

W4 D2 - 15 07 2025 - Giuseppe Gigliotti - Relazione Scheduling della CPU

L'esercizio di oggi verte sui meccanismi di pianificazione dell'utilizzo della CPU (o processore). In ottica di ottimizzazione della gestione dei processi, abbiamo visto come lo scheduler si sia evoluto nel tempo per passare da approccio mono-tasking ad approcci multi-tasking.

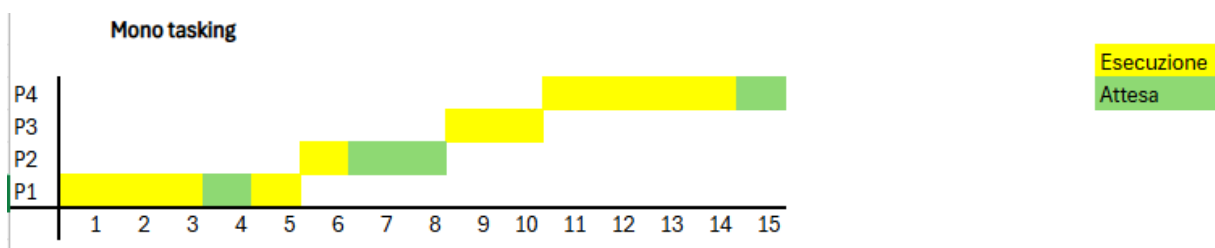
Traccia: Si considerino 4 processi (P1, P2, P3, P4) con i tempi di esecuzione e di attesa input/output dati in tabella. I processi arrivano alle CPU in ordine P1, P2, P3, P4. Individuare il modo più efficace per la gestione e l'esecuzione dei processi, tra i metodi già visti a lezione. Abbozzare un diagramma che abbia sulle ascisse il tempo passato da un istante «0» e sulle ordinate il nome del Processo. Processo Tempo di esecuzione Tempo di attesa Tempo di esecuzione dopo.

Processo	Tempo di esecuzione	Tempo di attesa	Tempo di esecuzione dopo attesa
P1	3 secondi	1 secondo	1 secondo
P2	1 secondo	2 secondi	-
P3	2 secondi	-	-
P4	4 secondi	1 secondo	-

Mono tasking:

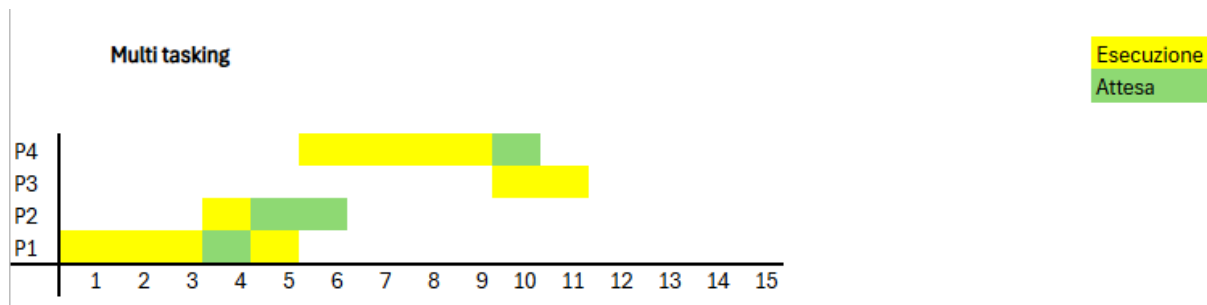
Si dicono mono-tasking, quei sistemi che non supportano l'esecuzione parallela di più processi. Nei sistemi mono-tasking, non è possibile sospendere l'esecuzione di un processo per assegnare la CPU ad un secondo processo. I sistemi operativi mono-tasking risultano essere piuttosto inefficienti, a causa dei periodi di inutilizzo frequenti della CPU, come da grafico sotto.

L'inefficienza dei sistemi mono-tasking sta nel fatto che la CPU passa una percentuale non trascurabile del suo tempo in attesa di eventi esterni, senza compiere nessuna azione.



Multi tasking:

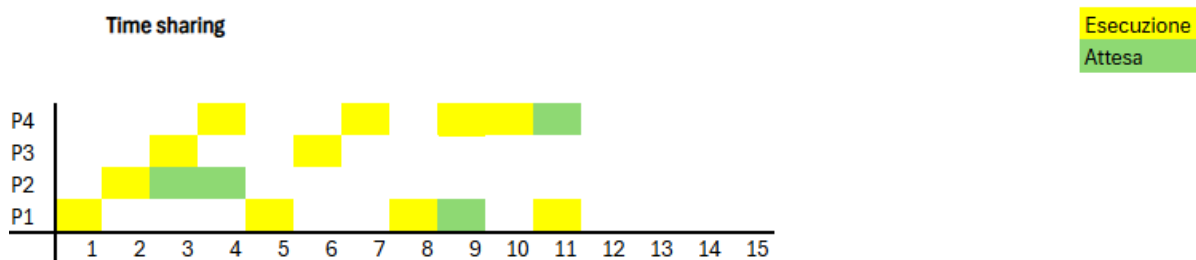
I sistemi operativi che permettono l'esecuzione contemporanea di più programmi sono detti, invece, multi-tasking. Nei sistemi multi-tasking, la pianificazione con prelazione (preemptive multitasking) fa in modo che quando un processo è in attesa di eventi esterni, la CPU possa essere impiegata per altro. Così, quando il processo A passerà dallo stato di esecuzione allo stato di attesa, la CPU potrà essere impiegata per eseguire le istruzioni del processo B.



Time sharing:

Infine, un'evoluzione dei sistemi multi-tasking sono i sistemi time-sharing. In un sistema time-sharing ogni processo viene eseguito in maniera ciclica per piccole porzioni di tempo che prendono il nome di "quanti". In presenza di una CPU con velocità sufficientemente elevata, il sistema time-sharing darà l'impressione di un'evoluzione parallela dei processi.

Nel nostro caso, supponiamo di avere 1s di quanto



FACOLTATIVO:

Considerare un insieme di cinque processi P1, P2, P3, P4, P5 con i seguenti tempi di arrivo e di esecuzione (in millisecondi):

Processo	Tempo di arrivo (t_0)	Tempo di esecuzione (T_x)
P1	0	14
P2	30	16
P3	6	40
P4	46	26
P5	22	28

- Descrivere lo scheduling di questi processi con politica Round Robin (time slice di 12 millisecondi).

Round Robin:

Il Round Robin è un algoritmo di **scheduling preemptive** che assegna a ogni processo un tempo fisso chiamato **quantum** (o time slice).

Come funziona:

1. Quantum (Time Slice)

- Ogni processo può eseguire per un tempo massimo = quantum
- Se il processo finisce prima del quantum passa al prossimo processo
- Se il quantum scade il processo viene **interrotto** e rimesso in coda

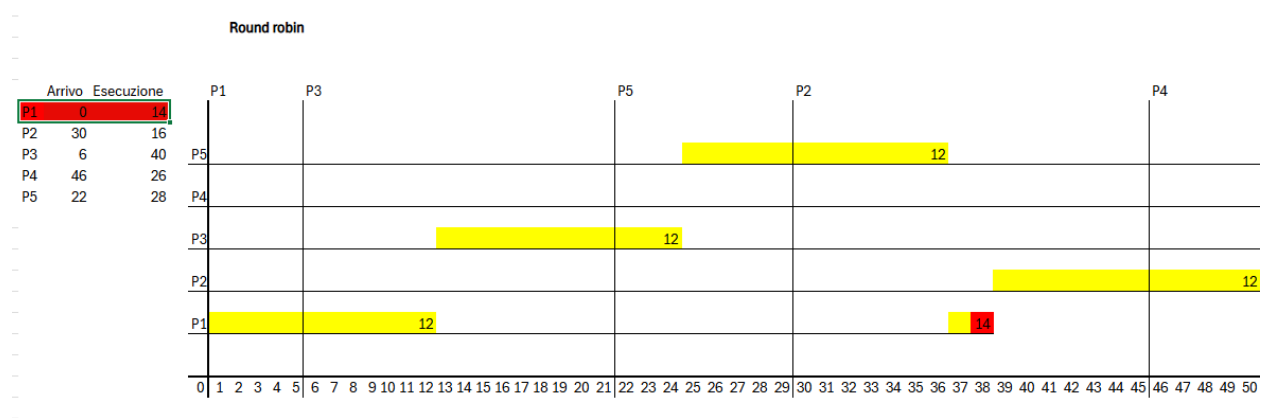
2. Coda FIFO (First In, First Out)

- I processi pronti vengono messi in una coda circolare
- Si prende sempre il primo processo dalla coda
- Se viene interrotto, va in fondo alla coda

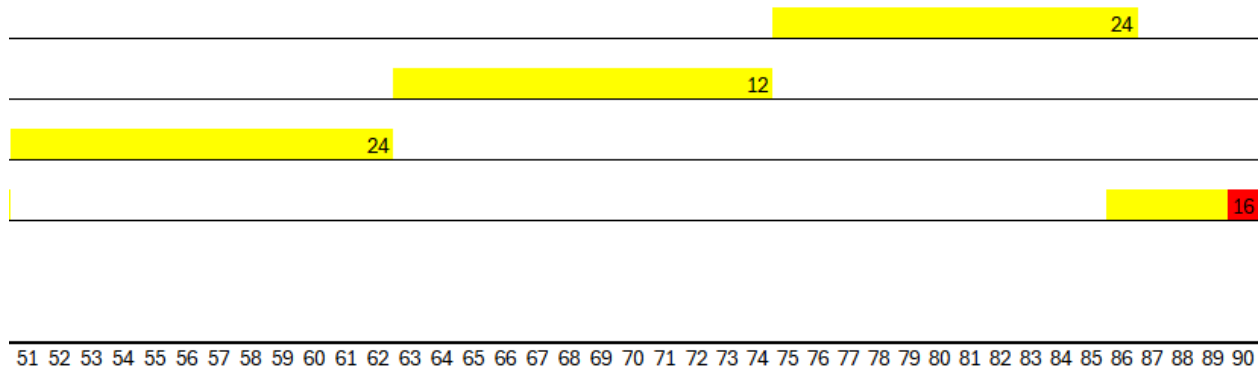
3. Preemption

- I processi possono essere interrotti quando il quantum scade
- Il sistema operativo riprende il controllo e passa al processo successivo

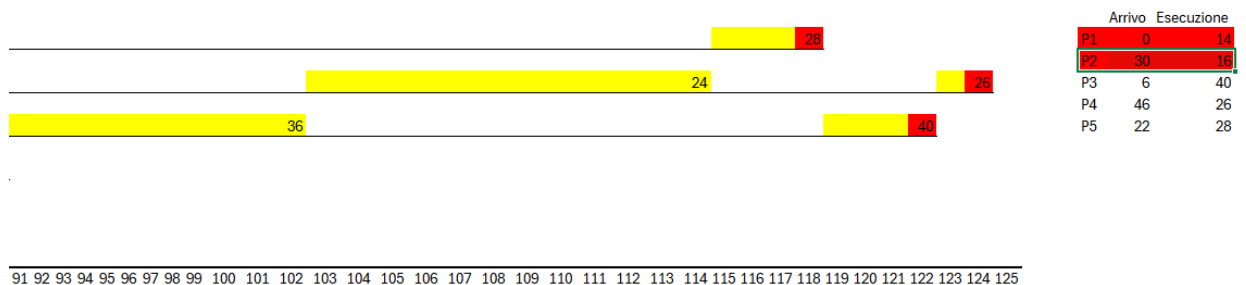
Partiamo il con il P1 essendo che il suo tempo di arrivo è 0



Dopo che P1 è terminato continuamo con la sequenza, il prossimo che terminerà sarà il P2



Continuando la sequenza alla fine i processi termineranno nell'ordine P5, P3, P4.



- Calcolare i tempi di attesa e di turnaround (durata) medi.

