

## Esercizio S3L1 - CPU Monotasking, Multitasking e Timesharing

La lezione di oggi riguardava l'utilizzo dei sistemi operativi e il loro funzionamento. Procedendo per il passaggio logico di hardware, memorie e funzionamento dei firmware, siamo arrivati al funzionamento dei processori (CPU). Abbiamo posto particolare attenzione alla differente gestione dei processi da parte del sistema operativo attraverso i metodi monotasking, multitasking e timesharing.

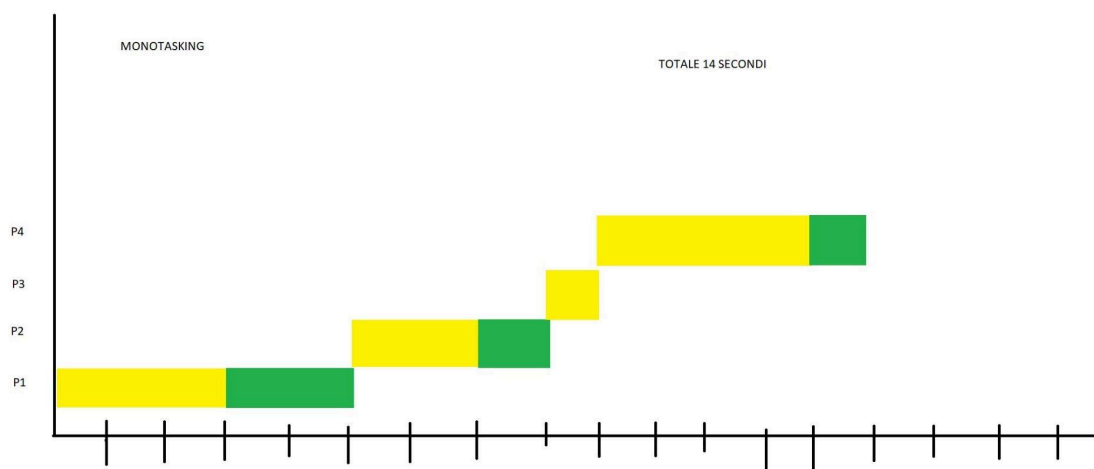
L'esercizio di oggi verte sui meccanismi di pianificazione dell'utilizzo della CPU (o processore). In ottica di ottimizzazione della gestione dei processi, abbiamo visto come lo scheduler si sia evoluto nel tempo per passare da approccio mono-tasking ad approcci multi-tasking.

### Traccia:

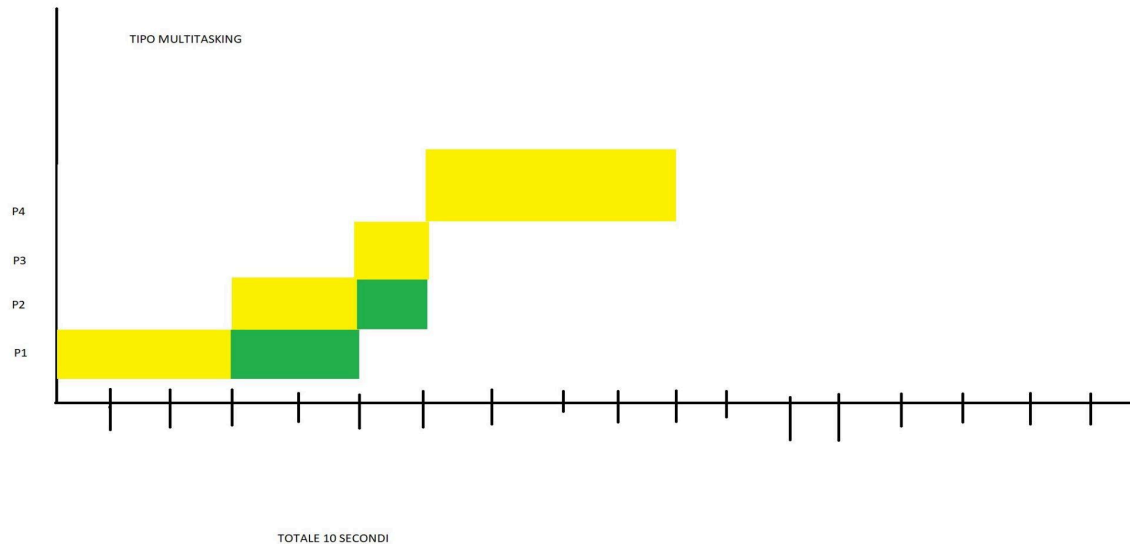
Si considerino 4 processi, che chiameremo P1,P2,P3,P4, con i tempi di esecuzione e di attesa input/output dati in tabella. I processi arrivano alle CPU in ordine P1,P2,P3,P4. Individuare il modo più efficace per la gestione e l'esecuzione dei processi, **tra i metodi visti nella lezione teorica**. Abbozzare 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 attesa
P1	3 secondi	2 secondi	1 secondo
P2	2 secondi	1 secondo	-
P3	1 secondi	-	-
P4	4 secondi	1 secondo	-

Il monotasking svolge i processi uno per uno incluse le attese che possono essere dettate dal processo stesso o da eventi esterni riguardanti il funzionamento del sistema, e si rivela essere il meno efficiente fra i tre. Dal momento che vengono svolti uno per uno, un processo come quello della consegna dell'esercizio (di cui qui sopra lo screenshot), impiegherebbe un totale di 14 secondi ovvero la somma di tutti i tempi di esecuzione e attesa. Qui sotto un prospetto di come potrebbe funzionare un processo multitasking.



Il multitasking invece approfitta delle pause per impiegare la CPU in maniera tale che non abbia mai un momento di “riposo” attraverso una pianificazione preventiva degli eventi dei processi stessi, e taglia così i tempi di svolgimento a 10 secondi totali, per un risparmio temporale all’incirca del 30%



Il timesharing invece, che è un’evoluzione diretta del multitasking, divide i tempi dell’esecuzione dei processi in quanti e tutti vengono svolti in maniera ciclica in maniera tale che invece di essere in coda uno dietro l’altro diano l’impressione di essere svolti simultaneamente e procedano in pari durante l’esecuzione. E’ bene notare che l’efficientamento di questo metodo è direttamente proporzionale alla velocità della CPU, che permette di evitare che lo spostamento fra i processi e l’esecuzione degli stessi non sembri un’esecuzione parallela ma piuttosto una versione rallentata di un processo multitasking.



Nonostante non ci sia un vero e proprio risparmio di tempo, la fluidità del processo timesharing si rivela la soluzione più avanzata e comoda per l’utente, CPU permettendo.