

Esercitazione sullo scheduling della CPU W4D1

Prendendo in considerazione i dati contenuti nella tabella sottostante, troviamo il metodo di scheduling della CPU più efficiente considerando solo i metodi analizzati a lezione.

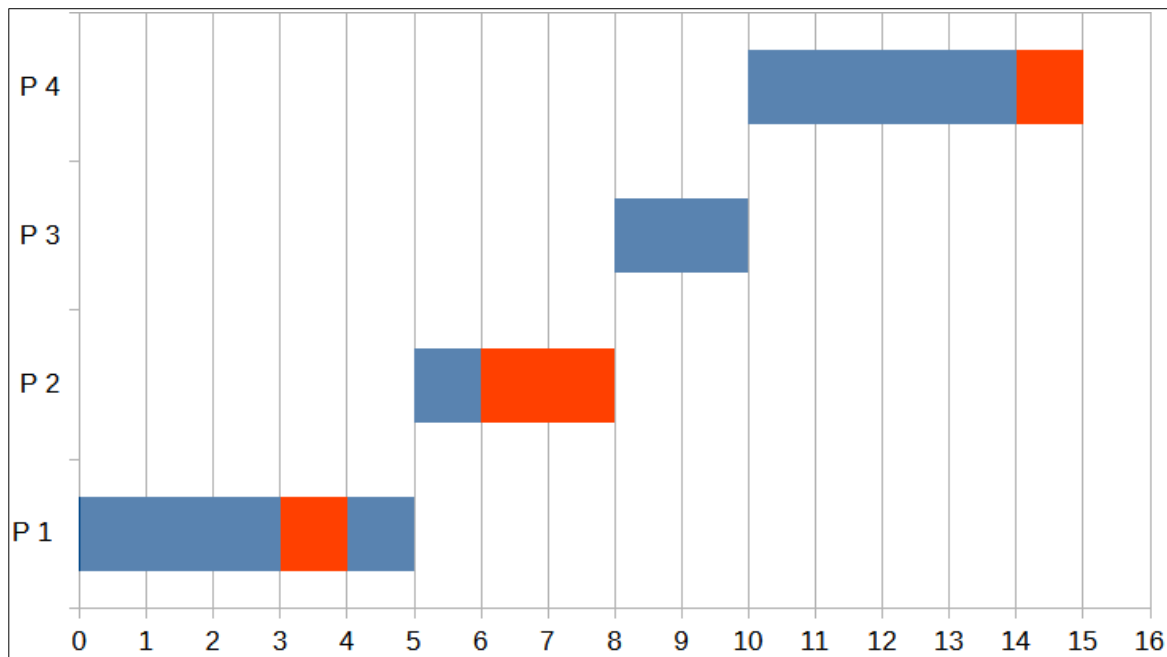
Processo	Tempo di esecuzione	Tempo di attesa	Tempo di esecuzione dopo l'attesa
P1	3	1	1
P2	1	2	-
P3	2	-	-
P4	4	1	-

Legenda Grafici:

- **Azzurro**: tempo utilizzo CPU
- **Arancione**: tempo di attesa eventi esterni

Esecuzione mono-tasking

Nell'esecuzione mono-tasking i processi vengono eseguiti uno alla volta senza possibilità di sospendere il precedente per dare priorità al successivo, quindi il P2 potrà essere eseguito solo al completamento del P1. Questo rende il mono-tasking un metodo poco efficace a causa dell'inutilizzo della CPU nei tempi morti ed è considerato antiquato.

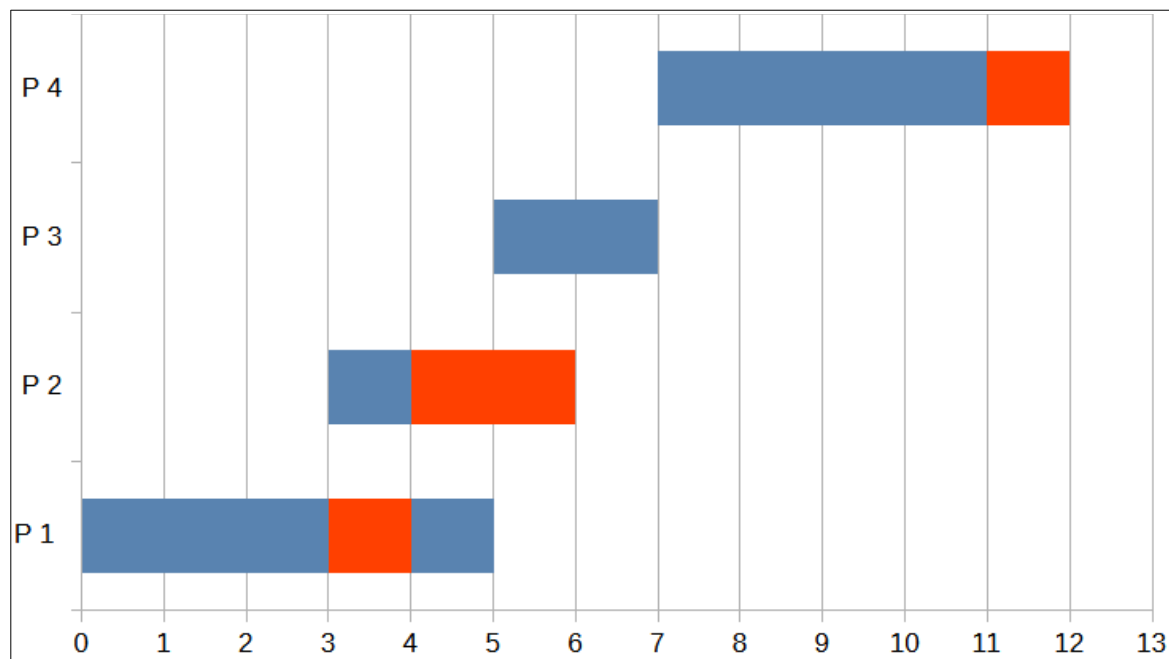


Esecuzione multi-tasking

Nell'esecuzione multi-tasking i processi possono essere svolti anche in contemporanea.

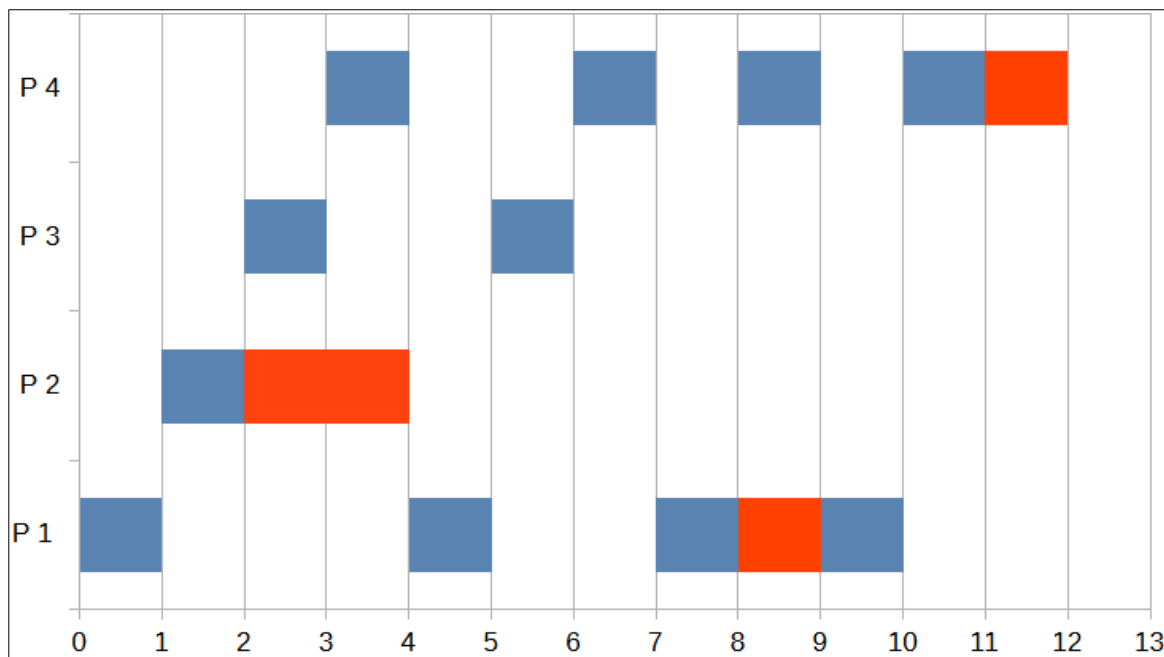
Utilizzando il preemptive multitasking, la CPU può essere impiegata per l'esecuzione di altri processi mentre il precedente processo è nella finestra di attesa di eventi esterni, basandosi sulla priorità assegnata ai processi.

Come possiamo vedere dal grafico, mentre P1 è in attesa, la CPU "riempie" quello spazio di tempo eseguendo P2.



Esecuzione time-sharing

L'esecuzione time-sharing è un'evoluzione del sistema multi-tasking, in questo sistema i processi vengono eseguiti ciclicamente per un tempo prestabilito chiamato *quanto*. Ad ogni fine *quanto*, la CPU passerà all'esecuzione del processo successivo. Questo sistema darà quindi l'illusione di un'esecuzione parallela dei processi. Come possiamo vedere dal grafico, l'esecuzione dei programmi viene divisa nei *quanti* e ci appare spezzettata, il risultato sarà un uso ottimizzato delle risorse della CPU e un'esecuzione veloce dei programmi.



Considerazioni Finali

Tra i metodi analizzati possiamo concludere che i sistemi multi-tasking e time-sharing siano i più efficienti in quanto ci mettono entrambi 12 secondi per eseguire tutti i processi contro i 15 secondi del mono-tasking.

Bisogna però precisare che nel sistema multitasking P1 terminerà l'esecuzione già nel secondo 5 mentre utilizzando il sistema time-sharing P1 completerà la sua esecuzione nel secondo 10.

Esercizio Facoltativo

Considerando i 5 processi nella tabella sottostante, calcoliamo lo scheduling secondo il sistema round robin (time slice 12ms), compresi tempi di attesa e turnaround medi.

Processo	Tempo di arrivo (t0)	Tempo di esecuzione (tx)
P1	0	14
P2	30	16
P3	6	40
P4	46	26
P5	22	28

Time slice	Inizio	Fine	Processo	Coda a fine time slice
1	0	12	P1	P3 (6 t0)
2	12	24	P3	P1 (12) P5 (22 t0)
3	24	26	P1 FINE	P5 (22) P3 (24)
4	26	38	P5	P3 (24) P2 (30 t0)
5	38	50	P3	P2 (30) P5 (38) P4 (46 t0)
6	50	62	P2	P5 (38) P4 (46) P3 (50)
7	62	74	P5	P4 (46) P3 (50) P2 (62)
8	74	86	P4	P3 (50) P2 (62) P5 (74)
9	86	98	P3	P2 (62) P5 (74) P4 (86)
10	98	102	P2 FINE	P5 (74) P4 (86) P3 (98)
11	102	106	P5 FINE	P4 (86) P3 (98)
12	106	118	P4	P3 (98)
13	118	122	P3 FINE	P4 (118)
14	122	124	P4 FINE	

	t0 (Tempo di arrivo)	Tx (Tempo di esecuzione)	Tf (Tempo fine esecuzione)	Turnaround Tt = Tf - t0	Attesa Ta = Tt - Tx
P1	0	14	26	26	12
P2	30	16	102	72	56
P3	6	40	122	116	76
P4	46	26	124	78	52
P5	22	28	106	84	56
TOT				376	252
Media (TOT/5)				75,2	50,4