

ESERCIZIO: pianificazione dell'utilizzo della CPU.

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	-

Diagramma Mono-Tasking

- Ogni processo termina completamente prima che inizi il successivo.
- La CPU resta **inattiva** durante i tempi di attesa (verde).
- Inefficiente perché non si sfrutta il tempo di attesa per eseguire altri processi.

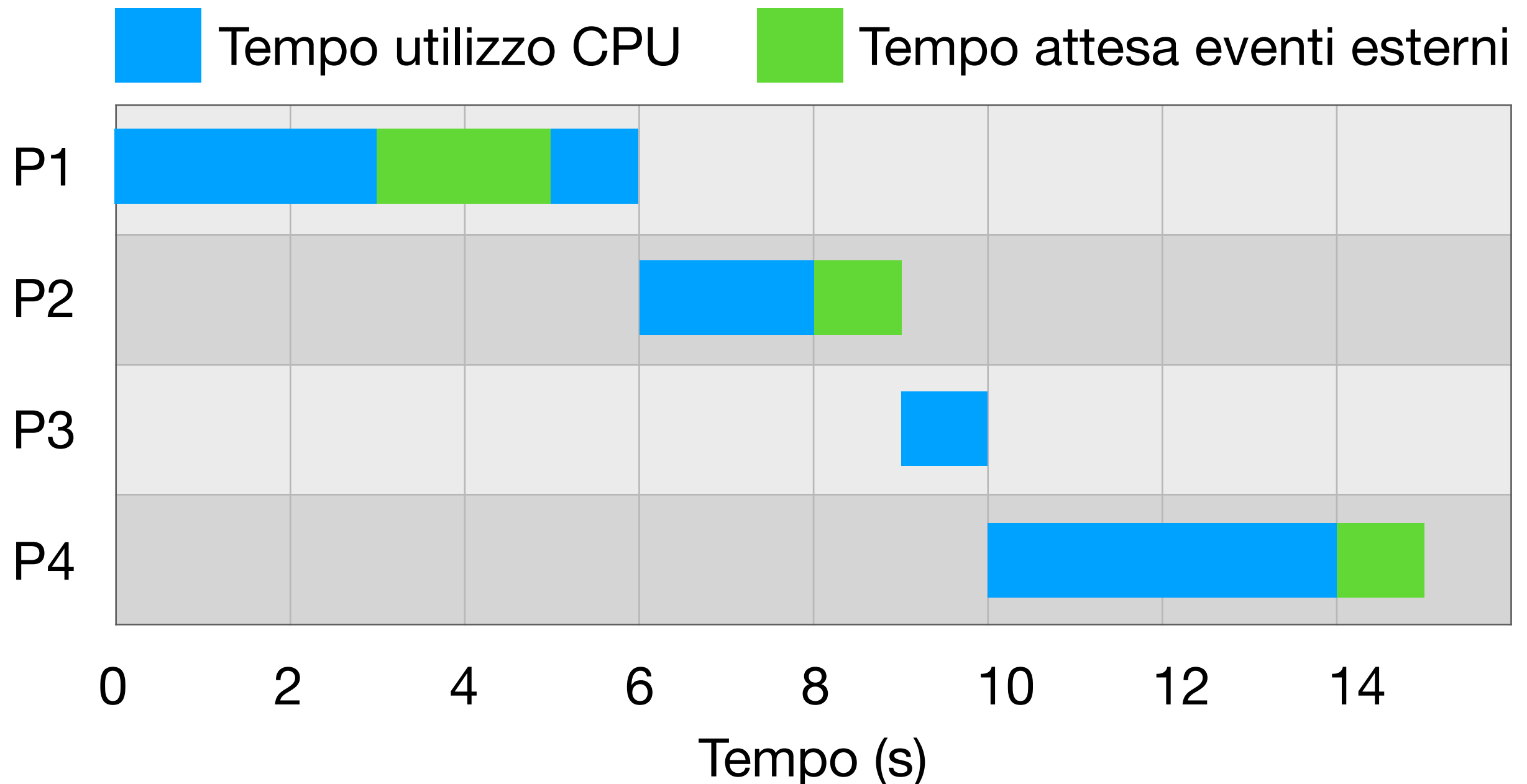


Diagramma Multi-Tasking (Preemptive)

- I processi **possono essere sospesi** durante l'attesa per far spazio ad altri.
- La CPU è **più sfruttata**, riducendo i tempi morti.
- È il modello più efficiente tra i tre.

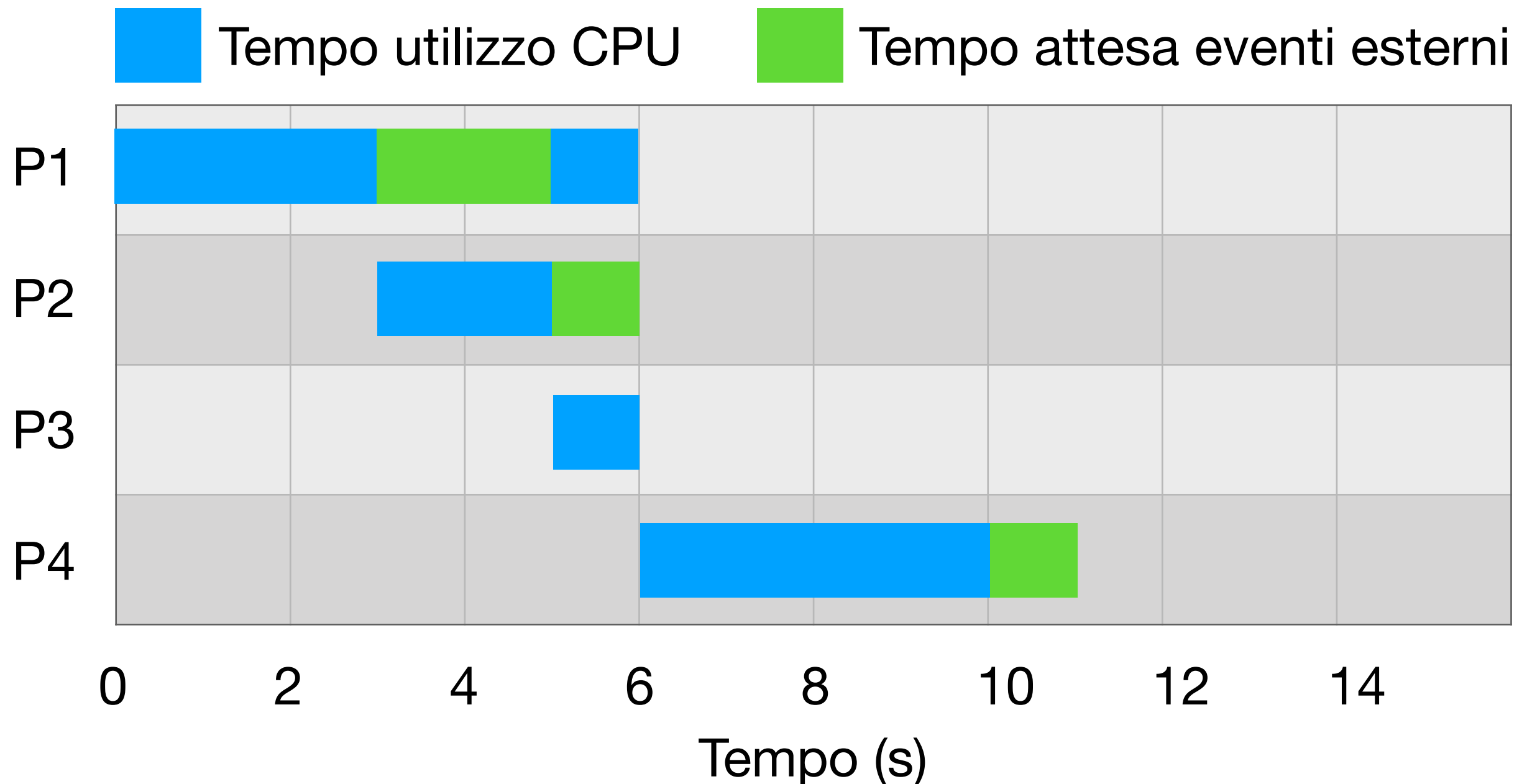


Diagramma Time-Sharing

- Ogni processo riceve un **quanto di tempo** (es. 1 secondo) in modo ciclico.
- Apparente parallelismo tra processi.
- Molto utile per sistemi interattivi e multi-utente, ma con un po' di overhead.

