

## REPORT S3/L1

### Pianificazione dell'utilizzo della CPU

#### 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 alla 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 secondo	2 secondo	1 secondo
P2	2 secondo	1 secondo	-
P3	1 secondo	-	-
P4	4 secondo	1 secondo	-

#### Svolgimento

In questo scenario, consideriamo quattro processi, chiamati P1, P2, P3 e P4, che arrivano alla CPU nell'ordine specificato e hanno i loro rispettivi tempi di esecuzione e tempi di attesa I/O. La scelta del modo più efficace per la gestione e l'esecuzione di questi processi dipende dalla natura del sistema operativo e dal tipo di applicazione che gestisce questi processi.

Le tre modalità di gestione dei processi sono:

1. *Mono-tasking*: Il sistema esegue un solo processo alla volta.
2. *Multi-tasking*: Il sistema esegue più processi simultaneamente, permettendo al processore di passare da un processo all'altro.
3. *Time-sharing*: È una forma avanzata di multi-tasking in cui il processore è diviso in intervalli di tempo (chiamati "quantum") e ogni processo ha diritto a usare la CPU per un breve periodo.

Per determinare quale modalità sia la più efficace, è importante considerare:

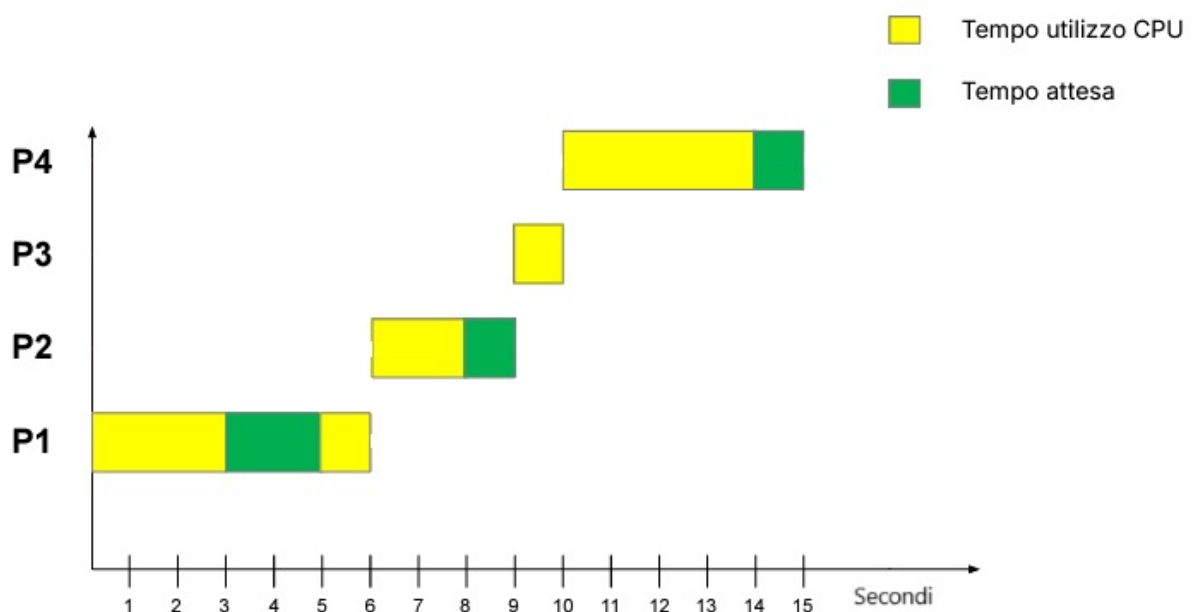
- I tempi di esecuzione di ciascun processo.
- I tempi di attesa, che potrebbero implicare che i processi siano in attesa di risorse e quindi non occupino la CPU durante quel periodo.

## Analisi dei Metodi

### Mono-tasking

Nel **mono-tasking**, viene eseguito solo un processo alla volta. Questo è il modello più semplice e in genere viene utilizzato nei sistemi più vecchi o in sistemi con risorse hardware limitate. In questa modalità, i processi vengono eseguiti uno dopo l'altro, e la CPU non può passare tra di essi.

- **Vantaggi:** La gestione è semplice e la CPU è completamente dedicata a un singolo processo.
- **Svantaggi:** Se un processo è in attesa di I/O, la CPU rimane inattiva durante quel periodo. Inoltre, se un processo ha una lunga durata di esecuzione, i tempi di attesa degli altri processi aumentano.



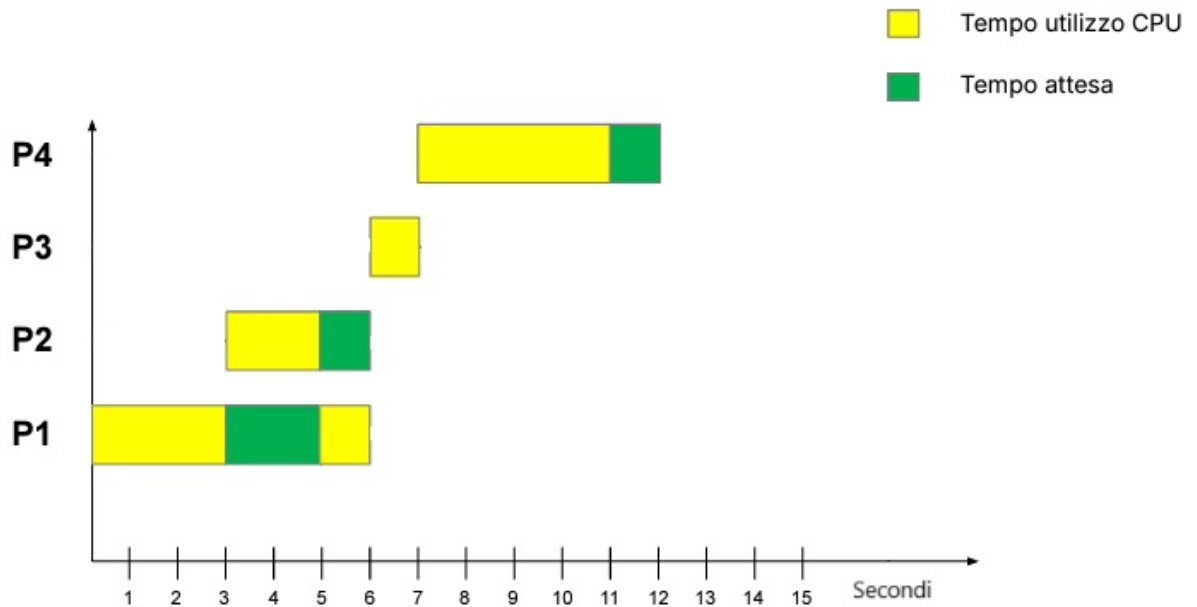
### Multi-tasking

Nel **multi-tasking**, più processi possono essere eseguiti contemporaneamente, ma solo uno alla volta occupa la CPU. In questa modalità, il sistema operativo gestisce la "commutazione" tra i processi. Ogni processo può essere eseguito per una parte del tempo, mentre gli altri sono in attesa di CPU o di I/O.

Nel nostro caso, se un processo è in attesa di I/O, la CPU può essere utilizzata da un altro processo. Quindi, se un processo è in attesa di I/O (ad esempio P2 con attesa di 2 secondi), la CPU può essere data ad altri processi nel frattempo.

Esecuzione con **multi-tasking**:

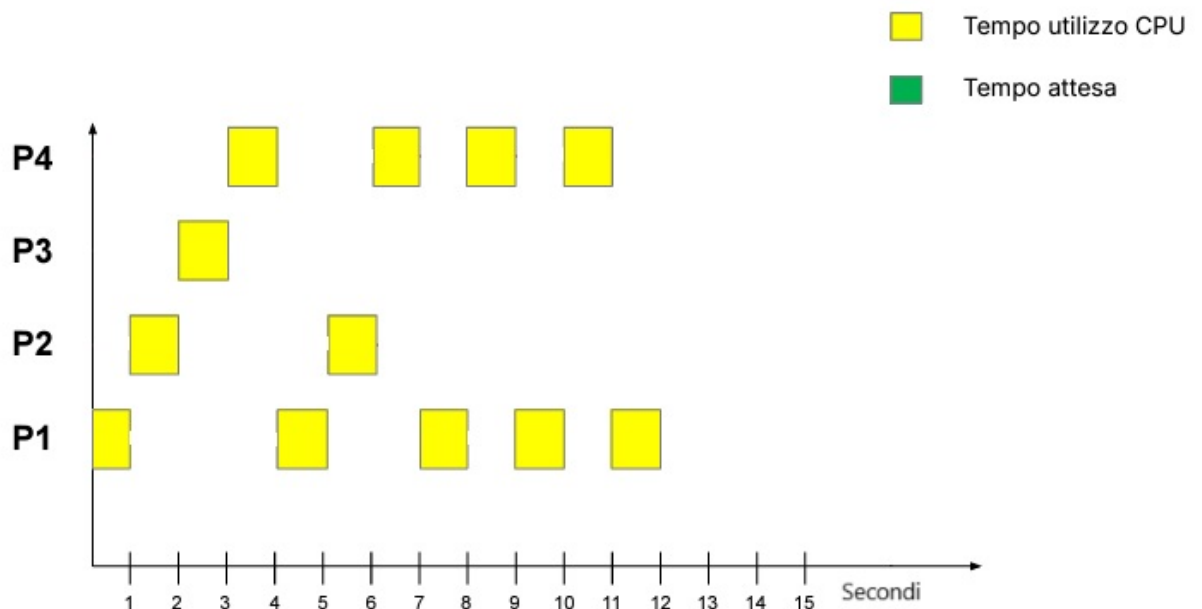
- Quando un processo è in attesa di I/O, la CPU può essere passata ad altri processi.
- Se P2 è in attesa di I/O, la CPU potrebbe essere data a P1 o P3.



### *Time-sharing*

Il **time-sharing** è una forma avanzata di **multi-tasking** in cui i processi vengono eseguiti per periodi di tempo limitati, chiamati "quantum". Ogni processo ottiene un breve periodo di esecuzione (ad esempio, 1-2 unità di tempo), e la CPU passa da un processo all'altro in modo regolare.

Questo approccio è molto utile nei sistemi in cui è necessaria una risposta rapida, poiché riduce il tempo di attesa di ciascun processo. Se un processo non finisce nel suo "quantum", la CPU passa al prossimo processo. Inoltre, se un processo è in attesa di I/O, il sistema può facilmente passare al prossimo processo senza sprechi di tempo.



## Qual è il metodo più efficace?

- **Mono-tasking** è inefficace per la gestione di più processi, specialmente se ci sono processi con lunghe attese I/O. In questo caso, la CPU potrebbe rimanere inattiva troppo a lungo.
- **Multi-tasking** è una soluzione migliore rispetto al mono-tasking, in quanto consente di utilizzare la CPU anche quando alcuni processi sono in attesa di I/O. Tuttavia, la gestione dei processi potrebbe non essere altrettanto ottimale come nel caso del **time-sharing**.
- **Time-sharing** è il metodo più efficiente per un sistema che deve gestire più processi, poiché ottimizza l'uso della CPU permettendo a ciascun processo di avere una porzione di tempo di esecuzione regolare e riducendo al minimo i tempi di attesa per ciascun processo. Inoltre, se i processi sono in attesa di I/O, il sistema può facilmente passare al prossimo processo senza perdere tempo.

## Conclusione

Nel caso di 4 processi con tempi di esecuzione e attese I/O differenti, **il metodo più efficace** per la gestione e l'esecuzione dei processi è **Time-sharing**, poiché garantisce che ogni processo ottenga tempo CPU in modo regolare e minimizza i tempi di attesa complessivi.