

Esercitazione 9

CPU Scheduling 1

Ettore Farris - 15/11/2023

1) Descrizione sintetica dell'esercitazione

L'esercitazione è finalizzata comprendere ad alto livello gli approcci di scheduling della CPU, ovvero quelle operazioni finalizzate ad ottimizzare l'uso del processore del computer mediante la gestione dei processi.

Definizione di processo:

Un processo è l'attività di esecuzione di un programma in modo sequenziale, ovvero un **compito** che il processore del computer deve portare a termine su richiesta dell'utente. Quando il computer deve svolgere numerosi compiti nello stesso lasso di tempo, ci si pone davanti il problema dello scheduling.

Definizione di scheduling

Lo scheduling è la **pianificazione** dei compiti che il computer deve eseguire. Aiuta a decidere quale processo eseguire, quando eseguirlo e per quanto tempo. Lo scheduling in un sistema operativo cerca di ottimizzare l'utilizzo del processore in modo che i processi vengano "smaltiti" in modo efficiente. Ci sono diversi **algoritmi** di scheduling che possono essere utilizzati per gestire i processi, e la **scelta** dipende dalle caratteristiche del sistema e dagli **obiettivi di prestazione**. Prima di adottare un algoritmo anziché un altro, si deve rispondere quindi alla domanda "cosa vogliamo ottimizzare?" (ad. es, tempo di esecuzione totale, tempo di attesa medio per processo ecc...).

Consegna dell'esercizio:

"L'esercizio di oggi verte sui meccanismi di pianificazione 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.

Traccia:

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" sulle ordinate il nome del processo".

Processo	Tempo Esecuzione	Tempo Attesa	Tempo Esecuzione dopo Attesa	Tempo Totale
P1	3	2	1	6
P2	2	1	0	3
P3	1	0	0	1
P4	4	1	2	7

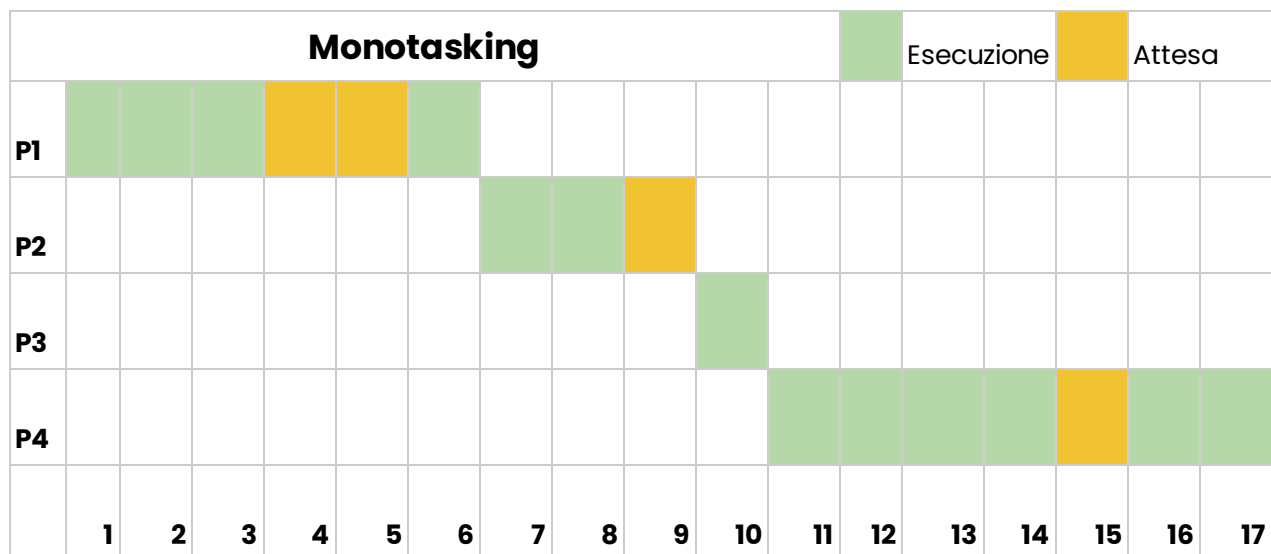
Un esempio reale di processo

Per comprendere meglio l'esercizio, ipotizziamo che il processo P1 sia un semplice programma che chiede all'utente di cliccare il tasto "Invia" per poter mandare un ping a un altro utente. Il processo viene terminato quando il ping viene inviato con successo. L'esecuzione di questo programma avviene secondo questo schema:

- Il processo, una volta lanciato, carica la schermata in *3 secondi*, impiegando quindi il "Tempo di esecuzione" (prima colonna) e chiede all'utente di cliccare "OK" per continuare.
- Il processo, per poter essere terminato, ha bisogno di un evento esterno, ovvero il clic dell'utente sul tasto "invia". Il lasso di tempo che intercorre tra il caricamento della schermata e l'azione dell'utente rappresenta il tempo di attesa, per ipotesi di 2 secondi nel processo 1.
- Dopo il clic su "invia", il ping viene inviato e il processo viene terminato dopo 1 secondo, ovvero il "Tempo Esecuzione dopo Attesa" (cioè dopo che l'utente ha cliccato invia).

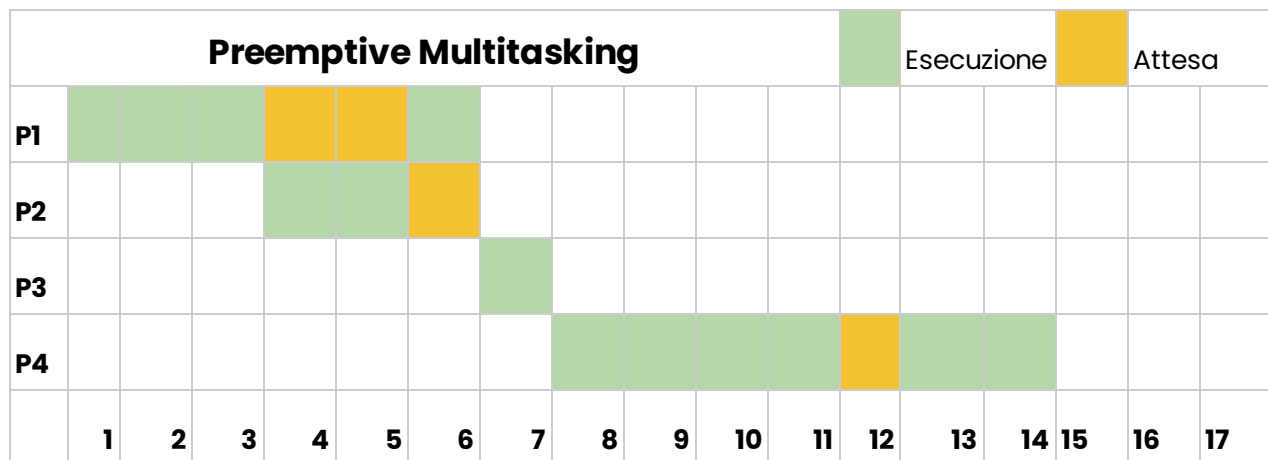
2) Approccio monotasking

Nell'approccio monotasking, i processi vengono schedulati in ordine di arrivo. Pertanto, un processo deve attendere il completamento di quello precedente per poter essere eseguito. E' un approccio inefficiente poiché la CPU non sfrutta i tempi di attesa per poter eseguire altri processi nel frattempo. Nel nostro esercizio, il tempo totale di esecuzione di tutti i processi è di **17 secondi**.



3) Approccio preemptive multitasking

Il multitasking consente alla CPU di eseguire più processi contemporaneamente. La pianificazione con prelazione (preemptive multitasking) consente di sfruttare i tempi di attesa di un processo per poterne eseguire un altro. Nell'esempio, il tempo di esecuzione di tutti i processi è di **14 secondi**.



4) Approccio time-sharing

Il time-sharing è un'evoluzione dei sistemi multitasking. I processi vengono eseguiti in modo ciclico, vale a dire ad ogni processo viene data un "po" di CPU a turno per ogni processo in coda immediatamente disponibile. Ogni processo viene eseguito per un una piccola porzione di tempo detto *quanto*, allo scadere del quale viene messo in pausa per far eseguire un quanto del processo successivo e così via sino a terminare tutti i processi.

Nell'esercizio, i tempi di attesa, essendo legati a fattori esterni come un click di un utente, sono fattori esogeni e pertanto non potrebbero essere quantizzati. Un utente clicca sul tasto "Invia" quando vuole e la sua azione non è soggetta allo scheduling della CPU. Il tempo di esecuzione totale è di **13 secondi**.

Time-sharing												Esecuzione	Attesa				
P1																	
P2																	
P3																	
P4																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

5) Conclusioni

L'approccio migliore per svolgere l'esercizio, ipotizzando come obiettivo quello di eseguire tutti i processi nel minor tempo possibile, è quello ***time-sharing***.