

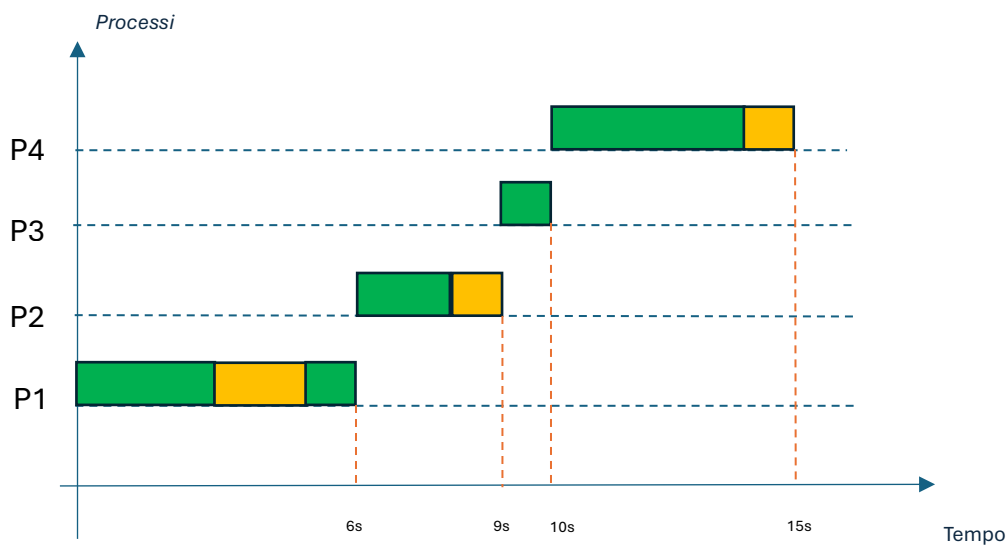
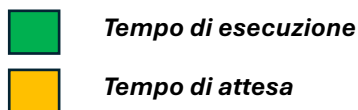
REPORT S3/L1

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 alle 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.

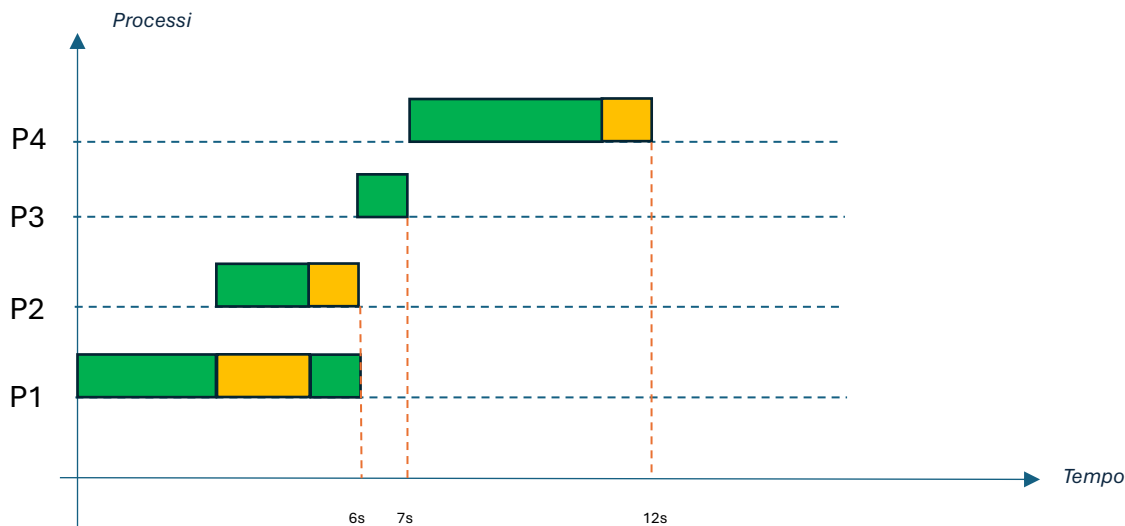
| Processo | Tempo di esecuzione | Tempo di attesa | Tempo di esecuzione dopo attesa |
|----------|---------------------|-----------------|---------------------------------|
| P1 | 3 secondi | 2 secondi | 1 secondo |
| P2 | 2 secondi | 1 secondo | - |
| P3 | 1 secondi | - | - |
| P4 | 4 secondi | 1 secondo | - |

Soluzione:

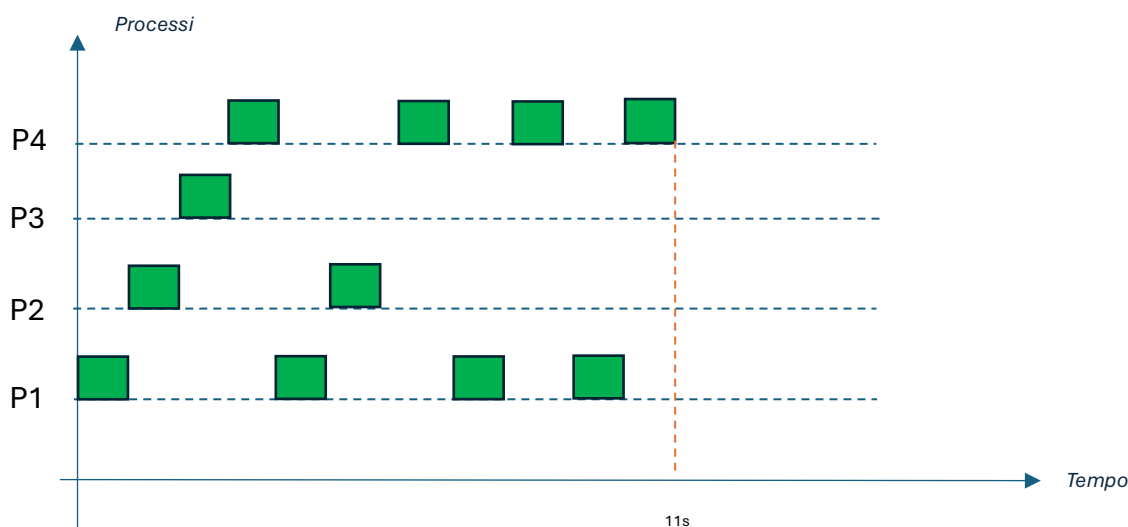
Mono-Tasking: In questo tipo di sistema i processi andranno in esecuzione in maniera lineare in ordine di arrivo, al termine dell'esecuzione del processo P1, seguiranno in sequenza i processi P2, P3, P4. Essi quindi sono in grado di gestire l'esecuzione di un programma per volta e la loro inefficienza è dovuta alla presenza dei "tempi morti".



MULTI-TASKING: Nel caso di un sistema MULTI-TASKING a differenza del caso precedente non è necessario che termini l'esecuzione di un processo per far sì che il processore venga assegnato al processo successivo, questo accade nel momento in cui uno dei processi passa in stato di attesa. Quando il processo P1 passerà in attesa dopo, 3 secondi, il processore sarà assegnato a P2 e così via. La differenza fondamentale quindi è che un processo può essere interrotto nel momento in cui va in stato di attesa di un evento esterno.



TIME-SHARING: In questo caso ho supposto un **quanto** di 1s, nei sistemi time sharing ai processi è assegnato 1 quanto di una determinata quantità di tempo ed essi verranno eseguiti in maniera ciclica, al termine del quanto il processore è assegnato al processo successivo e così via fino al termine dell'esecuzione di tutti i processi. Rispetto ai sistemi MULTI-TASKING qui l'esecuzione non si interrompe quando il processo passa in stato di attesa ma nel momento in cui termina il quanto assegnato al processo.



In conclusione i sistemi **MONO-TASKING** risulteranno i meno efficienti rispetto a **MULTI-TASKING** e **TIME-SHARING**.