

# Scheduling della CPU - Esercizio di Pratica W4D1

## RICHIESTA

L'esercizio relativo alla W4D1 verte sullo Scheduling della CPU, e in particolar modo su gestione ed esecuzione dei processi. La traccia chiede di tener conto di quattro processi, e di individuare il modo più efficace per la gestione e l'esecuzione dei processi.

~~~

## RISOLUZIONE

Il processore, detto anche CPU, è il componente principale di un computer. Uno dei suoi compiti più importanti è l'esecuzione delle istruzioni dei programmi. La CPU ha due modalità di funzionamento:

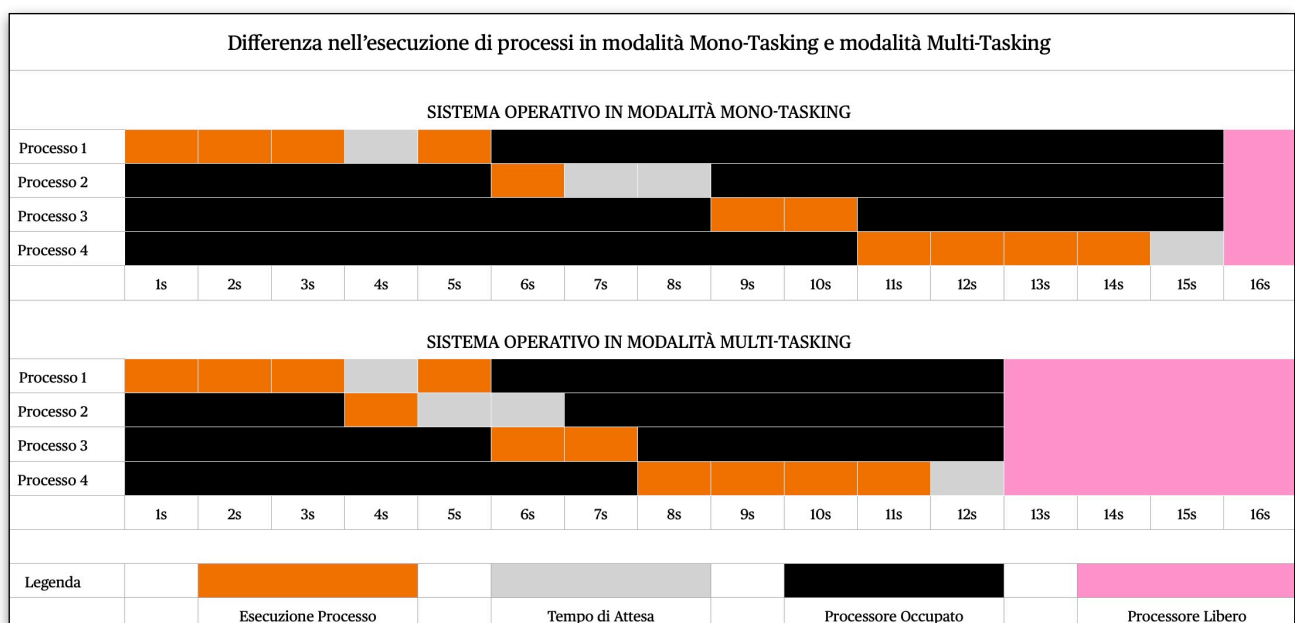
- **MONO-TASKING:** la CPU esegue un singolo compito alla volta
- **MULTI-TASKING:** la CPU può eseguire più attività contemporaneamente

In relazione all'esercizio assegnatoci, il modo più efficace per la gestione e l'esecuzione dei processi è la **modalità MULTI-TASKING**. Il suo utilizzo consente di concludere l'esecuzione dei processi in meno tempo rispetto alla modalità MONO-TASKING. Il risparmio di tempo è di 3 secondi.

~~~

## IMMAGINI

La seguente immagine mostra, in una rappresentazione schematica, l'intero processo di gestione ed esecuzione dei processi in modalità MONO-TASKING e MULTI-TASKING. Ogni quadrato del blocco corrisponde a 1 secondo trascorso



## **DETTAGLI TECNICI**

Lo scheduling del processore è il processo di gestione che il sistema operativo utilizza per assegnare i processi pronti all'esecuzione alla CPU. A seconda del sistema operativo utilizzato, lo scheduling può essere gestito in modalità MONO-TASKING o MULTI-TASKING. Ad oggi, tutti i più utilizzati sistemi operativi utilizzano la modalità MULTI-TASKING. Guardando all'applicazione di questi concetti all'esercizio:

- In modalità **MONO-TASKING** il processore esegue un unico compito alla volta, completandolo prima di passare ad altro. Questo è ben evidente nella prima parte della rappresentazione grafica, dove il processo P1 occupa i primi 5 secondi di elaborazione, nonostante ci sia un tempo di attesa di 1 secondo al quarto secondo del processo. Stessa cosa per i restanti processi. L'esecuzione completa dei quattro processi richiede un tempo di 15 secondi.
- In modalità **MULTI-TASKING**, invece, il processore permette di gestire ed eseguire più compiti in contemporanea. Questo è ben evidente guardando nuovamente al processo P1. A differenza di prima, quando P1 è in attesa (secondo 4), la CPU esegue parte del processo P2. Torna poi a completare il processo P1 (secondo 5), sfruttando l'attesa connessa al processo P2. Allo stesso modo, nell'altro secondo di attesa di P2 (secondo 6), il processore inizia l'esecuzione del processo P3, che si conclude al secondo 7. Alla fine procede con l'esecuzione del processo P4 (da 8 a 12 s). L'esecuzione completa dei quattro processi richiede un tempo di 12 secondi.

È del tutto evidente che **la modalità MULTI-TASKING è quella più efficace** per la gestione e l'esecuzione dei quattro processi. Esistono diverse tecniche e strategie di implementazione del MULTI-TASKING, con una delle principali che è quella del Time-Sharing.

~~~

## **PARTE FACOLTATIVA**

Come anticipato, il Time-Sharing è una delle principali tecniche di MULTI-TASKING. In questo caso, il processore divide il suo tempo di esecuzione in brevi intervalli, detti **QUANTI** (Quantum). All'interno della categoria Time-Sharing, trova spazio poi il **Round Robin**, un algoritmo che mette in pratica il Time-Sharing.

Nello specifico, l'algoritmo di Round Robin assegna a ogni processo in attesa di esecuzione una quantità fissa di tempo, e i processi vengono eseguiti a turno in modo circolare. Se un processo non termina entro il tempo prestabilito, viene rimesso in coda in attesa di ricevere un ulteriore turno.

L'esercizio chiede di considerare 5 diversi processi, e descriverne la gestione ed esecuzione in modalità MULTI-TASKING con utilizzo di Time-Sharing e algoritmo di Round Robin. La quantità fissa di tempo da considerare (time slice) è di 12 ms.

La tabella è la seguente:

|            |     |     |               |
|------------|-----|-----|---------------|
| Processo 1 | 0   | 12  |               |
| Processo 3 | 12  | 24  |               |
| Processo 1 | 24  | 26  | <b>FINITO</b> |
| Processo 5 | 26  | 38  |               |
| Processo 3 | 38  | 50  |               |
| Processo 2 | 50  | 62  |               |
| Processo 5 | 62  | 74  |               |
| Processo 4 | 74  | 86  |               |
| Processo 3 | 86  | 98  |               |
| Processo 2 | 98  | 102 | <b>FINITO</b> |
| Processo 5 | 102 | 106 | <b>FINITO</b> |
| Processo 4 | 106 | 118 |               |
| Processo 3 | 118 | 122 | <b>FINITO</b> |
| Processo 4 | 122 | 124 | <b>FINITO</b> |

Ecco la linea temporale:

0 ms : Parte l'esecuzione P1 - Coda (P1)

6 ms: P3 entra in coda mentre P1 è ancora in esecuzione - Coda (P1-P3)

12 ms: P1 esaurisce il suo time slice. Non è completo (mancano 2ms) e si accoda a P3 - Coda (P3-P1)

12 ms: Parte l'esecuzione di P3 - Coda (P3-P1)

22 ms: P5 entra in coda, mentre P3 è ancora in esecuzione - Coda (P3-P1-P5)

24 ms: P3 esaurisce il suo time slice. Non è completo (mancano 28ms) e si accoda a P5 - Coda (P1-P5-P3)

24 ms: Parte l'esecuzione di P1 - Coda (P1-P5-P3)

26 ms: Finisce l'esecuzione di P1 - **COMPLETO** - Coda (P5-P3)

26 ms: Parte l'esecuzione di P5 - Coda (P5-P3)

30 ms: P2 entra in coda, mentre P5 è ancora in esecuzione - Coda (P5-P3-P2)

38 ms: P5 esaurisce il suo time slice. Non è completo (mancano 16ms) e si accoda a P2 - Coda (P3-P2-P5)

38 ms: Parte l'esecuzione di P3

46 ms: Entra in coda P4, mentre P3 è ancora in esecuzione - Coda (P3-P2-P5-P4)

50 ms: P3 esaurisce il suo time slice. Non è completo (mancano 16ms) e si accoda a P4 - Coda (P2-P5-P4-P3)

50ms: Parte l'esecuzione di P2 - Coda (P2-P5-P4-P3)

62ms: P2 esaurisce il suo time slice. Non è completo (mancano 4ms) e si accoda a P3 - Coda (P5-P4-P3-P2)

62ms: Parte l'esecuzione di P5 - Coda (P5-P4-P3-P2)

74ms: P5 esaurisce il suo time slice. Non è completo (mancano 4ms) e si accoda a P2 - Coda (P4-P3-P2-P5)

74ms: Parte l'esecuzione di P4 - Coda (P4-P3-P2-P5)

86ms: P4 esaurisce il suo time slice. Non è completo (mancano 14ms) e si accoda a P5 - Coda (P3-P2-P5-P4)

86 ms: Parte l'esecuzione di P3 - Coda (P3-P2-P5-P4)

98 ms: P3 esaurisce il suo time slice. Non è completo (mancano 4ms) e si accoda a P4 - Coda (P2-P5-P4-P3)

98 ms: Parte l'esecuzione di P2 - Coda (P2-P5-P4-P3)

102 ms: Finisce l'esecuzione di P2 - **COMPLETO** - Coda (P5-P4-P3)

102 ms: Parte l'esecuzione di P5 - Coda (P5-P4-P3)

106 ms: Finisce l'esecuzione di P5 - **COMPLETO** - Coda (P4-P3)

106 ms: Parte l'esecuzione di P4 - Coda (P4-P3)

118 ms: P4 esaurisce il suo time slice. Non è completo (mancano 2 ms) e si accoda a P3 - Coda (P3-P4)

118 ms: Entra in esecuzione P3 - Coda (P3-P4)

122 ms: Finisce l'esecuzione di P3 - **COMPLETO** - Coda (P4)

122 ms: Parte l'esecuzione di P4 - Coda (P4)

124 ms: Finisce l'esecuzione di P4 - **COMPLETO** - Fine della coda