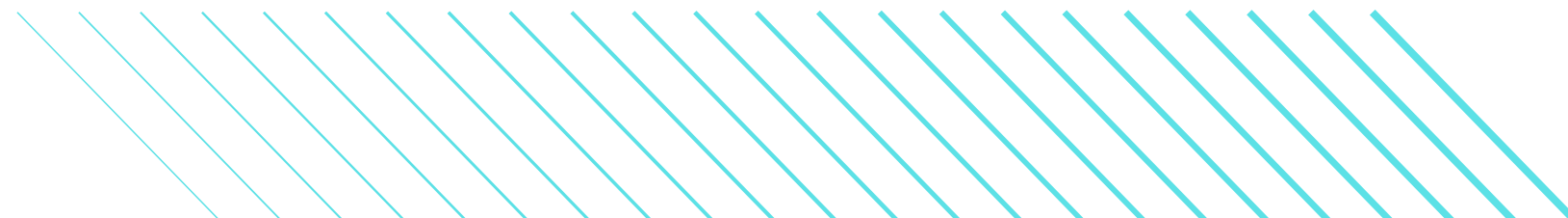
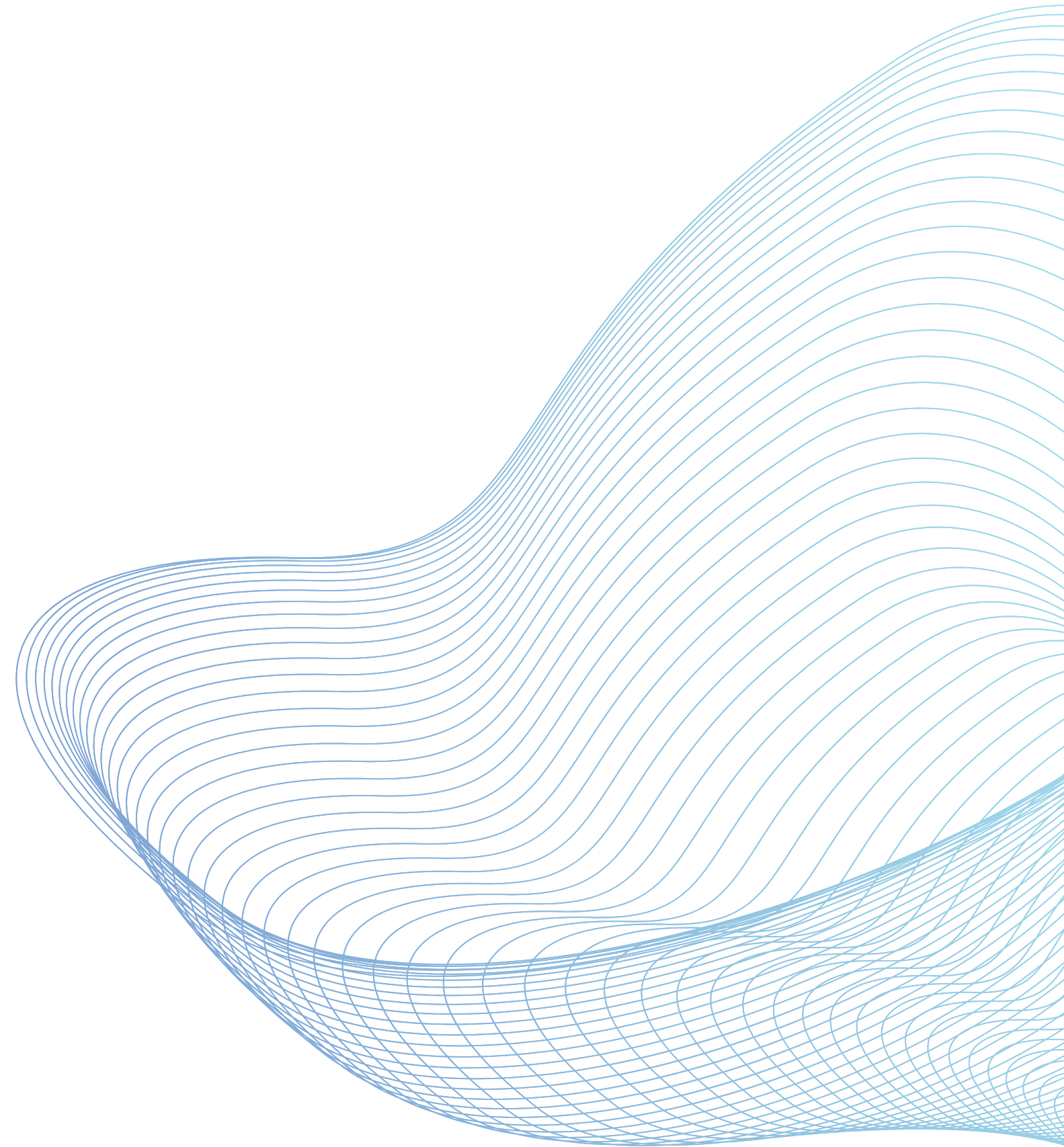


EPICODE

CS0424 **S3/L1**

Victoria M. Braile



SOMMARIO

Elenco dei contenuti della presentazione

- Traccia
- Legenda
- Mono-Tasking
- Multi-Tasking
- Time Sharing
- Conclusione

PRATICA - 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 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 secondo	-	-
P4	4 secondi	1 secondo	-

LEGENDA

Per realizzare un diagramma per ciascuno dei tre approcci di scheduling ho associato un determinato colore ai quadratini che rappresentano i secondi di tempo di utilizzo della CPU, ed un altro colore ai quadratini che rappresentano i secondi di tempo di attesa.



1 SECONDO DI UTILIZZO CPU

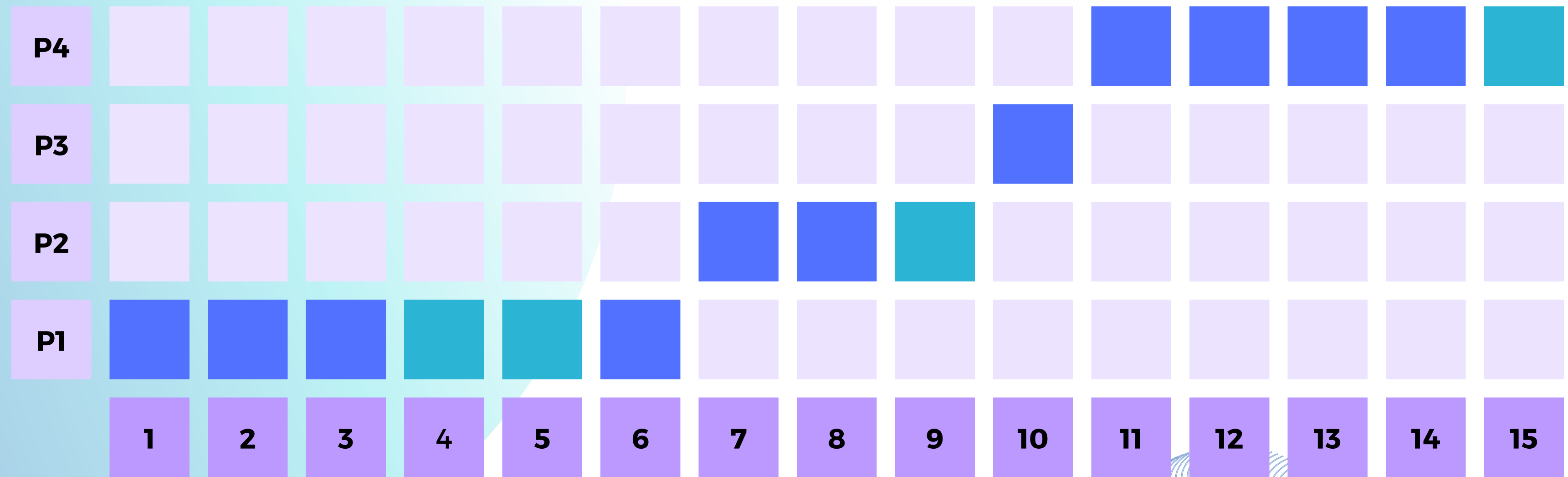


1 SECONDO DI ATTESA

MONO-TASKING

In questo approccio, ogni processo viene eseguito fino al suo completamento prima di passare al processo successivo.

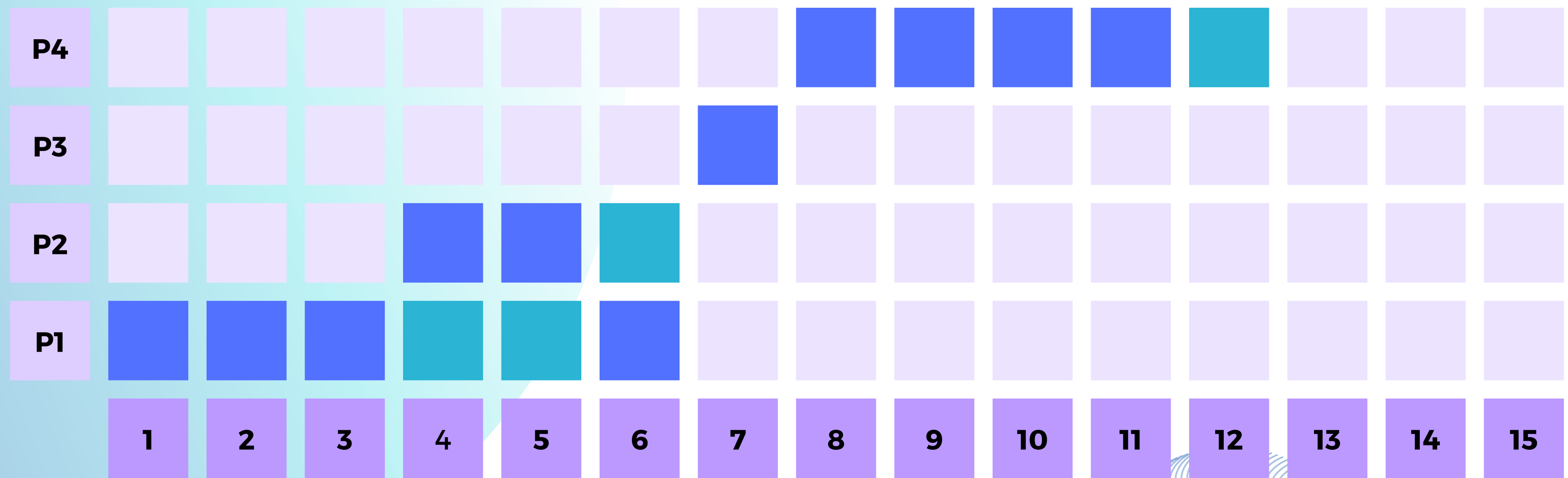
Ciò comporta tempi di attesa molto lunghi per i processi che arrivano successivamente.



Tempo (secondi)

MULTI-TASKING

Questo approccio permette l'esecuzione di più programmi. Ogni processo viene eseguito fino a quando non richiede un'operazione di I/O, quindi viene messo in attesa e il processo successivo viene eseguito. Ciò comporta tempi di attesa più brevi rispetto all'approccio mono-tasking.

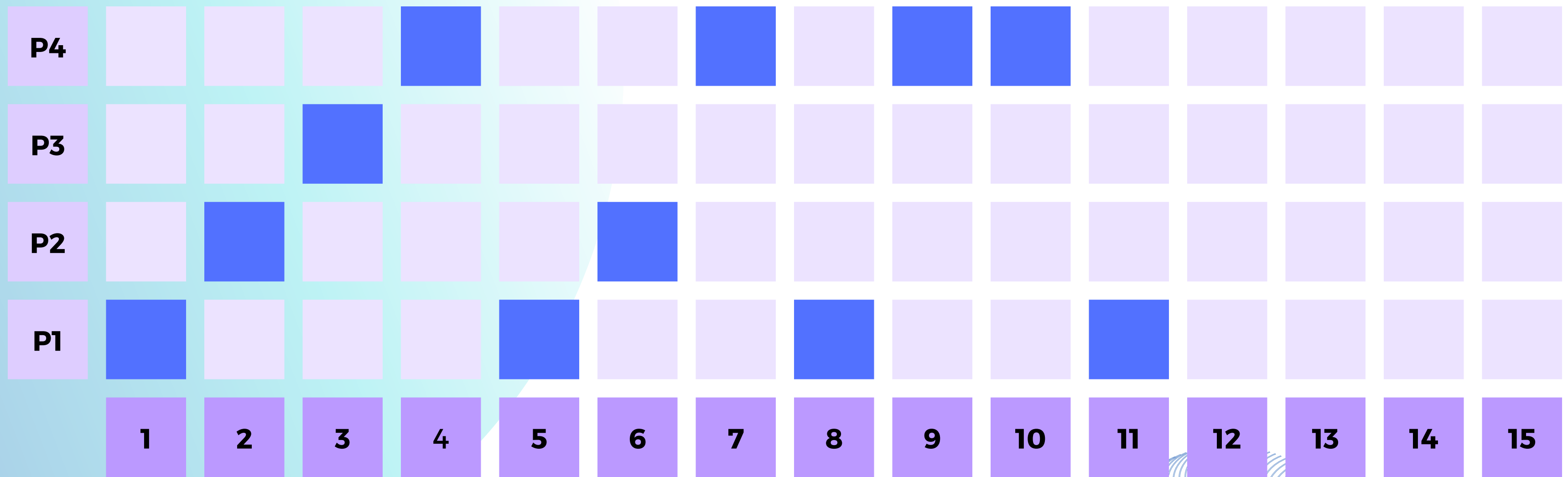


Tempo (secondi)

TIME SHARING

In questo caso, ogni processo viene eseguito per un certo intervallo di tempo chiamato “quanto” prima di passare al processo seguente.

Ciò comporta tempi di attesa molto brevi e un utilizzo più efficiente della CPU.



Tempo (quanti)

CONCLUSIONE

L'approccio più efficace per la gestione e l'esecuzione dei processi è il **time sharing**, che consente di minimizzare i tempi di attesa e di massimizzare l'utilizzo della CPU.

CS0424

S3/L1

Victoria M. Braile



[victoriabraile/CS0424_S3L1.git](https://github.com/victoriabraile/CS0424_S3L1.git)

