

Esercizio su scheduling della CPU

Traccia:

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. 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.

Il metodo più efficace (criterio di valutazione: posso usarlo su praticamente in tutte le macchine con utilizzi più disparati) è quello del **multi-tasking** detto anche **Priority Scheduling**.

Dati i processi:

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	-

Assegniamo una priorità in base ai tempi di esecuzione: il processo con **minor tempo di esecuzione** ottiene la **priorità più alta**.

P3 (1 secondo)

P2 (2 secondi)

P1 (3 secondi)

P4 (4 secondi)

Leggendo la tabella spieghiamo:

Il **tempo di attesa** di un processo sta per **tempo totale trascorso** prima che inizi la sua esecuzione

Il **Tempo di completamento** sta per **tempo di attesa + tempo di esecuzione**.

Processore 3

Tempo di attesa: 0 sec

Tempo di completamento: 1 sec

Processore 2

Tempo di attesa: 1 sec

Tempo di completamento: 3 sec

Processore 1

Tempo di attesa: 3 sec

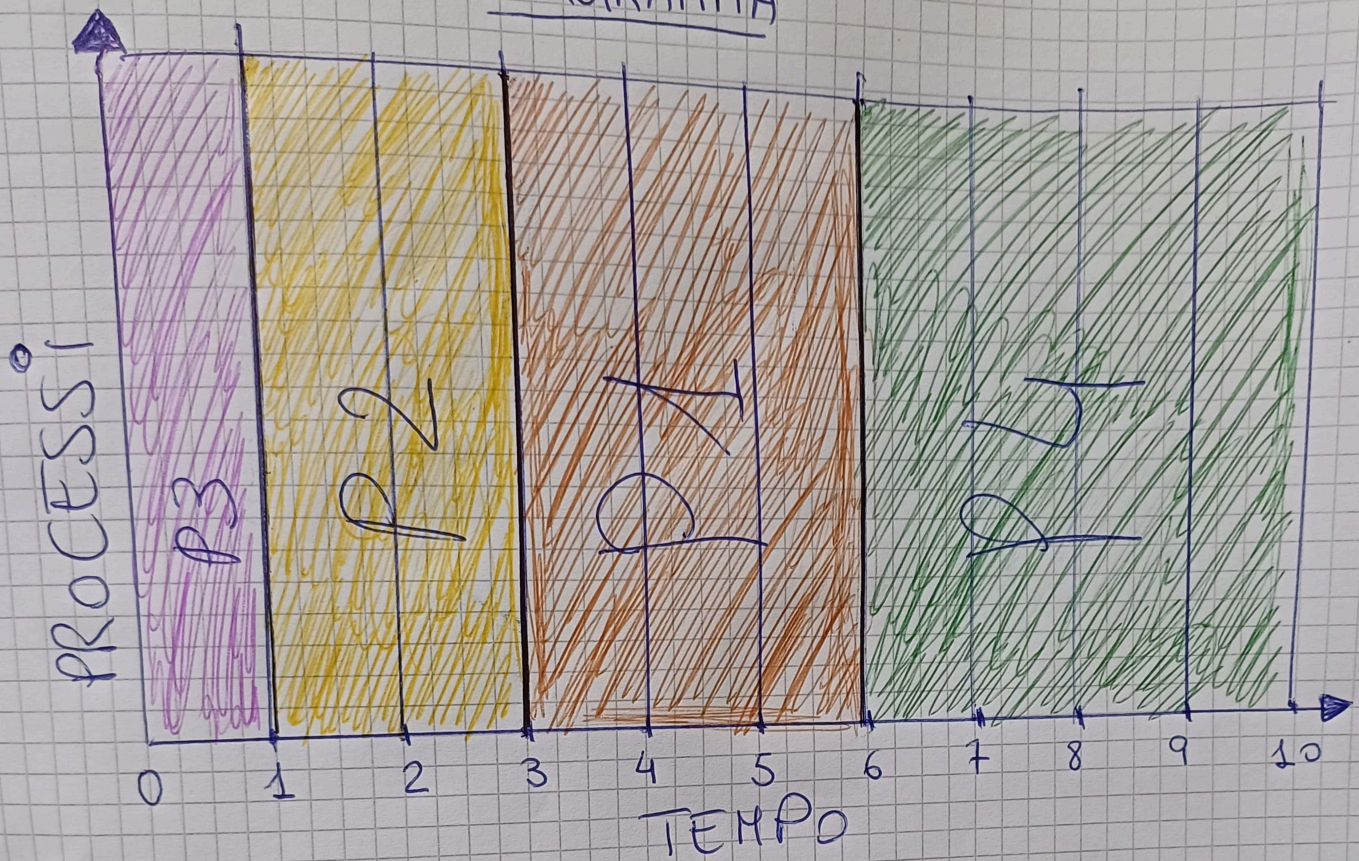
Tempo di completamento: 6 sec

Processore 4

Tempo di attesa: 6 sec

Tempo di completamento: 10 sec

DIAGRAMMA



- Leggenda

● P3

● P2

● P1

● P4