

Scheduling SPN (Shortest Process Next)

Il cassiere in un negozio trova la persona con il minor numero di articoli e controlla prima i loro articoli.

Il concetto che c'è dietro a questo algoritmo è quello di **CPU Burst**: un'applicazione quando prende la CPU può eventualmente chiamare il Kernel in attesa che magari qualche dispositivo completi la sua esecuzione, e questa applicazione rilascia la CPU, per poi riprendere la CPU più avanti per chiamare di nuovo il Kernel. La quantità di tempo in cui l'applicazione ha preso la CPU FINO AL rilascio e quindi alla chiamata verso il Sistema Operativo, si chiama CPU BURST.

QUANTO È DIVERSO DA CPU BURST. IL QUANTO VIENE SCELTO DAL SISTEMA

In questo algoritmo tutti i processi "Ready" li possiamo ORDINARE in modo CRESCENTE in base al CPU BURST.

Esso prevede che venga eseguito sempre il processo con il tempo di esecuzione più breve tra quelli in attesa. Si può dimostrare che questo algoritmo è ottimale, in quanto consente di ottenere sempre il valore più basso di tempo d'attesa medio. Sfortunatamente non è possibile applicarlo, in quanto non è possibile conoscere anticipatamente quanto durerà l'esecuzione del processo. Tuttavia si può provare a predirlo, in quanto è probabile che sia simile ai precedenti.

Prendiamo ad esempio che siano stati sottomessi contemporaneamente i seguenti processi, con la rispettiva durata di esecuzione in millisecondi:

- p1: 10
- p2: 2
- p3: 6
- p4: 4

I processi vengono eseguiti nel seguente ordine: $p2 \rightarrow p4 \rightarrow p3 \rightarrow p1$

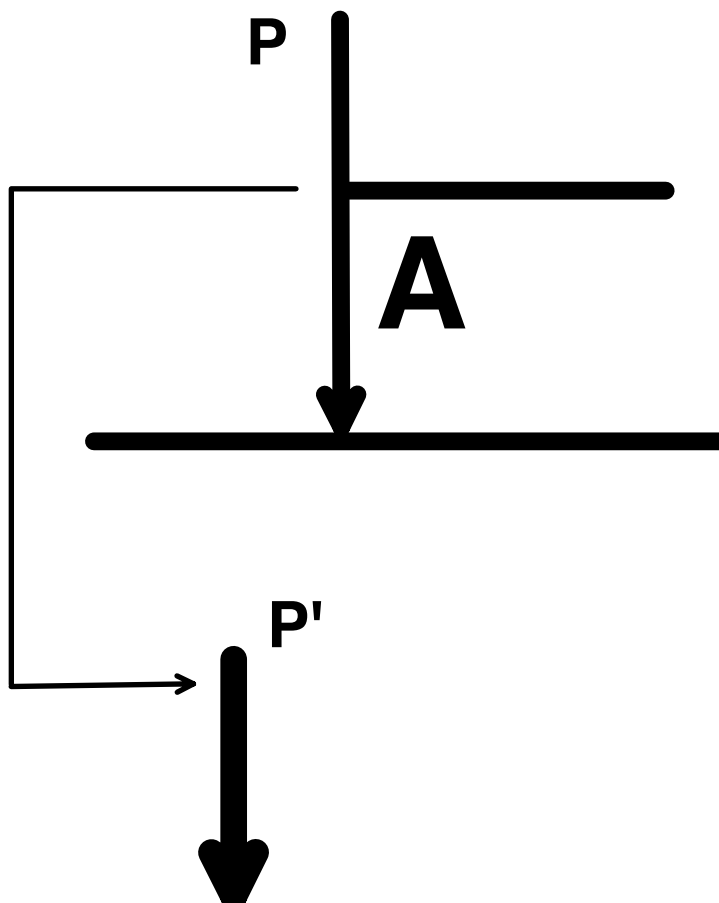
Non tenendo in considerazione il tempo necessario per il **context switch**, il processo p2 non attende nulla, perché entra subito in esecuzione, p4 attende 2 millisecondi, perché entra in esecuzione subito dopo p2, quindi p3 attende 6 millisecondi e p1 ne attende 12. Il tempo medio d'attesa è pari a $(0+2+6+12)/4 = 5$ millisecondi.

Sappiamo predire il CPU BURSTS di un processo, in base a ciò che è successo precedentemente: ossia quali sono i suoi CPU Burst precedenti.

C'è una variante dell'algoritmo che si chiama SRTN - Shortest Remaining Time Next, che ha la prelazione: ossia abbiamo assegnato la CPU ad un processo P, ad un certo istante esso comincia a prendere il controllo della CPU, idealmente lui voleva andare avanti fino ad un certo istante di tempo ma viene interrotto prima perché è arrivato un interrupt di un dispositivo di I/O e quindi uno dei processi che stavamo gestendo è diventato nuovamente ready, se quello che è diventato ready ha un prossimo CPU-BURST che è più corto di questo residuo A che P deve ancora spendere, allora assegniamo la CPU a lui.

Supponiamo che questo processo sia P', a questo istante la CPU va a P', E QUINDI LA PARTE RIMANENTE D'ESECUZIONE DI P NON VERRÀ MODIFICATA ORA.

L'algoritmo SRTF (*Shortest Remaining Time First*) si differenzia per il fatto che, quando viene sottomesso un nuovo processo la cui durata è minore del tempo necessario al processo in esecuzione per portare a termine la propria sessione, lo scheduler provvede ad effettuare un **context switch** e assegna l'uso della CPU al nuovo processo (prelazione - algoritmo preemptive). Il vantaggio è una gestione ottimale del **tempo di attesa** medio, in quanto processi dalla durata di esecuzione ridotta vengono eseguiti molto velocemente. Tra gli svantaggi potenziali, come avviene per l'SRT, vi è il fenomeno della **starvation** dei processi in quanto processi con tempi di esecuzione rimanenti lunghi potrebbero rimanere in attesa indefinitamente nel caso venissero continuamente aggiunti processi di durata rimanente inferiore.



Vantaggi

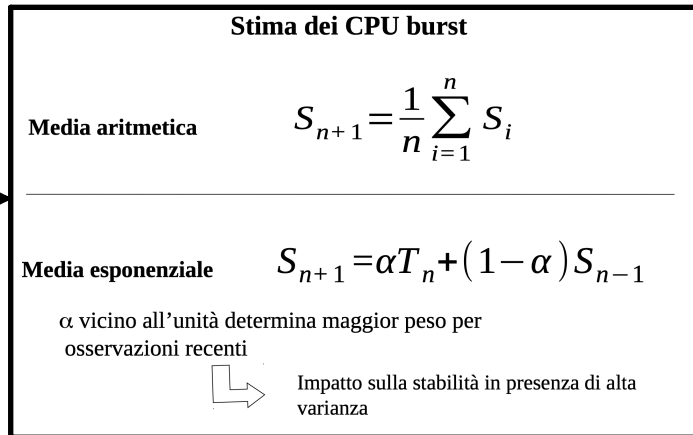
- minimizza il tempo di attesa, e di conseguenza il tempo di turnaround
- relativamente adeguato per la gestione di processi interattivi in caso di prelazione
- in generale non causa sottoutilizzo dei dispositivi in caso di prelazione

Svantaggi

Starvation

- necessita di meccanismi di predizione della lunghezza dei CPU Burst
- può provocare starvation a causa del particolare trattamento della priorità

La stima del CPU BURST
prossimo di un processo è
la media aritmetica o la
media esponenziale
(sotto).



Si può sbagliare la previsione del CPU BURST.

COME RISOLVIAMO LA STARVATION?