

S3/L1 - Scheduling CPU

L'obiettivo dell'esercizio di oggi è analizzare i meccanismi di pianificazione della CPU secondo gli approcci mono-tasking, multi-tasking e time-sharing, in ottica di ottimizzare la gestione dei processi.

In particolare:

Si considerino 4 processi, che chiameremo P1,P2,P3,P4, con i tempi di esecuzione e di attesa input/output dati in tabella.

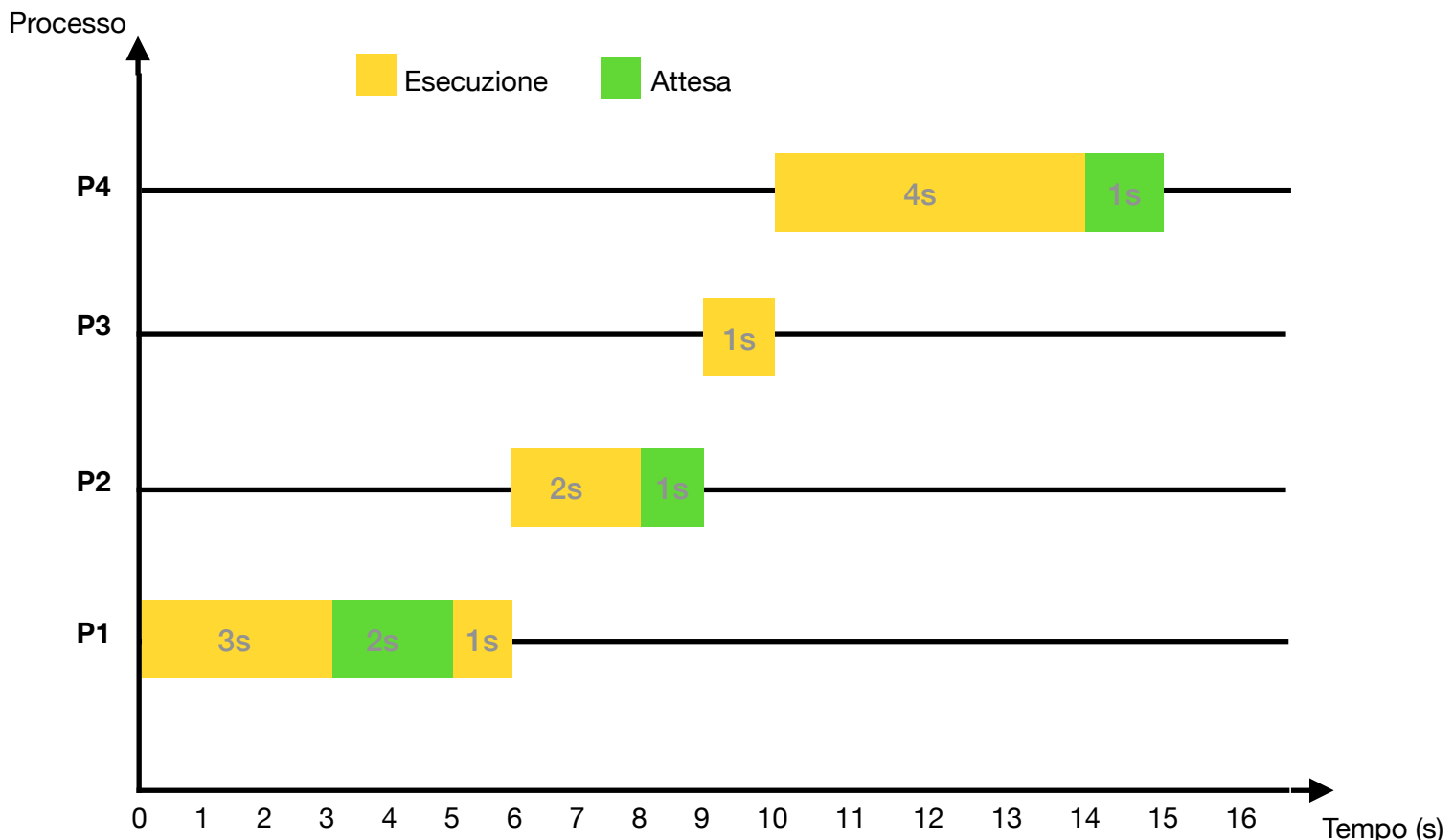
Processo	Tempo di esecuzione	Tempo di attesa	Tempo di esecuzione dopo attesa
P1	3 secondi	2 secondi	1 secondo
P2	2 secondi	1 secondo	-
P3	1 secondi	-	-
P4	4 secondi	1 secondo	-

I processi arrivano alle CPU in ordine P1,P2,P3,P4. Individuare il modo più efficace per la gestione e l'esecuzione dei processi.

Metodo: Analizzerò i dati secondo gli approcci allegando un grafico rappresentativo per ognuno di essi.

1. MONO-TASKING

- L'approccio mono-tasking prevede che la CPU gestisca un solo processo alla volta.
Quindi un processo deve terminare completamente, anche eventuali tempi di attesa, prima che il successivo possa iniziare.
Perciò non consente sovrapposizioni tra esecuzione ed attesa:

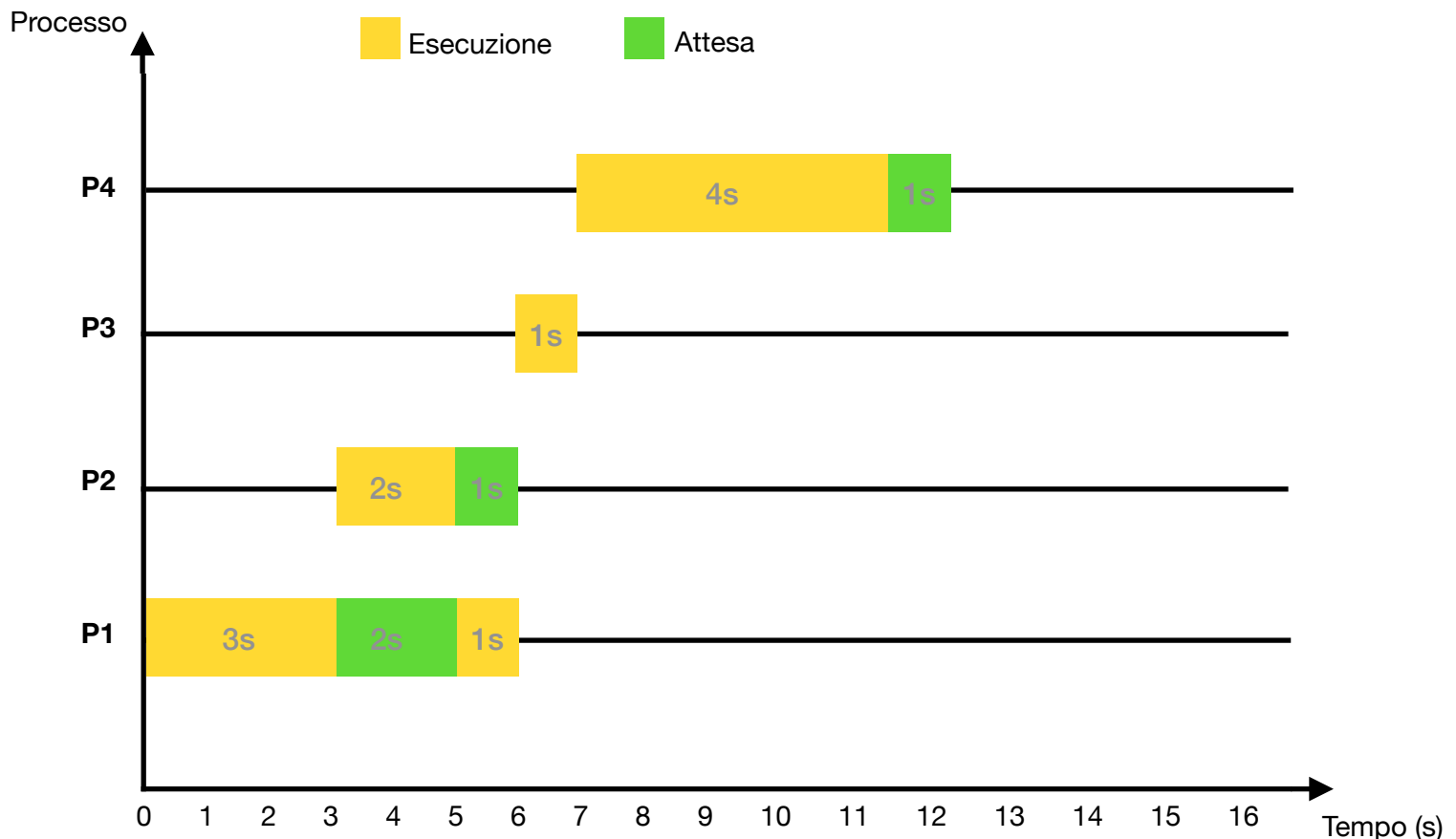


- P1 viene eseguito per 3 secondi, poi 2 secondi di attesa e infine 1 secondo di esecuzione dopo l'attesa.
- P2, P3 e P4 si susseguono senza sovrapposizioni.
- L'approccio mono-tasking mostra uno scarso utilizzo della CPU

2. **MULTI-TASKING**

- Nell'approccio multitasking la CPU può eseguire più processi in maniera simultanea.

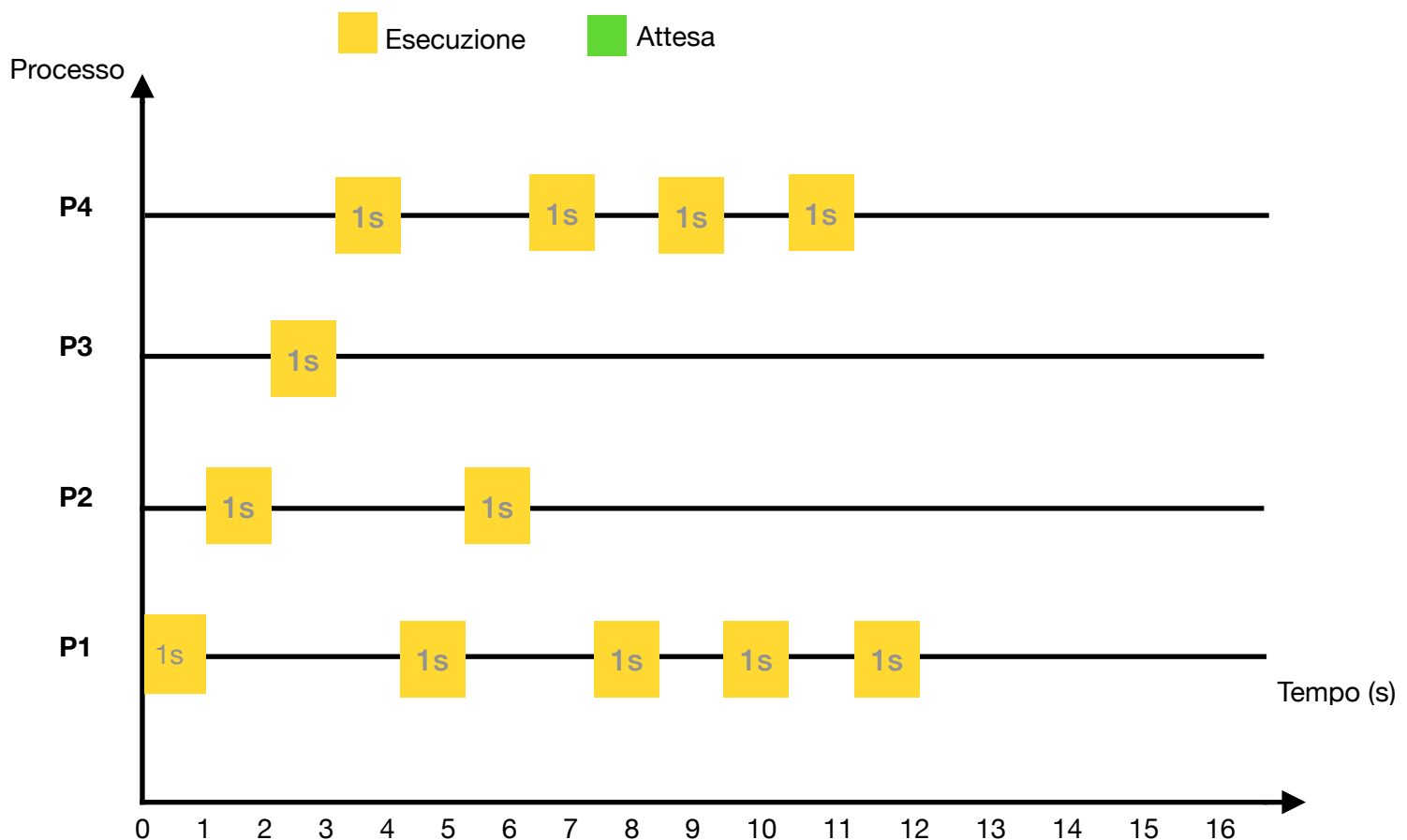
Durante il tempo di attesa di un processo la CPU passa all'esecuzione di un altro processo ottimizzando i tempi:



- Come si vede, durante i 2 secondi di attesa di P1, la CPU inizia a processare P2.
- In questo caso abbiamo un utilizzo più efficiente della CPU, riducendo il tempo totale necessario per eseguire tutti i processi.

3. TIME-SHARING

- L'approccio time-sharing è un'evoluzione del multi-tasking; ogni processo viene eseguito in maniera ciclica per brevi lassi di tempo detti quanti.
Quindi il processo in esecuzione viene interrotto per eseguire un altro processo, che poi viene interrotto per eseguirne un altro...:



- È evidente che i processi P1, P2, P3 e P4 vengono eseguiti a intervalli regolari (quanti).
- Questo approccio ottimizza l'utilizzo della CPU ma può talvolta risultare inefficiente per processi molto brevi che richiedono un solo ciclo di esecuzione.

Conclusioni:

Gli approcci più efficienti sono sicuramente il multi-tasking e il time-sharing. Come detto sopra, il sistema time-sharing può in qualche caso essere inefficiente in quanto il valore dei “quanti” viene definito a monte e quindi potrebbe interrompere dei processi anche molto brevi. Il multi-tasking invece si serve dello stato di attesa di ogni processo per dare via al successivo, il che potrebbe tradursi in tempi di esecuzione abbastanza lunghi prima di passare al processo successivo.