

# Esercizio sullo scheduling della CPU S3/L1

Manuel Di Gangi

5 febbraio 2024

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.

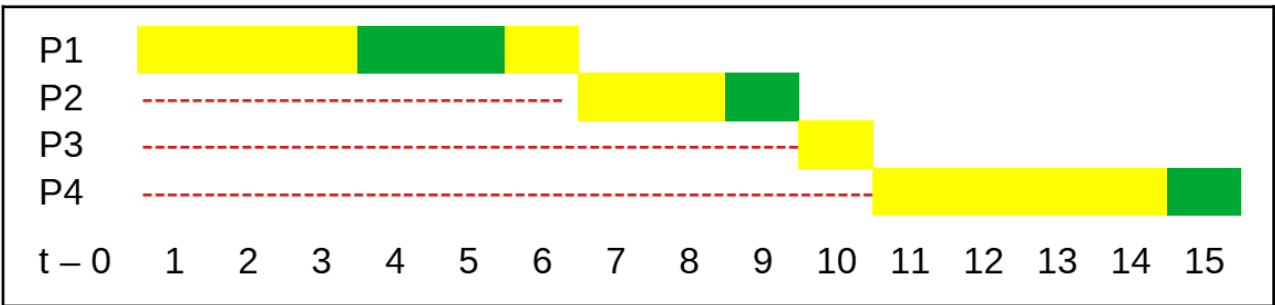
## TRACCIA

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 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 secondi	2 secondi	1 secondo
P2	2 secondi	1 secondo	-
P3	1 secondi	-	-
P4	4 secondi	1 secondo	-

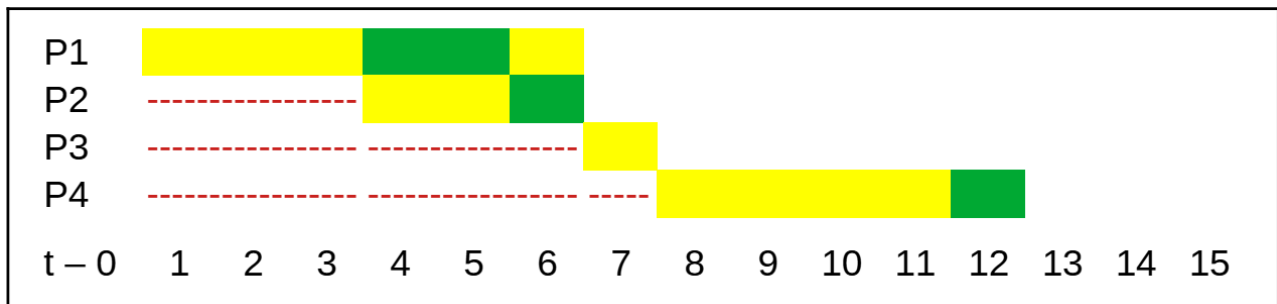
## Scheduler mono-tasking

In questi sistemi non è possibile sospendere l'esecuzione di un programma per assegnare la CPU ad un altro, risultando inefficiente a cause dei periodi di inutilizzo frequenti del processore. Penalizza in particolar modo i programmi che richiedono la CPU per un brevissimo lasso di tempo, come nel nostro caso P3, che deve attendere la terminazione dei processi che lo precedono.



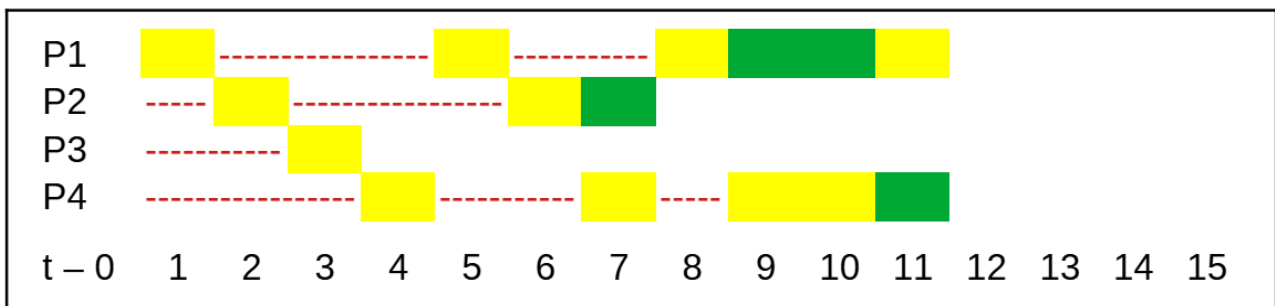
## Scheduler preemptive multi-tasking

Permettono l'esecuzione contemporanea di più programmi: la pianificazione con prelazione (preemptive multitasking) fa in modo che quando un processo è in attesa di eventi esterni, la CPU possa essere impiegata per altro, piuttosto che attendere inattiva. Così, quando il processo P1 passerà dallo stato di esecuzione allo stato di attesa, la CPU potrà essere impiegata per eseguire le istruzioni del processo P2.



## Scheduler time-sharing

Un'evoluzione dei sistemi multi-tasking sono i sistemi time-sharing. Ogni processo viene eseguito in maniera ciclica per piccole porzioni di tempo che prendono il nome di "quanti". Se un programma non rilascia la CPU entro questo tempo, viene interrotto e posto in coda fra i processi in stato "Pronto", facendoli così avanzare parallelamente.



## Conclusioni

Nel caso preso in considerazione lo scheduler più performante è il time-sharing, il quale permette di eseguire "parallelamente" tutti i programmi, non penalizzando i programmi piccoli facendoli attendere in coda ai più grandi e ottimizzando al meglio l'uso del processore.