

CS0424IT — ESERCITAZIONE S3L1 - SISTEMI OPERATIVI

Simone La Porta



TRACCIA

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. 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 t_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 secondo	-	-
P4	4 secondi	1 secondo	-

Tabella 1: Tempi di esecuzione e attesa dei processi

SVOLGIMENTO

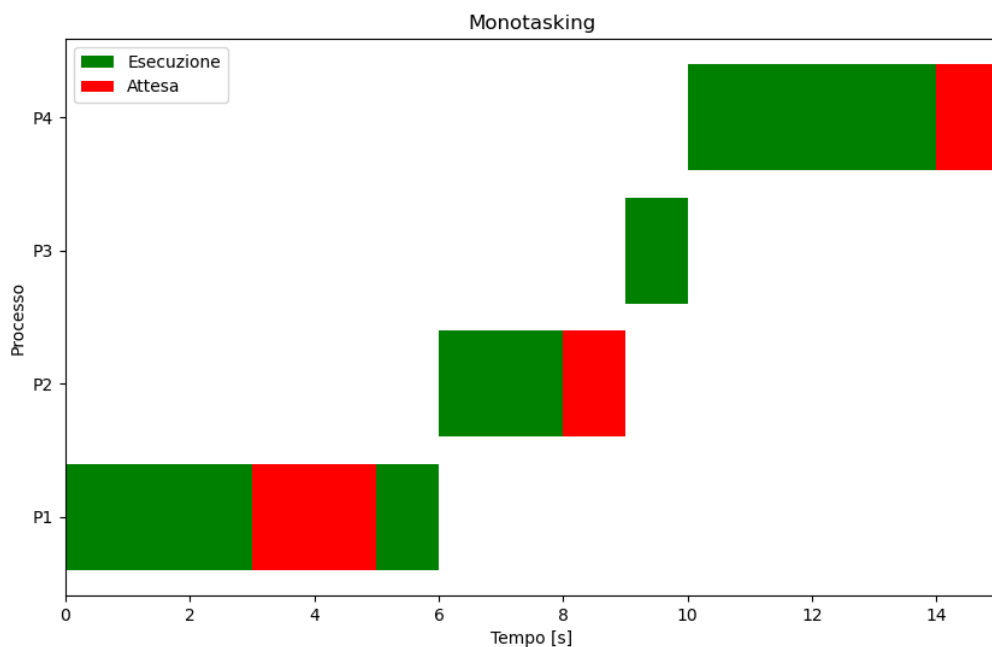
La pianificazione della CPU è un concetto fondamentale nei sistemi operativi e riguarda l'allocazione del tempo della CPU ai vari processi. Questo è cruciale per l'esecuzione efficiente

dei processi, garantendo un uso ottimale della CPU. Esistono diversi algoritmi di pianificazione, ciascuno con i propri vantaggi e svantaggi.

Algoritmi di pianificazione monotasking

Tra i principali algoritmi di pianificazione monotasking si ha il **First-Come, First-Served (FCFS)**:

- **First-Come, First-Served (FCFS):**
 - **Descrizione:** I processi vengono eseguiti nell'ordine in cui arrivano. Un processo deve completare la sua esecuzione prima che il prossimo processo possa iniziare.
 - **Pro:** Semplice da implementare.
 - **Contro:** Può portare all'effetto "convoglio", dove i processi più brevi aspettano che quelli più lunghi finiscano, risultando in un tempo medio di risposta scadente.



Un sistema operativo monotasking, come mostrato nel grafico, è un sistema che consente l'esecuzione di un solo processo alla volta, senza la possibilità di sospenderlo per assegnare la CPU ad un altro. Questo può portare a un'esperienza utente più semplice e meno esigente

in termini di risorse del computer, ma la CPU passa una percentuale non trascurabile del suo tempo in attesa di eventi esterni, senza compiere nessuna azione, limitandone fortemente la produttività’.

Algoritmi di pianificazione multitasking

Gli algoritmi multitasking o del time sharing, permettono a più processi di condividere la CPU, spesso preemptively.

1. Shortest Job Next (SJN):

- **Descrizione:** Viene selezionato il processo con il tempo di esecuzione più breve.
- **Pro:** Può ridurre il tempo medio di attesa.
- **Contro:** Richiede una conoscenza precisa dei tempi di esecuzione, spesso impraticabile.
- **Nota:** Anche se SJN non è esplicitamente preemptive, in pratica è usato in ambienti multitasking.

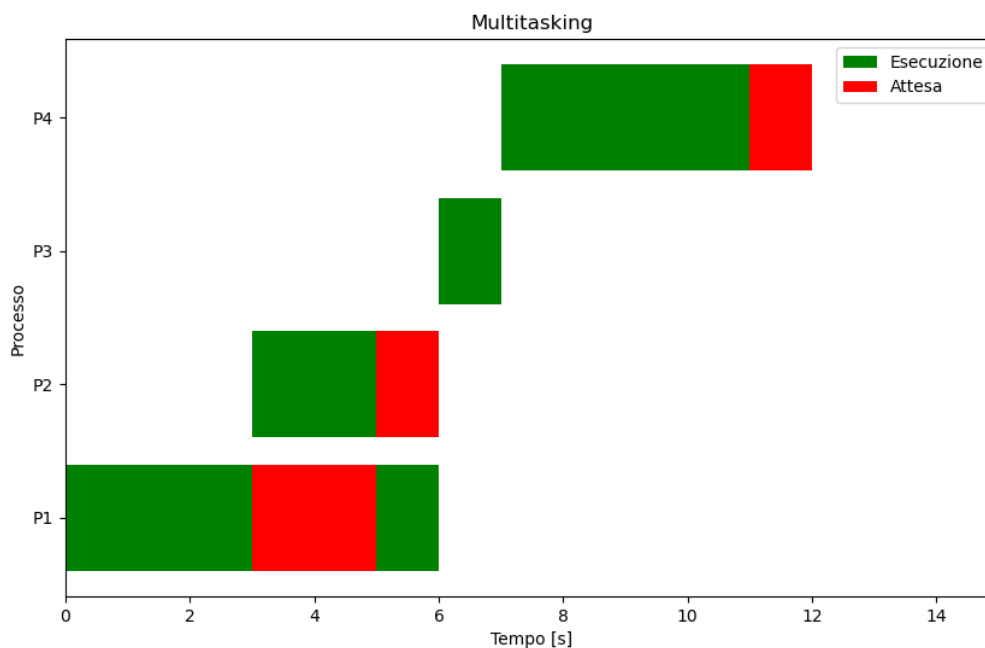
2. Round-Robin (RR):

- **Descrizione:** Ogni processo viene eseguito per un tempo predefinito (quantum). Se un processo non termina entro il suo quantum, va in fondo alla coda.
- **Pro:** Garantisce equità e riduce il rischio di affamamento dei processi.
- **Contro:** Le prestazioni dipendono molto dalla scelta del quantum.

3. Multitasking (Preemptive):

- **Descrizione:** I processi possono essere interrotti e ripresi, permettendo a più processi di fare progressi. La CPU può passare da un processo all’altro, soprattutto se un processo è in attesa di operazioni I/O.
- **Pro:** Migliora l’utilizzo della CPU e la reattività.
- **Contro:** Sovraccarico di commutazione di contesto (context switch).

Un sistema operativo multitasking è un sistema che ha la capacità di eseguire più processi o attività simultaneamente, sfruttando gli eventuali tempi di attesa. Il multitasking è raggruppabile in due categorie: Prelazione e Cooperativo. Nel grafico è mostrata la pianificazione con



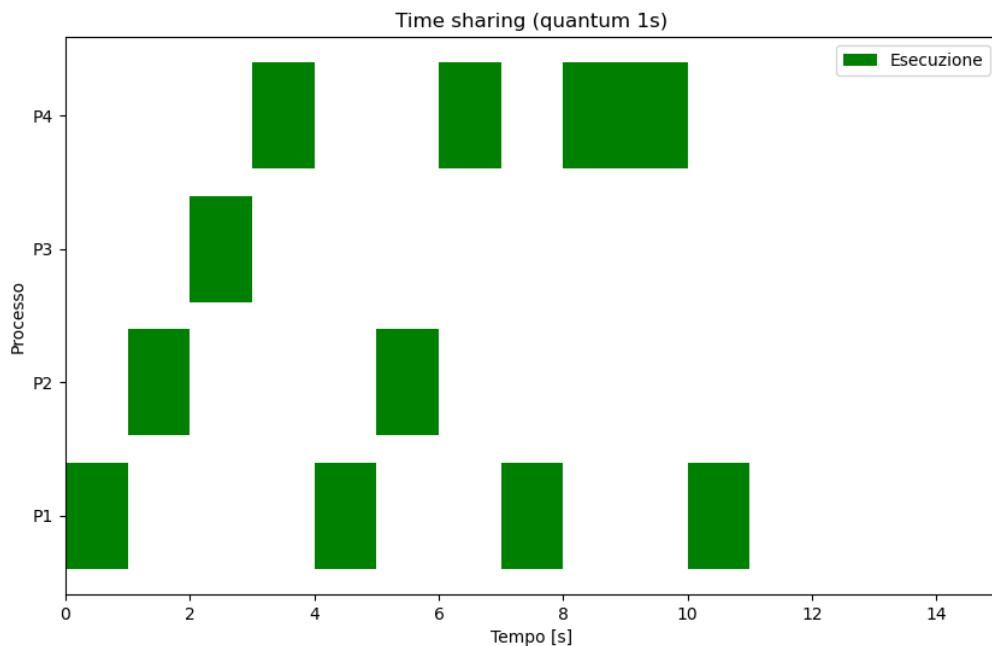
prelazione, nella quale la CPU impiegata nell'esecuzione di un processo può essere utilizzata anche per l'esecuzione di un processo successivo in eventuali tempi di attesa, garantendo una distribuzione equa delle risorse del sistema. Al contrario, nel multitasking cooperativo i processi cooperano volontariamente per condividere la CPU. Ogni processo deve rilasciare esplicitamente il controllo della CPU quando ha completato il suo lavoro o quando è bloccato. Se un processo non rilascia la CPU, può causare ritardi o blocchi nell'esecuzione degli altri processi. Il multitasking cooperativo è meno comune rispetto al multitasking prelazione ed è più suscettibile a problemi di prestazioni e stabilità.

Algoritmi di pianificazione Time Sharing

Il time sharing è una forma di multitasking in cui la CPU assegna un tempo fisso (quantum) a ciascun processo in modo ciclico. Se un processo non termina entro il suo quantum, viene interrotto e messo in coda per essere eseguito nel ciclo successivo.

- **Time Sharing:**

- **Descrizione:** Simile al Round-Robin, ma si concentra sul fornire un'esperienza di calcolo condivisa in cui ogni processo ottiene una giusta quota di tempo della CPU.



- **Pro:** Garantisce che tutti gli utenti ottengano una quantità equa di tempo della CPU.
- **Contro:** La commutazione di contesto può essere costosa, ad esempio se il quantum di tempo è molto piccolo, la CPU deve eseguire molte commutazioni di contesto in un breve periodo. Questo può portare a un elevato overhead di commutazione di contesto, riducendo l'efficienza complessiva del sistema..

Il sistema time-sharing, fa in modo che ogni processo venga eseguito ciclicamente per determinati intervalli di tempo detti *quanti*. Questo lasso di tempo, scelto arbitrariamente, determina i tempi di esecuzione dei processi. Come mostrato in figura, una volta determinato la grandezza di un quanto, ogni processo viene eseguito sequenzialmente per la durata di un quanto, e successivamente interrotto per far eseguire il processo successivo per un altro quanto e così via fino a che tutti i processi non verranno eseguiti.

0.1 Conclusion

Il sistema meno efficiente è sicuramente il monotasking in quanto permette l'esecuzione di un solo programma per volta. Il sistema multitasking e time sharing sono invece più efficienti perché permettono alla CPU di non rimanere inattiva durante l'attesa, riducendo così i tempi

di esecuzione complessivi. Utilizzando il monotasking vengono impiegati 15 secondi per l'esecuzione dei processi, con il multitasking 12 secondi e 11 secondi con il time sharing.