

Lezione S3/L1

Metodologie calcolo per sistemi operativi

L'esercizio di oggi richiede la realizzazione di 3 grafici per meccanismi di pianificazione dell'utilizzo della CPU (o processore). I 3 grafici dovranno mostrare 3 diversi approcci:

- **Mono-tasking**
- **Multi-tasking**
- **Time-sharing**

ciascun grafico sarà fatto in funzione del tempo e ci permetterà di osservare in maniera chiara eventuali tempi di attesa.

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	-

tabella 1: i processi

Mono-tasking:

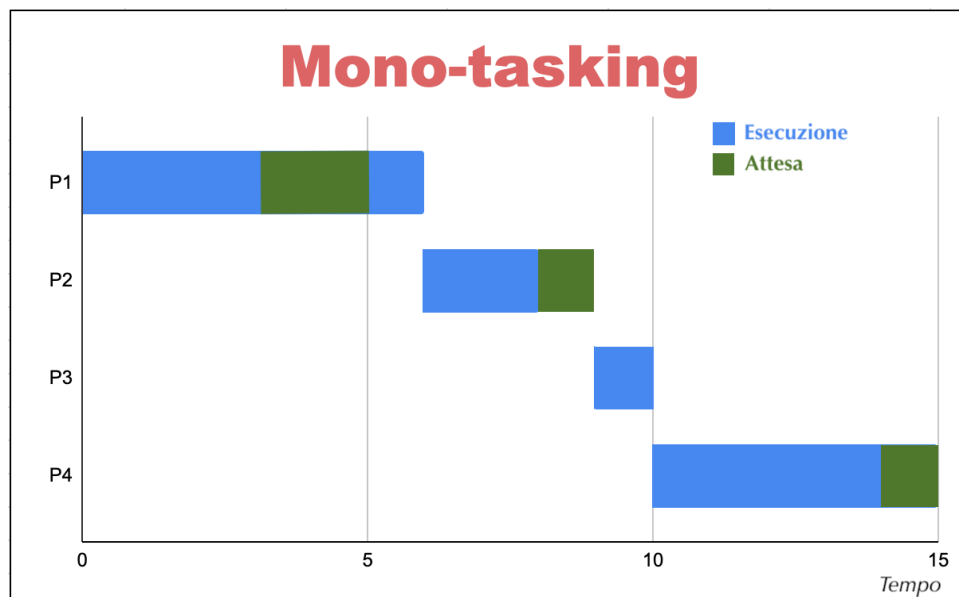


figura 1: grafico mono-tasking.

Dalla figura 1 siamo in grado di osservare che il tempo totale di runtime è pari a **15** secondi, di questi 15 secondi , 4 sono tempi dove la cpu non compie alcun tipo di lavoro quindi ha un'efficienza pari al **73,3%**, alquanto bassa.

In un sistema mono-tasking, la CPU si concentra su un solo processo alla volta, eseguendolo dall'inizio alla fine prima di passare a un altro. Il mono-tasking riduce la complessità gestionale e i conflitti tra processi, garantendo che ciascun processo riceva tutta la potenza di calcolo disponibile. È utile in applicazioni dedicate e in ambienti con risorse limitate.

Multi-tasking:

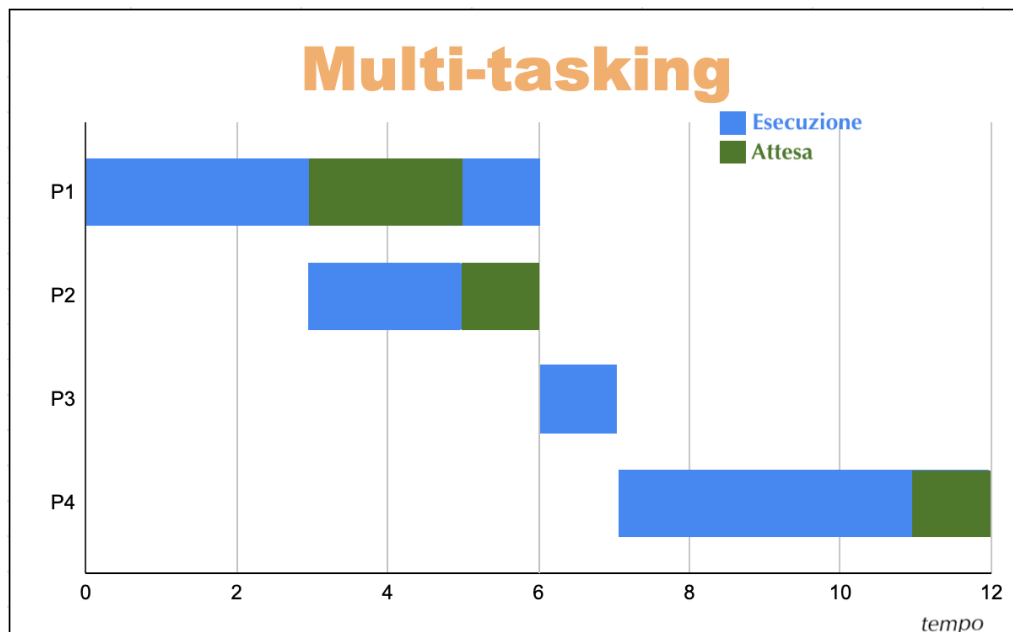


figura 2: grafico multi-tasking.

Dalla figura 2 siamo in grado di osservare che il tempo totale di runtime è pari a **12** secondi, di questi 12 secondi , 1 secondo è il tempo dove la cpu non compie alcun tipo di lavoro quindi ha un'efficienza pari al **91,7%**, con un'ottima performance.

Nel multitasking, la CPU gestisce più processi apparentemente contemporaneamente, alternandosi rapidamente tra di essi. Questo approccio aumenta l'efficienza, sfruttando al meglio i tempi di inattività del processore. Il multitasking è essenziale nei sistemi operativi moderni, permettendo agli utenti di eseguire più applicazioni simultaneamente. Tuttavia, richiede un'accurata gestione delle risorse e della memoria per evitare conflitti e garantire che tutti i processi ricevano attenzione sufficiente. Sebbene più complesso rispetto al mono-tasking, il multitasking è fondamentale per garantire prestazioni elevate e flessibilità nei sistemi moderni.

Time-sharing:

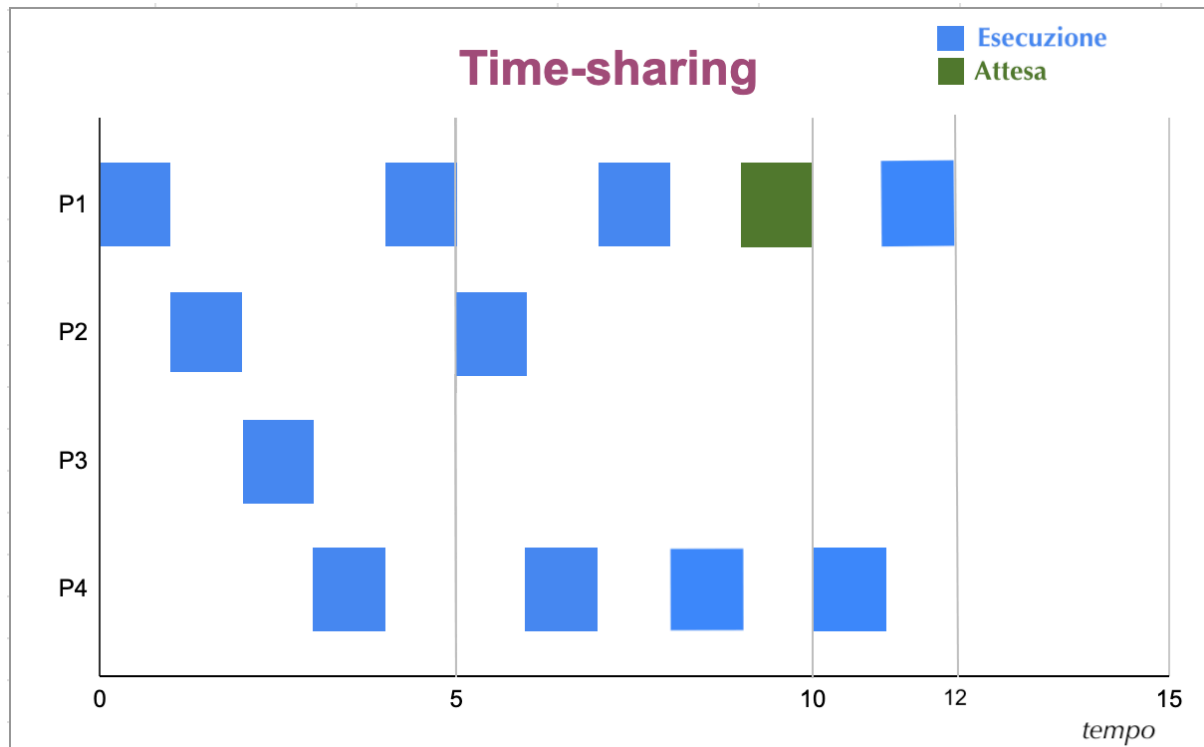


figura 3: grafico time-sharing.

Dalla figura 3 siamo in grado di osservare che il tempo totale di runtime è pari a **12** secondi, di questi 12 secondi, 1 sono tempi dove la cpu non compie alcun tipo di lavoro quindi ha un'efficienza pari al **91,6%**, come per il multi-tasking.

Il *time-sharing* è una tecnica di gestione che consente alla CPU di alternare rapidamente l'esecuzione di più processi, dando a ciascuno una "fetta" di tempo di CPU. In pratica, il sistema operativo suddivide il tempo disponibile in piccoli intervalli, detti *quantum* (*1 secondo in questo caso*), e assegna questi intervalli ai vari processi in esecuzione. Ogni processo viene interrotto quando scade il suo quantum, e la CPU passa al successivo. Questa modalità permette a più utenti o applicazioni di condividere il processore senza che un singolo processo monopolizzi l'intero sistema, migliorando l'efficienza e la reattività del sistema. Il time-sharing è cruciale nei sistemi multiutente e nelle applicazioni in cui è necessario gestire più attività contemporaneamente.

Nota bene: nel time sharing i tempi di attesa vengono "svolti" anche se la cpu agisce su altri blocchi, questo non significa che vengono eliminati bensì vengono svolti se il tempo tra un'esecuzione e l'altra lo consente, infatti tra il secondo 8-10 il processo P1 ha un tempo di attesa pari a due secondi ma lo spazio disponibile era solo per 1 secondo quindi al 10 secondo la cpu è ferma ed attende.