

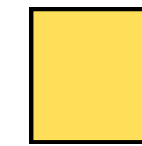
Esercizio S3L1

L'esercizio di oggi verte sui meccanismi di pianificazione dell'utilizzo della CPU. Data la tabella sottostante, bisogna analizzare i dati utilizzando i 3 metodi di gestione dei processi e capire qual è il più efficace, vediamo dunque come si presentano uno ad uno in un diagramma.

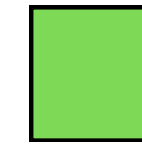
Processo	Tempo di esecuzione	Tempo di attesa	Tempo di esecuzione dopo attesa
P1	3 secondi	2 secondi	1 secondo
P2	2 secondi	1 secondo	-
P3	1 secondi	-	-
P4	4 secondi	1 secondo	-

Sistema mono-tasking

Processo



Tempo utilizzo CPU



Tempo attesa eventi esterni

P4

P3

P2

P1

1s

2s

3s

4s

5s

6s

7s

8s

9s

10s

11s

12s

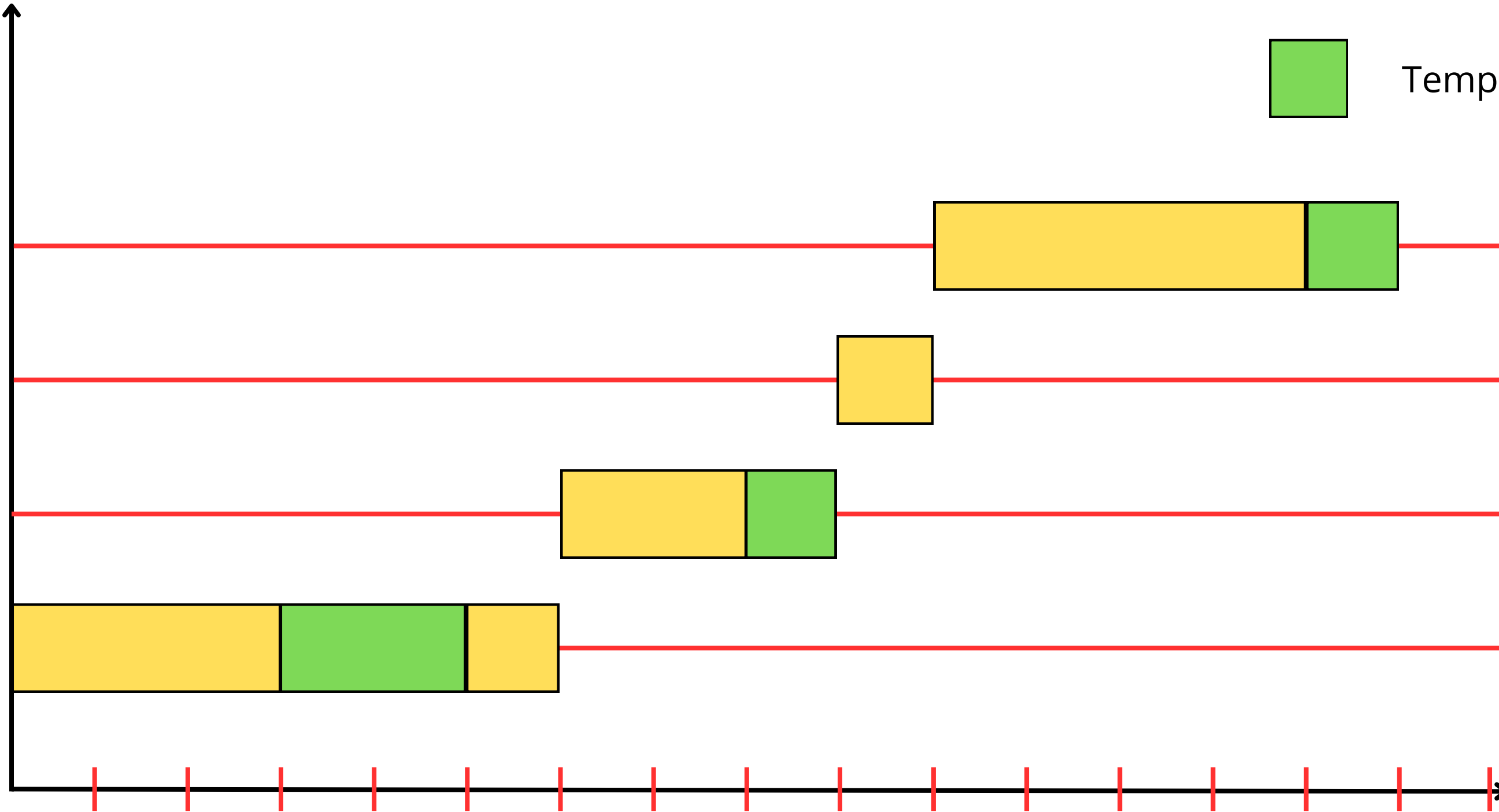
13s

14s

15s

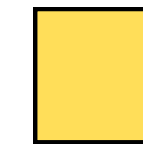
16s

tempo (t)

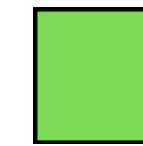


Sistema multi-tasking

Processo



Tempo utilizzo CPU



Tempo attesa eventi esterni

P4

P3

P2

P1

1s

2s

3s

4s

5s

6s

7s

8s

9s

10s

11s

12s

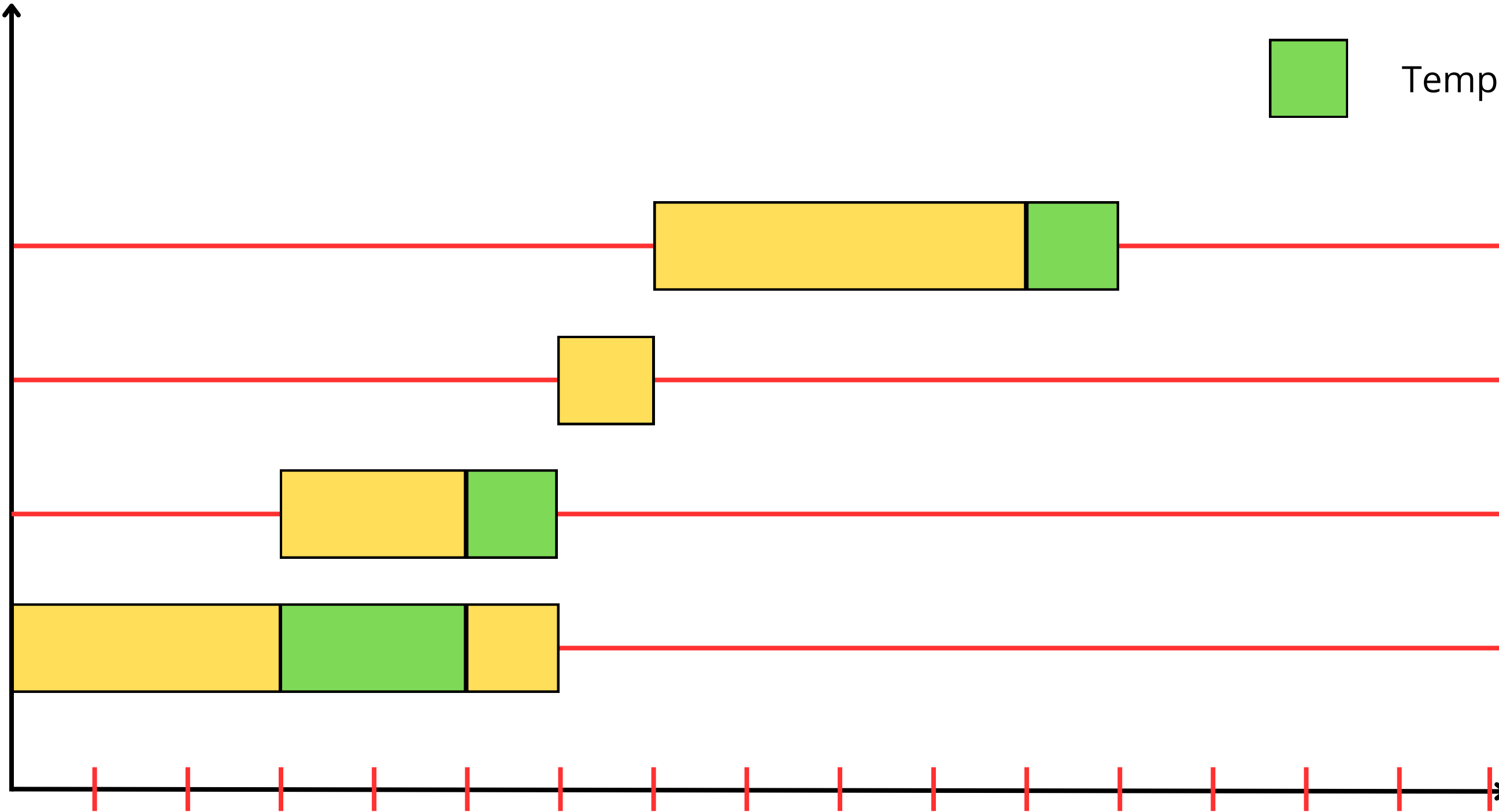
13s

14s

15s

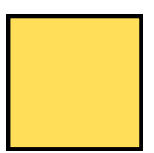
16s

tempo (t)

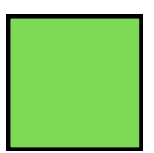


Sistema time-sharing (1s di quanto)

Processo



Tempo utilizzo CPU



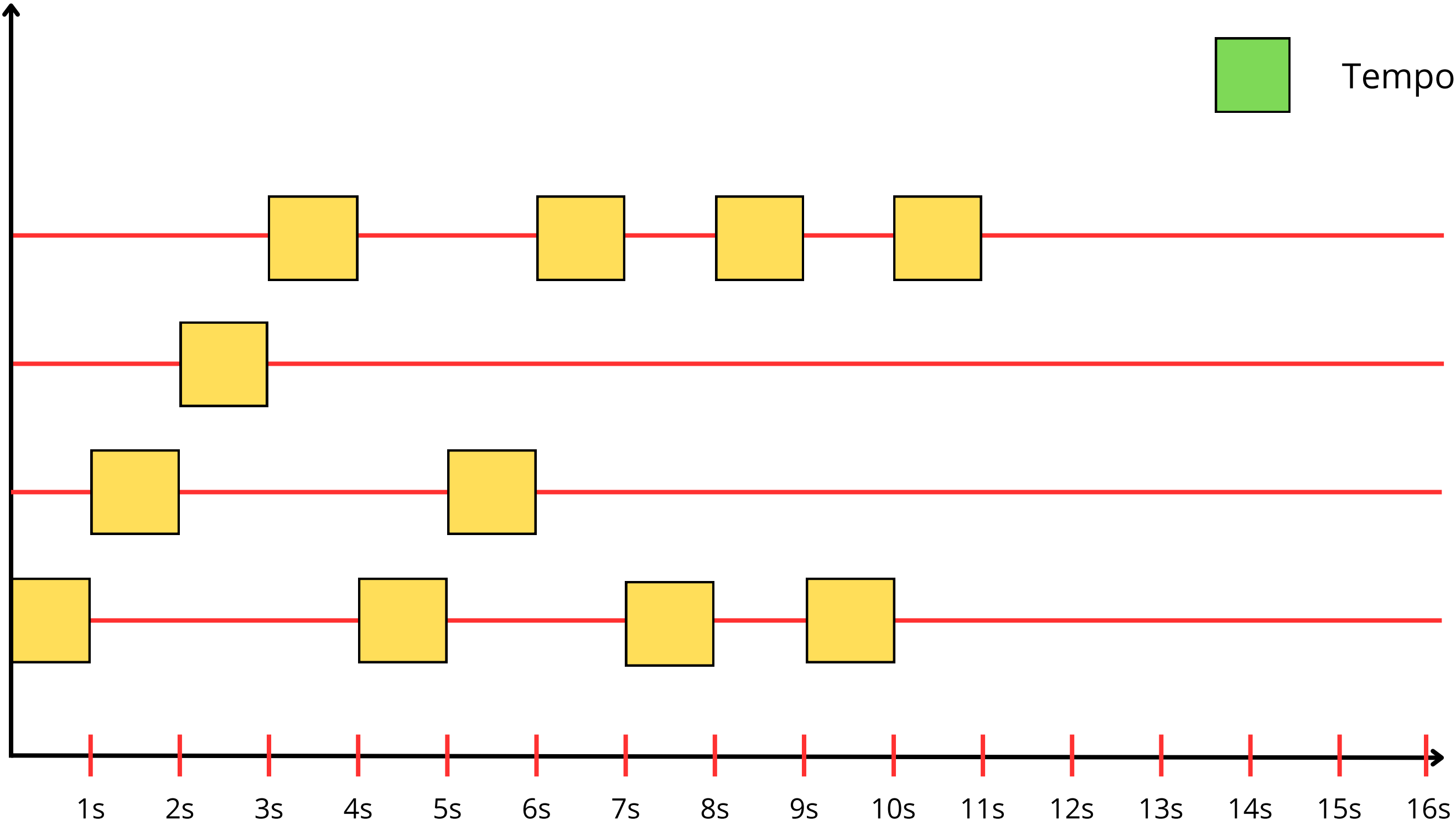
Tempo attesa eventi esterni

P4

P3

P2

P1



tempo (t)

Considerazioni finali

La prima cosa che si può evincere è che il metodo **mono-tasking** è sicuramente quello più sconveniente perchè può eseguire un solo processo alla volta, e questo è controproducente soprattutto in termini di tempo (ben 15s di attesa per completare tutti e 4 i processi).

I sistemi **multi-tasking** e **time-sharing** sono ovviamente quelli più convenienti in termini di resa e tempistiche. Il sistema multi-tasking può impiegare CPU in altri processi quando un determinato processo passa dallo stato di esecuzione allo stato di attesa di eventi esterni; mentre il sistema time-sharing distribuisce l'esecuzione dei processi in piccoli range temporali detti "**quanti**" (**1s** nel caso preso in esame).

Concludendo, nel primo caso ci impiegherà 15 secondi, mentre nel secondo e nel terzo rispettivamente 12 secondi e 11 secondi, risultando quindi i più efficienti. A rigor di tempistiche, sceglierei il sistema **time-sharing**.