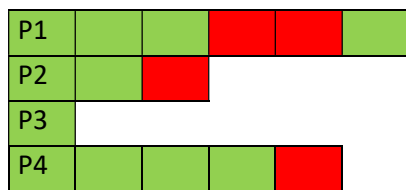


S3-L1

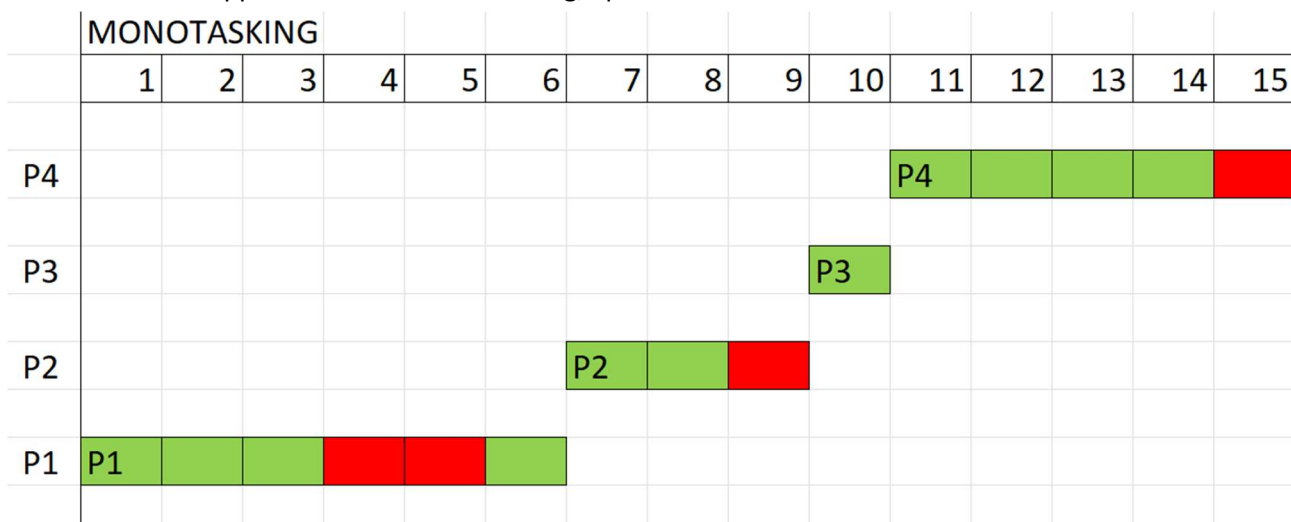
L'esercizio di oggi verte sui meccanismi di pianificazione dell'utilizzo della CPU (o processore). In ottica di ottimizzazione della gestione dei processi, abbiamo visto come lo scheduler si sia evoluto nel tempo per passare da approccio mono-tasking ad approcci multi-tasking. Traccia: Si considerino 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 secondo	-	-
P4	4 secondi	1 secondo	-

Per prima cosa rappresentiamo i 4 processi per avere una visuale più chiara della situazione:

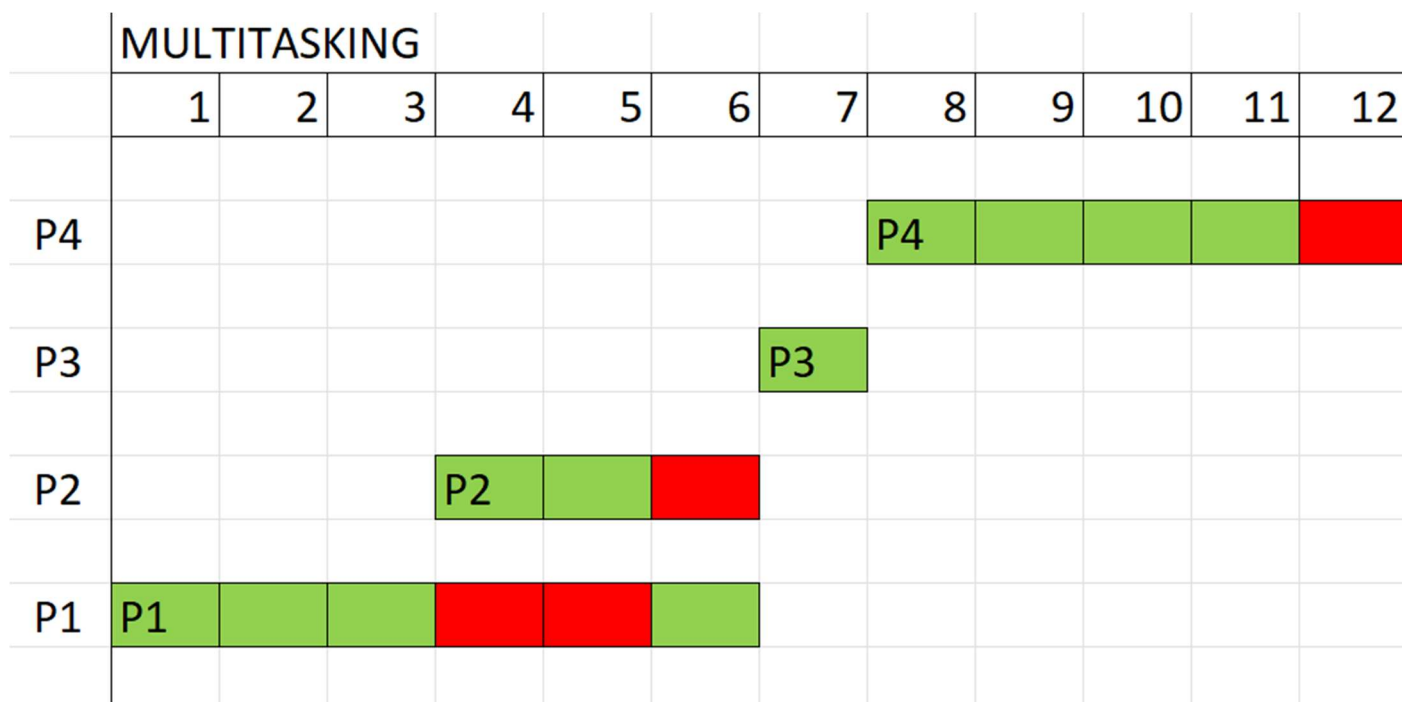


Partiamo con la rappresentazione mono-tasking; i processi verranno svolti nell'ordine di arrivo:



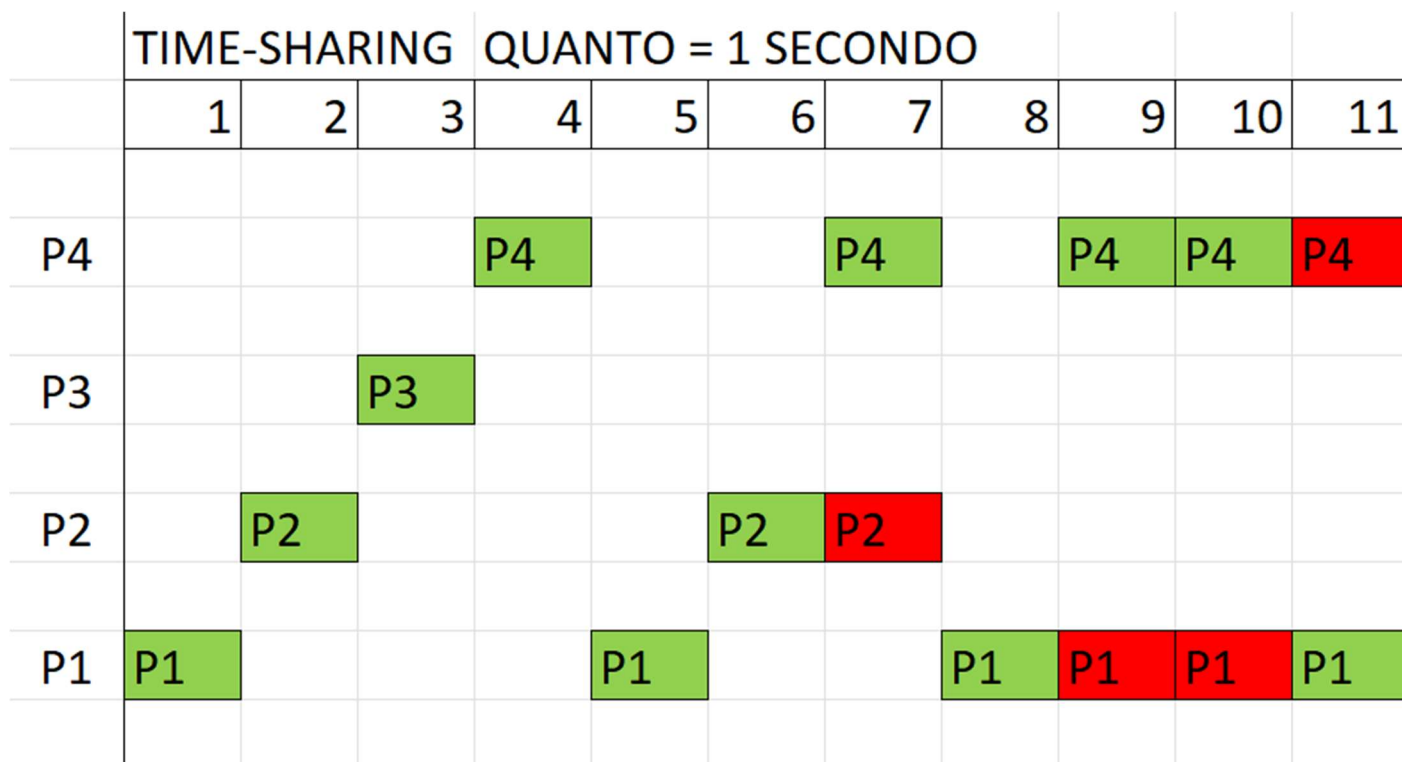
Essendo il monotasking inefficiente, il completamento dei 4 processi ci impiega 15 secondi, perché nei tempi d'attesa il processore attende e non impiega il tempo per svolgere altre attività.

Questa è invece la rappresentazione del multi-tasking:



In questo caso, il processore sfrutterà i tempi di attesa per proseguire con altre task. I tempi si abbassano da 15 a 11 secondi (considerando che il 12esimo secondo è un tempo di attesa e il processore potrà iniziare a svolgere un eventuale P5).

Abbiamo infine il time-sharing:



In questo caso, considerato il **quanto** di un secondo, il processore ci metterà sempre 11 secondi a svolgere tutti e 4 i processi, alternandoli 1 secondo alla volta.