

# Meccanismi di pianificazione dell'utilizzo della CPU

## Obiettivo:

In questo esercizio descrivo i principali meccanismi di pianificazione della CPU, illustrando in generale come funzionano i sistemi mono-tasking, multi-tasking e time-sharing e svolgo la traccia facoltativa relativa al Round Robin.

## 1. Introduzione

In ottica di ottimizzazione della gestione dei processi, la pianificazione della CPU (scheduling) ha subito un'evoluzione importante. Storicamente, si è passati da sistemi molto semplici, in cui veniva eseguito un solo processo alla volta, a sistemi complessi e performanti, capaci di gestire decine o centinaia di processi contemporaneamente.

Nel nostro caso di studio, ci vengono forniti 4 processi (P1, P2, P3, P4) con tempi di esecuzione e di attesa input/output, arrivati in CPU in ordine di arrivo (P1 → P2 → P3 → P4)

Lo scopo è descrivere il metodo di scheduling di questi processi con metodo Round Robin (12 millisecondi) calcolando i tempi di attesa e durata medi.

## 2. Sistemi di pianificazione: definizioni e caratteristiche

### 2.1 Sistemi mono-tasking

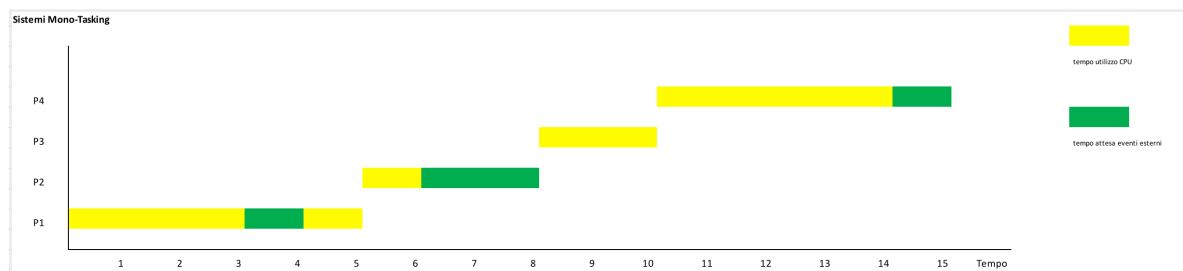
In un sistema mono-tasking, la CPU esegue un solo processo alla volta dall'inizio alla fine, senza interruzioni, finché questo non termina o passa a uno stato di attesa (ad esempio per input/output).

Vantaggio: semplice da implementare.

Svantaggio: se un processo è lungo o resta bloccato in attesa I/O, la CPU resta inutilizzata o altri processi restano in attesa inutilmente.

Questi sistemi sono ormai obsoleti e usati solo su microcontrollori o dispositivi estremamente semplici.

Rappresentazione grafica:



Nel grafico in figura possiamo vedere come ogni processo inizia quando il precedente termina. Questo non permette di sfruttare tutti i tempi di attesa che abbiamo a disposizione, aumentando la tempistica di completamento dei processi.

## 2.2 Sistemi multi-tasking

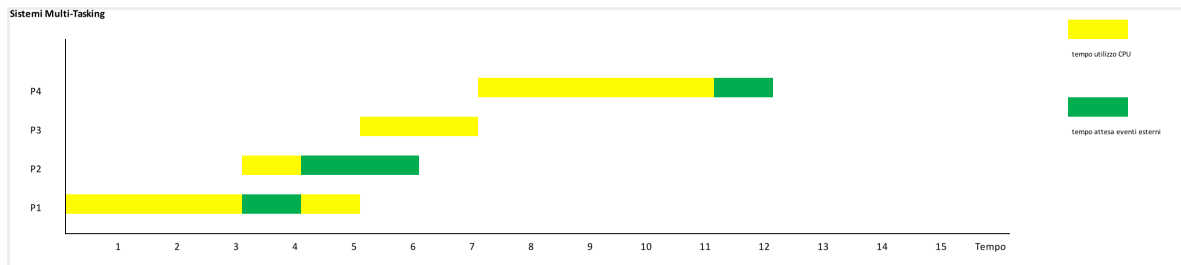
In un sistema multi-tasking, più processi possono essere caricati contemporaneamente in memoria e gestiti in modo tale da sembrare eseguiti nello stesso momento.

La CPU esegue un processo finché questo non viene bloccato (per esempio in attesa di I/O) o anticipato da un processo con priorità più alta.

Vantaggio: migliora l'utilizzo delle risorse, riduce i tempi di attesa complessivi.

Svantaggio: un processo può comunque monopolizzare la CPU se non bloccato o anticipato

Rappresentazione grafica:



Dal grafico in figura si evince subito come i tempi siano ridotti sfruttando l'attesa per iniziare altri processi.

## 2.3 Sistemi time-sharing

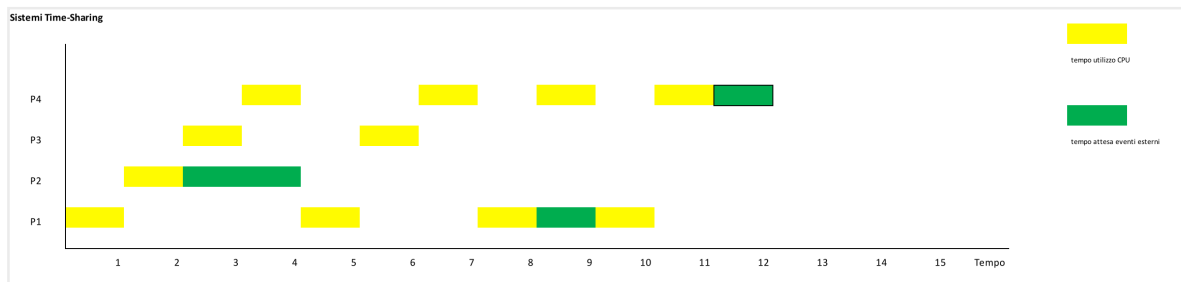
Il time-sharing è una particolare forma di multi-tasking in cui la CPU viene suddivisa in time slice o quantum (intervalli di tempo predefiniti), assegnati in modo equo ai vari processi.

Dopo ogni quantum, la CPU passa al processo successivo.

Vantaggio: garantisce che nessun processo monopolizzi la CPU, tempi di risposta più regolari.

Svantaggio: perdita di prestazioni causata da troppi cambi di contesto in poco tempo. Ogni cambio è lavoro extra per salvare o caricare stati piuttosto che eseguire il vero lavoro.

Rappresentazione grafica:



Dal grafico si evince che le tempistiche sono uguali a quelle dei multi-tasking, ma anche che il procedimento utilizzato sia diverso. Suddividendo il processo in piccole parti eseguite in modo ciclico si perde la continuità del lavoro per salvare e caricare piuttosto che l'esecuzione in sé.

### 3. Considerazioni finali

I sistemi multi-tasking e time-sharing sono, oggi, da preferire rispetto ai sistemi mono-tasking.

La differenza principale tra i due sta nella regolarità: il time-sharing, grazie alla divisione in quantum, permette un utilizzo più equo e tempi di risposta più prevedibili per tutti i processi.

In questo esercizio, dato che i processi arrivano in ordine e hanno tempi e attese differenti, un sistema time-sharing potrebbe essere particolarmente indicato per evitare che processi lunghi penalizzino quelli più brevi.

### 4. Parte facoltativa: Round Robin

Obiettivo:

Nella traccia facoltativa ci viene chiesto di considerare 5 processi (P1, P2, P3, P4, P5) e di schedarli usando la politica Round Robin con un quantum di 12 millisecondi.

Dobbiamo calcolare i tempi di attesa medi e i tempi di durata medi

Processo	Tempo di arrivo ( $t_0$ )	Tempo di esecuzione ( $T_x$ )
P1	0	14
P2	30	16
P3	6	40
P4	46	26
P5	22	28

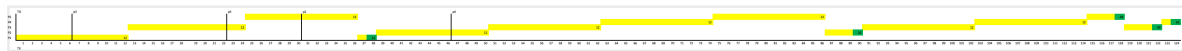
#### 4.1 Cos'è il Round Robin

Il Round Robin è un algoritmo di time-sharing che assegna a ogni processo un intervallo di tempo fisso (quantum) e li esegue in ordine circolare.

Ogni processo viene interrotto allo scadere del quantum e messo in coda se

non ha finito, per riprendere al giro successivo.

Questo garantisce che tutti i processi abbiano accesso equo alla CPU.



Nel grafico sopra riportato ho rappresentato lo svolgimento di ogni processo, dove il colore giallo rappresenta la durata di esecuzione di un programma e il verde il momento nel quale il programma finisce. Le linee verticali rappresentano il momento in cui un programma (segnalato di fianco la linea) è pronto per iniziare. Dal grafico sono riuscita a ricavare tutti i valori riportati nella tabella seguente

Time Slice	Inizio	Fine	Processo	Processo in attesa
1	0	12	P1	P3(6)
2	13	24	P3	P1(12) - P5(22)
3	25	36	P5	P1(12)-P2(30)-P3(24)
4	37	38	P1	P2(30)-P3(24)-P5(36)
5	39	50	P2	P3(24)-P4(46)-P5(36)
6	51	62	P3	P2(50)-P4(46)-P5(36)
7	63	74	P4	P2(50)-P3(62)-P5(36)
8	75	86	P5	P2(50)-P3(62)-P4(74)
9	87	90	P2	P3(50)-P4(74)-P5(86)
10	91	102	P3	P4(74)-P5(86)
11	103	114	P4	P3(102)-P5(86)
12	115	118	P5	P3(102)-P4(114)
13	119	122	P3	P4(114)
14	123	124	P4	

Per ogni processo in esecuzione ci sono altri in attesa e ho indicato da che momento. Ho rappresentato l'inizio e la fine di ognuno, che in modo ciclico vengono eseguiti, alternandosi in base ai tempi.

## 5. Conclusioni

La pianificazione della CPU è una componente fondamentale dei sistemi operativi.

Nel caso in esame, il sistema time-sharing o Round Robin rappresentano le scelte più adatte per bilanciare tempi di attesa e utilizzo della CPU.

Il sistema mono-tasking è ormai superato e inefficiente, mentre il multi-tasking senza time slice rischia comunque di penalizzare i processi più brevi.

Il Round Robin con un quantum adeguato garantisce equità ed è facile da implementare.