

S3L1

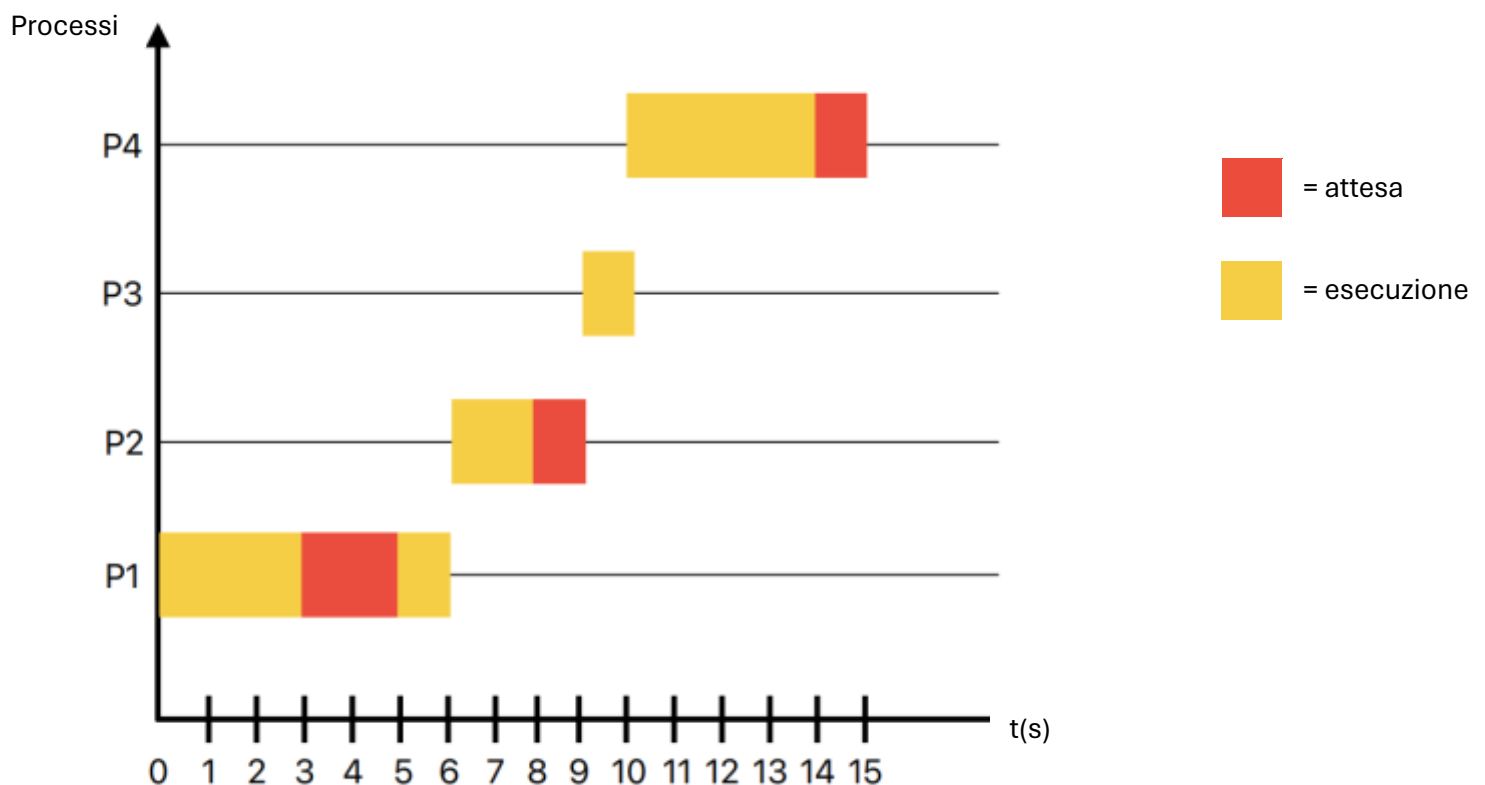
Si considerino 4 processi, che chiameremo P1,P2,P3,P4, con i tempi di esecuzione e di attesa input/output dati in tabella. I processi arrivano alle CPU in ordine P1,P2,P3,P4. Individuare il modo più efficace per la gestione e l'esecuzione dei processi, tra i metodi visti nella lezione teorica. Abbozzare 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	-

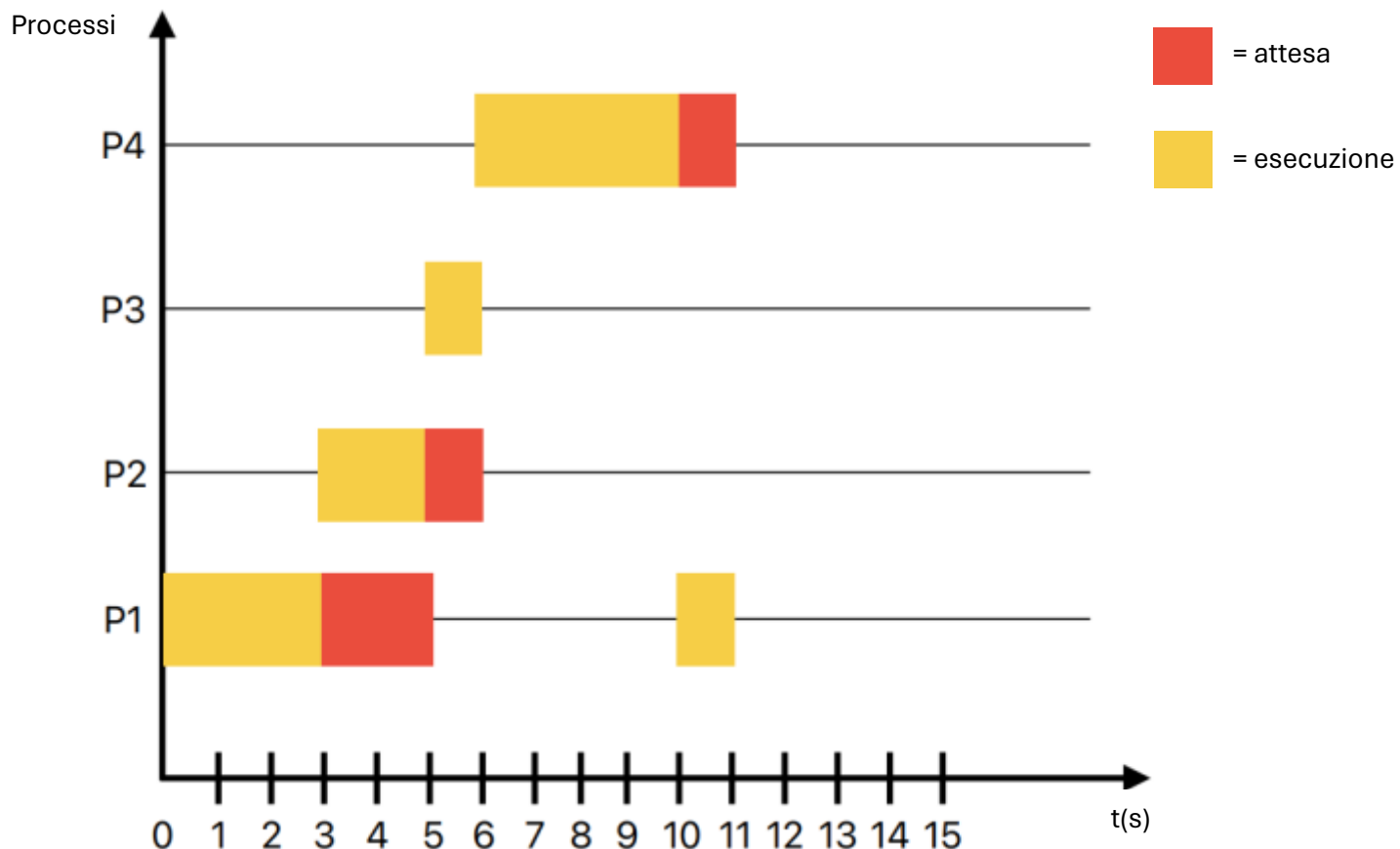
Andremo ad analizzare i metodi di gestione d'esecuzione dei processi per individuare il più efficace.

Lo **scheduler** è il modulo gestore dei processi può operare in diversi modi in base al sistema operativo ovvero in modalità monotasking, multitasking, time sharing.

Monotasking: Nei sistemi mono-tasking non è possibile effettuare più processi per volta, quindi non è possibile sospendere l'esecuzione di un programma per assegnare alla CPU un altro processo. Questo risulta penalizzare processi molto veloci e che hanno bisogno di poco tempo per essere eseguiti, che quindi devono aspettare che altri processi più lunghi vengano terminati prima di lui. È un esempio il P3 che deve attendere che processi come il P1 e P2 vengano eseguiti e finiti. Questo sistema è inefficiente dal punto di vista dei tempi di inutilizzo frequente del processore.

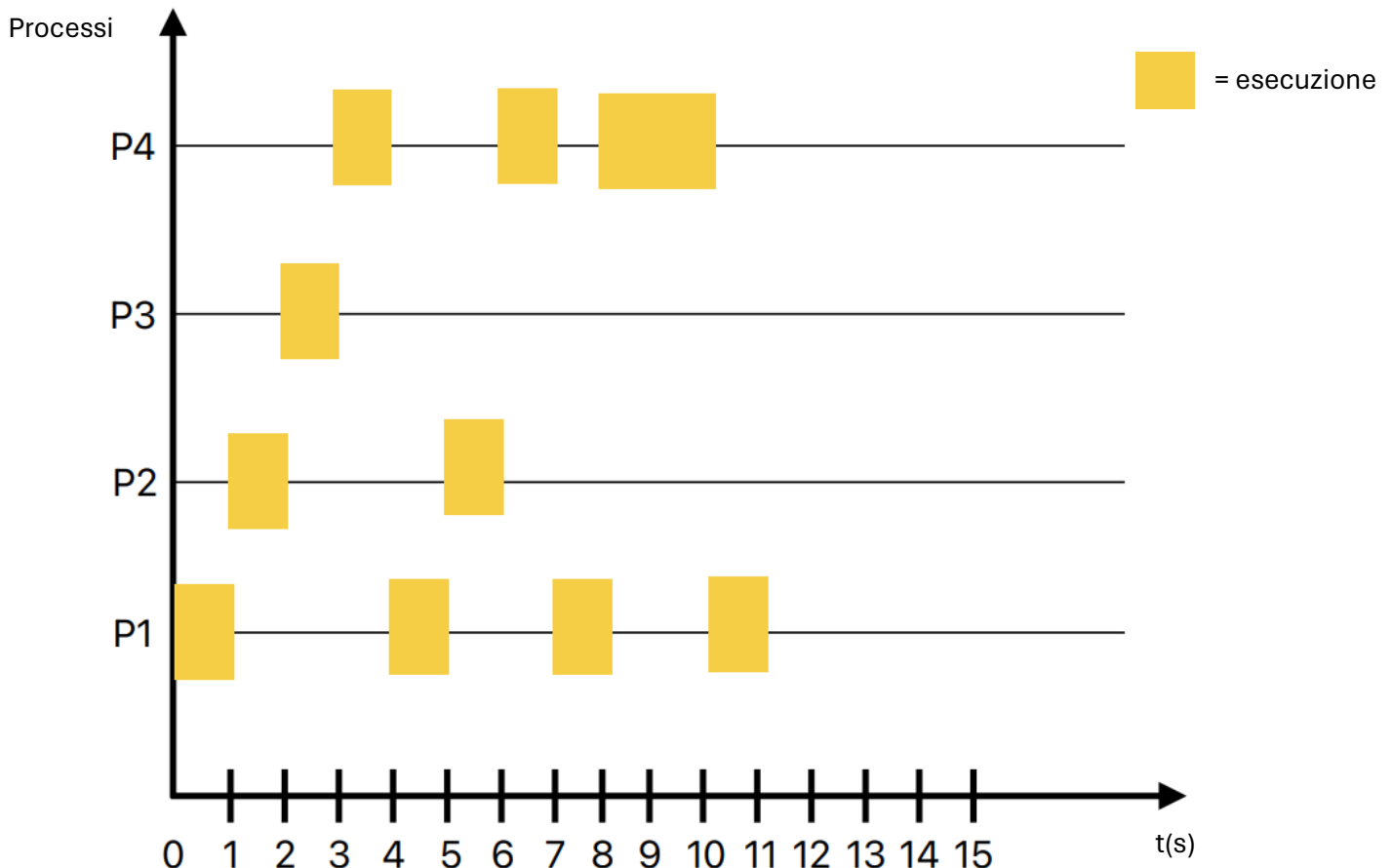


Multitasking: I sistemi multitasking sono progettati per consentire a un computer di eseguire più attività contemporaneamente, o almeno sembra così per l'utente. In realtà, il processore si occupa di eseguire rapidamente cicli di istruzioni per ogni attività in modo alternato, dando l'impressione che tutte le attività si svolgano simultaneamente. Questo è fondamentale per i moderni sistemi operativi, che devono gestire diverse richieste degli utenti e dei processi in esecuzione contemporaneamente. La pianificazione con prelazione fa in modo che, quando un processo è in attesa di eventi esterni la CPU può essere impiegata per un altro processo piuttosto che attendere inattiva. Ne fa un esempio il processo P2, che viene eseguito nel tempo di attesa di P1. Quando il processo P2 sarà in attesa non si potrà tornare indietro ma dovrà eseguire i processi successivi per poi tornare a quello iniziale.



Time Sharing: è un'evoluzione dei sistemi multitasking con la differenza che in questo caso il tempo di esecuzione della CPU viene diviso in tanti piccoli intervalli di tempo detti **quanti**.

Il processo viene interrotto per passare ad eseguire un altro processo per 1 quanto e così via.



In conclusione, il monotasking non è l'ideale in situazioni in cui è necessario rispondere rapidamente a richieste o eventuali compiti che richiedono una rapida alternanza di attenzione.

Mentre il multitasking, a differenza del monotasking, permette l'esecuzione di altri processi nei tempi di attesa, riducendo l'inattività della CPU, senza partizionare il tempo in parti uguali come il time sharing. Per questo il più efficiente è il time sharing, il quale permette di eseguire più processi ponendo la stessa attenzione per tutti i processi, piccoli o lunghi, e ottimizzando al meglio la CPU.