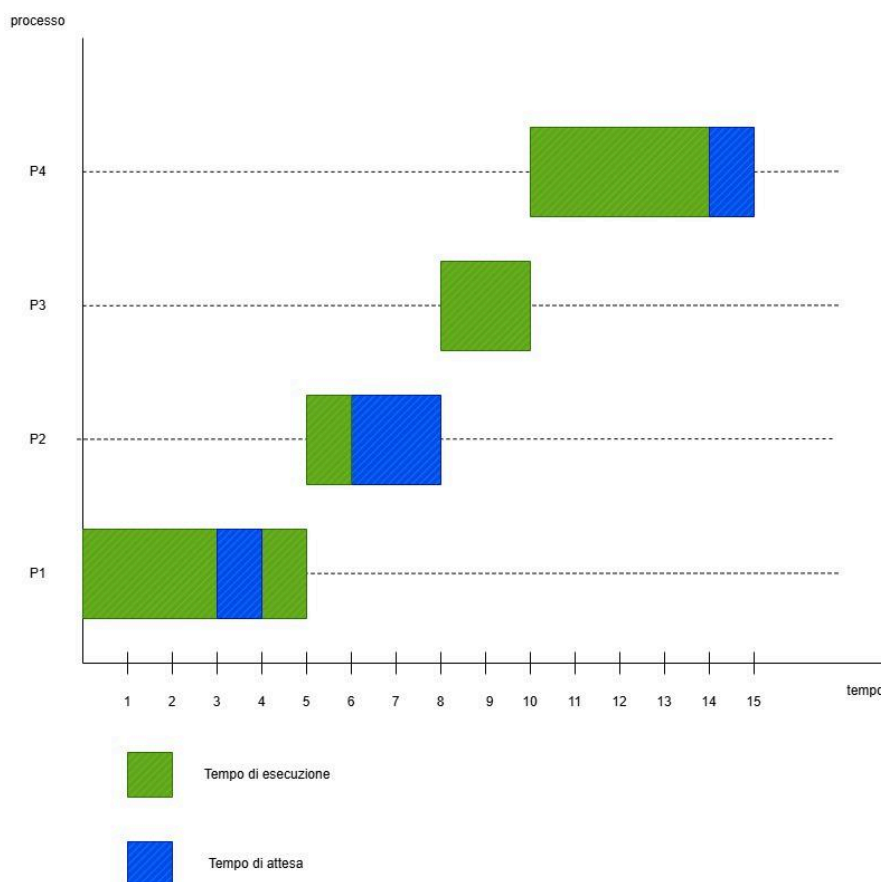


INTRODUZIONE

In questa esercitazione analizziamo tre modalità di ottimizzazione dei processi gestiti dallo scheduler della CPU: mono-tasking, multi-tasking e time-sharing.

L'obiettivo è determinare quale di questi metodi risulti più efficiente nella gestione ed esecuzione contemporanea di più processi. Per ogni modalità, sono stati considerati quattro processi (P1, P2, P3, P4), analizzandone i tempi di esecuzione e di attesa. I risultati sono stati rappresentati graficamente mediante diagrammi con il tempo sull'asse X e il numero del processo sull'asse Y.

MONO-TASKING

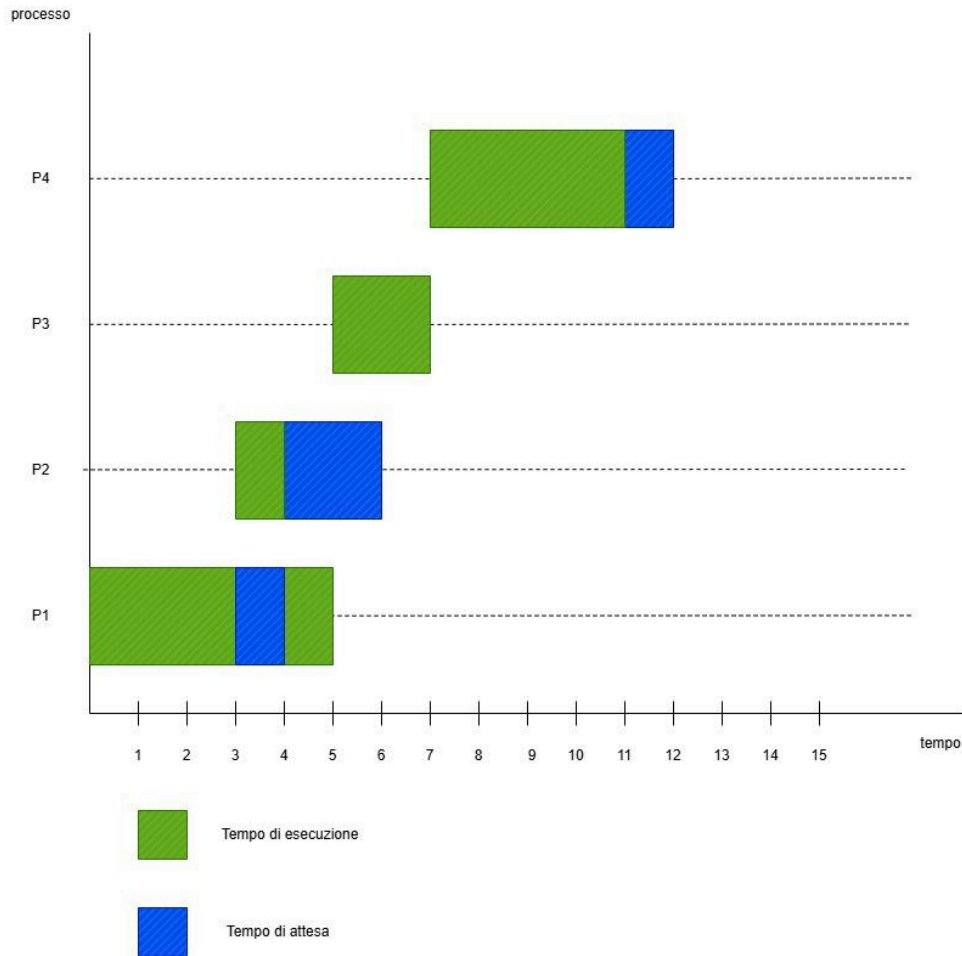


La prima prova di ottimizzazione, con sistema monotasking, ci mostra come ogni processo deve attendere il totale completamento del precedente per poter proseguire.

È comodo e semplice, perché gestisce un processo alla volta senza complicare lo scheduling ed è estremamente lineare.

Questo significa però che la macchina non sembrerà elaborare tutti i processi assieme, ma uno alla volta, risultando molto lenta. Anche durante i tempi di attesa di un processo, il sistema mono-tasking non permette di elaborare il processo successivo, risultando in una gestione poco dinamica. Come possiamo notare lo scheduler impiega ben 15 secondi per completare i 4 processi.

MULTI-TASKING

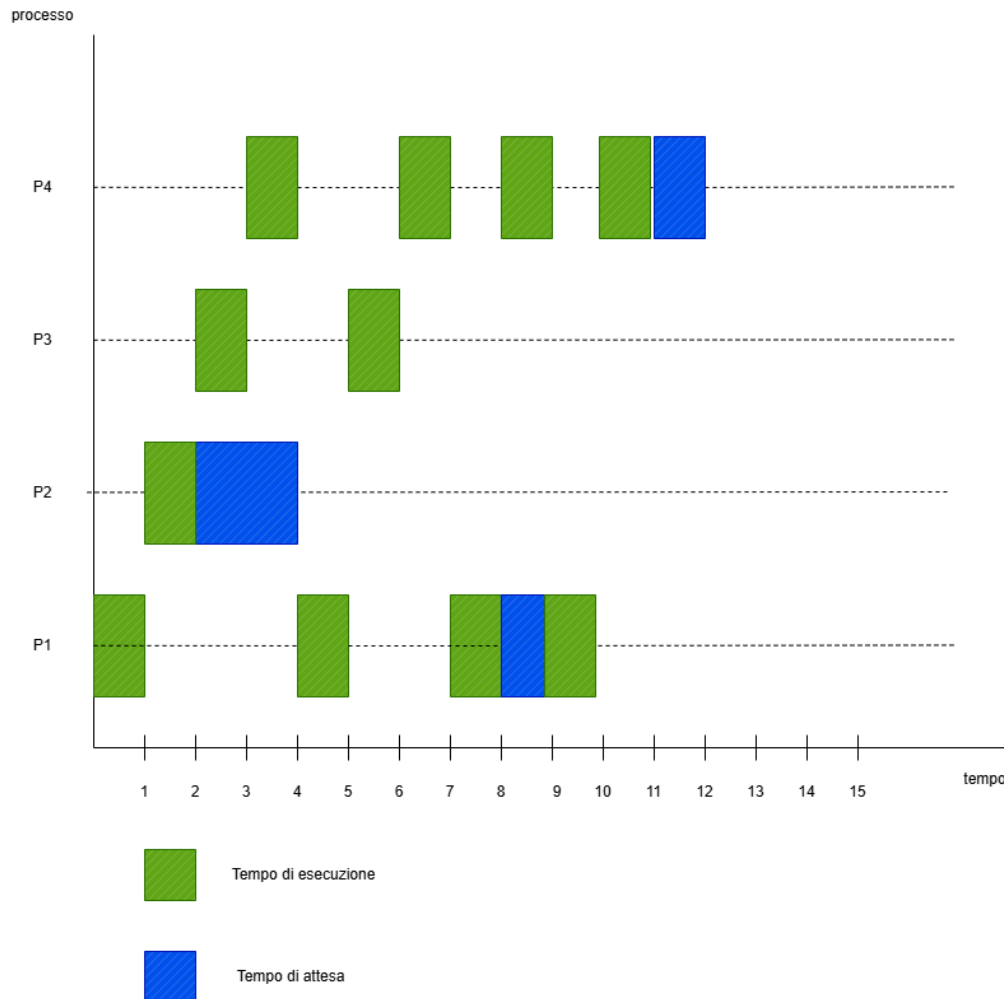


La seconda prova di ottimizzazione, con sistema multitasking, mostra un miglioramento. L'approccio precedente contava i tempi di attesa del processo X come parte dello stesso processo, mentre il multi-tasking permette di sfruttarli per iniziare il processo successivo e ottimizzare i tempi, nell'attesa che il processo prima torni in esecuzione. In questo modo possiamo ridurre i tempi di attesa complessivi, poiché i processi non devono attendere il completamento totale dei precedenti per iniziare la propria esecuzione.

Lo scheduling diventa più complesso rispetto al monotasking, ma offre una maggiore dinamicità nel sistema, creando l'impressione che i processi siano eseguiti in parallelo anche se in realtà non è proprio così.

A differenza del mono-tasking, lo scheduler impiega 12 secondi per completare i 4 processi, risparmiando 3 secondi rispetto.

TIME-SHARING



La terza prova sfrutta il modello di time-sharing, in cui la CPU assegna a ciascun processo un piccolo intervallo di tempo (quantum) di 1 secondo, prima di passare al successivo. Questo permette una rotazione regolare tra tutti i processi, migliorando la distribuzione del tempo e dando l'idea di una evoluzione parallela degli stessi.

A livello di tempo totale il sistema impiega 12 secondi, esattamente come nella modalità multi-tasking. Tuttavia la gestione è diversa: mentre il multi-tasking può assegnare più tempo consecutivo a un singolo processo, dando l'impressione di un'attesa maggiore, nel time-sharing ogni processo viene eseguito in piccole porzioni, alternate regolarmente. Anche se il tempo finale sarà lo stesso, l'impressione di un sistema in grado di fare più cose in contemporanea sarà più evidente.

Il time-sharing si dimostra ancora più efficace in scenari dove è importante garantire che tutti i processi abbiano accesso equo alla CPU.

CONCLUSIONE

Confrontando i tre approcci, è evidente come il monotasking, pur essendo il più semplice, risulti anche il meno efficiente: gestisce un solo processo alla volta e non sfrutta i tempi di inattività, portando a un tempo di completamento più elevato (15 secondi). Il multitasking rappresenta un netto miglioramento, riducendo i tempi morti grazie alla possibilità di alternare i processi durante le fasi di attesa, con un tempo totale di 12 secondi. Di contro però non garantisce una distribuzione uniforme della CPU.

Il modello di time-sharing, pur mantenendo lo stesso tempo totale del multitasking (12 secondi), è più equo e bilanciato: ogni processo riceve un intervallo regolare di CPU di 1 secondo, migliorando la reattività dello scheduling e l'equità nella gestione delle risorse.

In definitiva, il monotasking può andar bene per sistemi semplici, avendo una gestione più lenta e lineare, il multitasking è invece adatto per ottimizzare i tempi e avere una schedulazione dei processi più rapida, mentre il metodo time sharing

Il multitasking è adatto quando serve ottimizzare i tempi e avere una macchina in grado di fare più operazioni "contemporaneamente" mentre il time-sharing è utile in ambienti multiutente dove è necessaria una gestione esattamente equa della CPU e abbastanza rapida da sembrare contemporanea, o comunque con sistemi e applicazioni che richiedono un'ottima reattività dei processi.