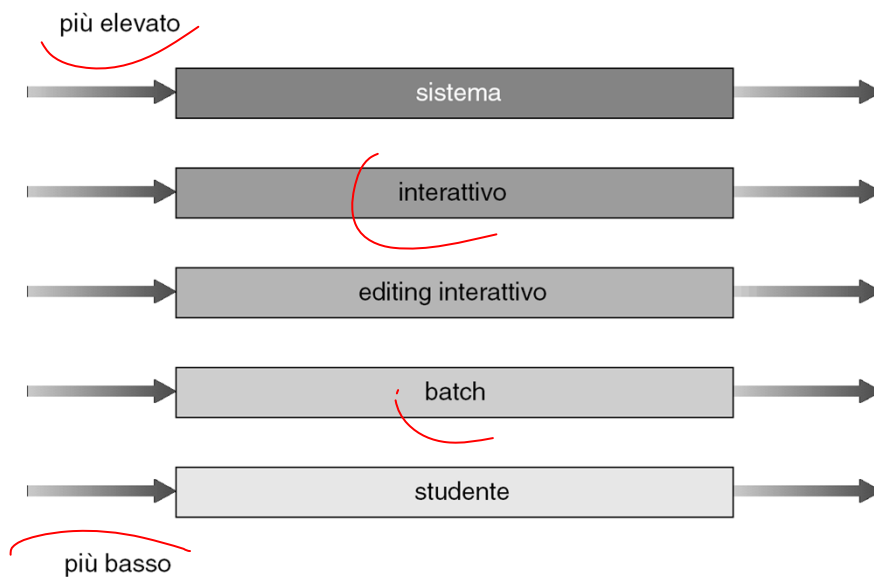
## Round-Robin RR (2)

- Distribuzione uniforme del tempo di elaborazione tra i processi pronti
- Velocità di esecuzione dei processi dipende dal numero di processi pronti
- Turnaround dipende dalla durata del quanto di tempo
- Comportamento dell'algoritmo RR dipende dalla durata del quanto di tempo:
  - molto lungo: RR  $\rightarrow$  FCFS
  - molto breve: RR  $\rightarrow$  condivisione del processore  
ognuno degli  $N$  processi sembra vedere un processore con  $1/N$  di capacità computazionale  
problema: sovraccarico di gestione dovuto ai frequenti cambiamenti di contesto
  - ideale empirico: 80% delle richieste di elaborazione deve essere completata in un quanto di tempo

### Coda a più livelli C+L (1)

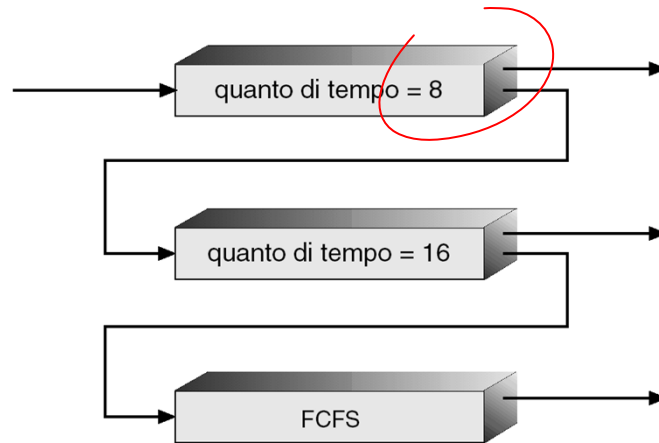
- I processi sono raggruppati per tipologie omogenee
- Ogni tipologia è assegnata in modo permanente a un livello della coda di schedulazione, rappresentato da una coda di attesa specifica
- Ogni coda di attesa ha un suo algoritmo di schedulazione
- L'insieme delle code di attesa viene schedulato da un algoritmo dedicato
  - Usualmente schedulazione pre-emptive a priorità fisse

### Coda a più livelli C+L (2)



### Coda a più livelli con retroazione C+LR (1)

- Coda a più livelli che permette ai processi di migrare da un livello a quello adiacente



### Coda a più livelli con retroazione C+LR (2)

- Code di attesa separate in funzione dell'uso dinamico del processore da parte dei processi
- Algoritmo di schedulazione specifico per ciascuna coda di attesa
- Politica di promozione
- Politica di degradazione
- Politica di allocazione



### **Schedulazione in sistemi multiprocessore (1)**

**La schedulazione deve considerare le caratteristiche dell'architettura del sistema di elaborazione:**

- processori
  - omogenei ✓
  - eterogenei ✓
- memoria
  - solo condivisa
  - anche locale
- periferiche
  - accessibili da singolo processore
  - accessibili da tutti i processori

### **Schedulazione in sistemi multiprocessore (2)**

- Sistemi con processori omogenei, memoria solo condivisa e periferiche accessibili da tutti i processori:
  - Coda unica
  - Una coda per processore in memoria condivisa (suddivisione del carico - load sharing)
- Sistemi con processori omogenei, memoria anche locale e periferiche accessibili da tutti i processori:
  - Coda unica
  - Una coda per processore in memoria condivisa o in memoria locale (allocazione dei processi ai processori e load sharing)

### **Schedulazione in sistemi multiprocessore (3)**

- Sistemi con processori omogenei, memoria solo condivisa e periferiche accessibili da alcuni processori:
  - Una coda per processori omogenei in memoria condivisa, una coda per ogni processore che gestisce una specifica periferica
- Sistemi con processori omogenei, memoria anche locale e periferiche accessibili da alcuni processori:
  - Una coda per processori omogenei in memoria condivisa, una coda per ogni processore che gestisce una specifica periferica
  - Una coda per ognuno dei processori omogenei in memoria locale, una coda per ogni processore che gestisce una specifica periferica
- Sistemi con processori eterogenei:
  - Una coda per processore
  - Una coda per gruppi di processori omogenei

### **Schedulazione in sistemi multiprocessore (4)**

#### **Tipi di multiprocessing:**

- **Multiprocessing asimmetrico:**
  - Processore master esegue sistema operativo (schedulazione di tutti i processi per tutti i processori)
  - Processori slave eseguono solo processi applicativi
- **Multiprocessing simmetrico:**
  - Ogni processore esegue il sistema operativo (schedulazione dei processi assegnati al processore) e i processi applicativi

## In sintesi

- Politiche e algoritmi di schedulazione
  - First Come, First Served
  - Shortest Job First
  - Priorità
  - Round Robin
  - Coda a più livelli
  - Coda a più livelli con retroazione
- Schedulazione in sistemi multiprocessore

