ESERCITAZIONE 7

Algoritmi di schedulazione

Fattibilità di sistemi real-time

Gestione della memoria e del disco

Algoritmi di schedulazione

Algoritmi di schedulazione (1)

1) Supponendo di utilizzare l'algoritmo first-come first-served per lo scheduling in un sistema batch, se arrivano 4 job (in ordine A, poi B dopo 2ms, C dopo 4ms e D dopo 5ms) con i seguenti tempi di esecuzione A=8, B=4, C=2 e D=4 quali sono i tempi di turnaround? Qual è il tempo medio di turnaround e di attesa?

Algoritmi di schedulazione (1) Idea (1)

- First come – First served: Ordine in cui chi arriva prima viene eseguito per primo.
- Tempo turnaround: tempo che intercorre dal momento in cui un processo entra in coda per essere eseguito fino al termine della sua esecuzione.
- Tempo di attesa: tempo che un processo trascorre in coda prima di essere eseguito.
- TA = Time of arrival = tempo assoluto di ingresso in coda di un processo
- TE = Time of execution = tempo necessario per l'esecuzione di un processo

Fattibilità di sistemi real-time

Fattibilità di sistemi real-time (1)

1)Supponendo di dover valutare la fattibilità di un sistema soft real-time con eventi periodici P0=300μs, P1=900μs, P2=45ms, P3=150ms e rispettivi tempi di elaborazione C0=25μs, C1=300μs, C2=9ms, C3=50μs il sistema è sostenibile? Se si aggiunge un nuovo evento periodico P4=110ms, quanto è il tempo massimo di elaborazione affinché il sistema rimanga sostenibile?

Fattibilità di sistemi real-time (1) Idea (1)

- **Deadline**: tempo entro il quale le operazioni devono necessariamente essere completate
- Sistemi non obbligatoriamente **veloci** ma piuttosto **affidabili**.
- Eventi **periodici**: eseguiti a ciclo continuo con cadenza ben definita.
- Eventi periodici hanno:
  - **Periodo**: tempo entro il quale l'evento deve essere ripetuto (Pi)
  - **Tempo d'esecuzione** (Ci)
- Sistema RT sostenibile (con m eventi periodici):
  - Vale $\sum_{i=1}^{m} (\frac{C_i}{P_i}) \leq 1$

Gestione della memoria e del disco

Gestione della memoria e del disco (1)

1) Nell'ambito della gestione della memoria con liste concatenate, considerando una lista parzialmente piena di questo tipo (per ogni tripla, il primo elemento è un flag per capire se si tratta di un buco o un processo, il secondo è l'indirizzo di partenza e il terzo è la lunghezza dell'elemento):

P,0,6,→H,6,3,→P,9,8→P,17,4,→H,21,2,→P,23,6,→H,29,4,X

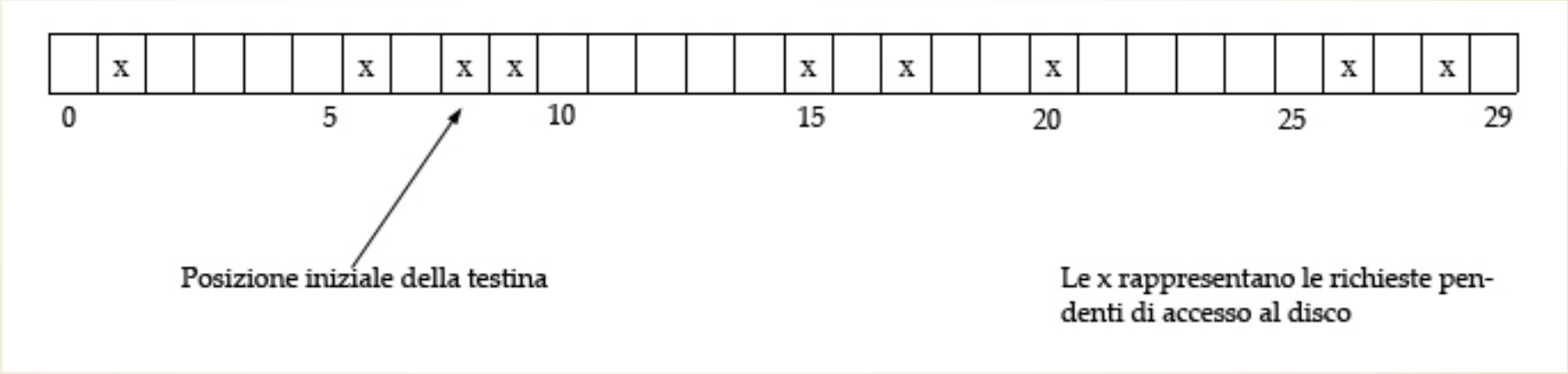
Dove viene posizionato il nuovo processo P che occupa 4 blocchi? Dove verrebbe posizionato se invece ne occupasse solo 2, secondo gli algoritmi first fit e best fit?

Gestione della memoria e del disco (1) Idea (1)

- Lettera P rappresenta un processo
- Lettera H rappresenta un "buco" di un certo numero di blocchi
- **First-fit**: inserisce il nuovo processo nella prima posizione possibile dall'inizio della lista
- **Best-fit**: inserisce il nuovo processo nella posizione tale da contenerlo nel modo migliore e, dunque, nel buco più piccolo disponibile per contenere il processo.
- Ref: "I moderni sistemi operativi", *Tanenbaum,* pagina 183 e seguenti (nell'edizione in mio possesso), paragrafo 4.2.2. Gestione della memoria con liste concatenate.

Gestione della memoria e del disco (2)

2) Nell'ambito della schedulazione delle richieste al disco, si consideri la situazione delle richieste pendenti e della testina illustrata in figura. Come si comporterà la testina nel caso in cui l'algoritmo di schedulazione utilizzato sarà l'SSF (Shortest Seek First)? Come si comporterà nel caso in cui sarà l'algoritmo dell'ascensore?



Gestione della memoriae del disco (2) Idea (1)

- Ref: I moderni sistemi operativi, pag 296 e seguenti, paragrafo 5.4.3. Gli algoritmi di schedulazione del braccio del disco.
- Algoritmo **SSF** (Shortest Seek First): risolte per prima le richieste più vicine alla posizione attuale della testina. La posizione successiva k è stabilita dalla formula seguente: $\underset{k}{argmin} |(P - R_k)|$
- Algoritmo dell'**ascensore**: risolte tutte le richieste in una direzione (salita, ad esempio), poi risolte quelle nell'altra direzione. Garantisce la *fairness* del sistema, rendendo impossibile la *starvation*.