ESERCITAZIONE 7 - Soluzioni

Algoritmi di schedulazione

Fattibilità di sistemi real-time

Gestione della memoria, paginazione e gestione 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) Soluzioni (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

Algoritmi di schedulazione (1) Soluzioni (2)

- Tabella con TA e TE di ogni processo dell'esercizio.

PROCESSO	TA	TE
A	0	8
B	2	4
C	4	2
D	5	4

Algoritmi di schedulazione (1) Soluzioni (3)

- Strategia FCFS (FIFO): ordine A-B-C-D
- TS = Tempo assoluto di inizio di esecuzione di ogni processo
- TF = tempo assoluto di fine dell'esecuzione di un processo

Processo	TA	TE	TS	TF
A	0	8	0	8
B	2	4	8	12
C	4	2	12	14
D	5	4	14	18

Algoritmi di schedulazione (1) Soluzioni (4)

- TT = Tempo Turnaround = TF-TA
- TW = Tempo di attesa = TS-TA

Processo	TA	TE	TS	TF	TW	TT
A	0	8	0	8	0	8
B	2	4	8	12	6	10
C	4	2	12	14	8	10
D	5	4	14	18	9	13

Algoritmi di schedulazione (1) Soluzioni (5)

- Tempo medio (entambi i casi): somma tempi e divisione per numero processi.

$$TW(medio) = \frac{0+6+8+9}{4} = 5.75$$

$$TT(medio) = \frac{(8+10+10+13)}{4} = 10.25$$

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) Soluzioni (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 $\sum_{i=1}^{m} (\frac{C_i}{P_i}) \leq 1$

Fattibilità di sistemi real-time (1) Soluzioni (2)

- Sostituzione valori all'interno della formula:

$$\frac{C_0}{P_0}+\frac{C_1}{P_1}+\frac{C_2}{P_2}+\frac{C_3}{P_3}=\frac{25}{300}+\frac{300}{900}+\frac{9}{45}+\frac{50}{150}=\frac{57}{60}\leq 1$$

Ref testo: Moderni sistemi operativi, paragrafo 2.5.4. *Schedulazione nei sistemi real time*, pag 135 e successive

Fattibilità di sistemi real-time (1) Soluzioni (3)

- Massimo tempo di esecuzione per il nuovo processo P4.

$$\frac{C_0}{P_0}+\frac{C_1}{P_1}+\frac{C_2}{P_2}+\frac{C_3}{P_3}+\frac{C_4}{P_4}=\frac{57}{60}+\frac{C_4}{110}\leq 1$$

$$\frac{C_4}{110}\leq 1-57/60$$

$$C_4\leq \left(\frac{3}{60}\right)* 110 \Rightarrow C_4\leq 5.5$$

Massimo tempo di esecuzione: 5.5ms

Gestione della memoria, paginazione e gestione del disco

Gestione della memoria, paginazione e gestione 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, paginazione e gestione del disco (1) - Soluzioni (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, paginazione e gestione del disco (1) - Soluzioni (2)

- Caso "base", con processo di 4 blocchi
- P,0,6,→H,6,3,→P,9,8→P,17,4,→H,21,2,→P,23,6,→H,29,4,X
- Spazio necessario per contenere un processo di dimensione 4 blocchi solo a partire dall'indirizzo 29.

Gestione della memoria, paginazione e gestione del disco (1) - Soluzioni (3)

- Caso first fit, con processo di 2 blocchi
- P,0,6,→H,6,3,→P,9,8→P,17,4,→H,21,2,→P,23,6,→H,29,4,X
- Primo spazio sufficiente per contenere il processo inizia all'inidirizzo 6.
- Si ottiene:
- P,0,6,→P,6,2,→H,8,1,→ P,9,8→P,17,4,→H,21,2,→P,23,6,→H,29,4,X

Gestione della memoria, paginazione e gestione del disco (1) - Soluzioni (4)

- Caso first fit, con processo di 2 blocchi
- P,0,6,→H,6,3,→P,9,8→P,17,4,→H,21,2,→P,23,6,→H,29,4,X
- Miglior spazio per contenere il processo inizia all'inidirizzo 21.
- Si ottiene:
- P,0,6,→H,6,3,→P,9,8→P,17,4,→P,21,2,→P,23,6,→H,29,4,X

Gestione della memoria, paginazione e gestione del disco (2)

2) Nell'ambito del rimpiazzamento delle pagine, supponendo di usare l'algoritmo not recently used (NRU), e seguendo la tabella delle pagine in figura, nel caso in cui avvenga un *page fault*, quale sarà la pagina che verrà posta su disco? Perché? Se invece considerassimo la tabella come una lista ordinata (nell'ordine A,…,G), cosa succederebbe per l'algoritmo second chance? Quale pagina sarebbe rimpiazzata?

Pagina	R	M
A	1	0
B	1	1
C	1	0
D	0	1
E	0	0
F	1	1
G	1	0

Gestione della memoria, paginazione e gestione del disco (2) - Soluzioni (1)

- Algoritmo **NRU** (Not Recently Used):
  - Bit R: posto a 1 ogni qual volta pagina sia riferita (cioè scritta o letta)
  - Bit M: posto a 1 ogni qual volta pagina sia modificata.
- Reset a 0 da software per entrambi i bit.
- Priorità :
  - [Classe 0]: R=0, M=0, (classe più conveniente da swappare)
  - [Classe1]: R=0, M=1,
  - [Classe2]: R=1, M=0,
  - [Classe3]: R=1, M=1 (classe meno conveniente da swappare)
- Ref: I moderni sistemi operativi", pag 197 e successive, capitolo 4.4. Algoritmi di rimpiazzamento delle pagine.

Gestione della memoria, paginazione e gestione del disco (2) - Soluzioni (2)

- Algoritmo **second chance** (della seconda opportunità):
- Pagine in lista per ordine di ingresso
- Bit M non considerato
- Algoritmo:
  - Scorrere la lista
  - Se R = 1, porre R=0, reinserire pagina in fondo a lista
  - Se R=0, effettuare swap della pagina corrispondente.

Gestione della memoria, paginazione e gestione del disco (2) - Soluzioni (3)

- Soluzione algoritmo LRU:
  - Una sola pagina appartiene alla Classe 0, con massima priorità di swap.
  - La pagina è E (M=0, R=0).

Gestione della memoria, paginazione e gestione del disco (2) - Soluzioni (4)

- Soluzione algoritmo second chance:
  - A ha R=1: reset a 0 e spostamento in fondo alla lista,
  - B ha R=1: reset a 0 e spostamento in fondo alla lista,
  - C ha R=1: reset a 0 e spostamento in fondo alla lista,
  - D ha R=0: swap su disco
  - Situazione finale in figura.

Pagina	R
E	0
F	1
G	1
A	0
B	0
C	0

Gestione della memoria, paginazione e gestione del disco (3)

3) Utilizzando l'algoritmo di paginazione Aging (con contatore a 8 bit), si considerino le seguenti 4 pagine, con valore di bit R, rispettivamente, 1000. Ai cicli successivi, i valori sono 0111, 0011, 1101, 1100, 0001, 1010 e 1110. Si forniscano i valori dei 4 contatori dopo l'ultimo intervallo, specificando quale pagina viene spostata su disco.

Gestione della memoria, paginazione e gestione del disco (3) - Soluzioni (1)

- Algoritmo **Aging**:
  - Tabella con andamento dei bit R per ogni pagina.
  - Ad ogni nuova scrittura, shift verso destra.
  - Swap di pagina con valore binario minore nella riga.

Gestione della memoria, paginazione e gestione del disco (3) - Soluzioni (2)

- Algoritmo **Aging**: soluzione

Pag								
P_0	1	1	0	1	1	0	0	1
P_1	1	0	0	1	1	0	1	0
P_2	1	1	0	0	0	1	1	0
P_3	0	0	1	0	1	1	1	0

Valore binario minore: P_3. P_3 spostata su disco

Gestione della memoria, paginazione e gestione del disco (4)

4) Usando l'algoritmo di paginazione LRU (least recently userd), nel caso in cui la macchina abbia 4 pagine fisiche, e nel caso in cui le pagine siano usate nell'ordine:

0 1 2 3 1 0 3 2

Stabilire quale pagina verrà spostata su disco, tenendo presente la matrice delle pagine.

Gestione della memoria, paginazione e gestione del disco (4) - Soluzioni (1)

- Algoritmo **LRU** (Least Recently Used):
  - Estrarre la pagina usata più lontana nel passato
  - Creare tabella con entrambe le dimensioni pari al numero di pagine
  - Inizializzazione della tabella con tutti valori 0,
  - All'uso di una pagina:
    - Porre a 1 tutta la riga corrispondente alla pagina
    - Porre a 0 tutta la colonna corrispondente alla pagina
  - Estrarre la pagina corrispondente al valore binario minore sulla riga.

Gestione della memoria, paginazione e gestione del disco (4) - Soluzioni (2)

- Algoritmo **LRU** (Least Recently Used):
  - Svolgimento

ID	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

Gestione della memoria, paginazione e gestione del disco (4) - Soluzioni (3)

PAG 0

ID	0	1	2	3
0	0	1	1	1
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

PAG 1

ID	0	1	2	3
0	0	0	1	1
1	1	0	1	1
2	0	0	0	0
3	0	0	0	0

Gestione della memoria, paginazione e gestione del disco (4) - Soluzioni (4)

PAG 2

ID	0	1	2	3
0	0	0	0	1
1	1	0	0	1
2	1	1	0	1
3	0	0	0	0

PAG 3

ID	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

Gestione della memoria, paginazione e gestione del disco (4) - Soluzioni (5)

PAG 1

ID	0	1	2	3
0	0	0	0	0
1	1	0	1	1
2	1	0	0	0
3	1	0	1	0

PAG 0

ID	0	1	2	3
0	0	1	1	1
1	0	0	1	1
2	0	0	0	0
3	0	0	1	0

Gestione della memoria, paginazione e gestione del disco (4) - Soluzioni (6)

PAG 3

ID	0	1	2	3
0	0	1	1	0
1	0	0	1	0
2	0	0	0	0
3	1	1	1	0

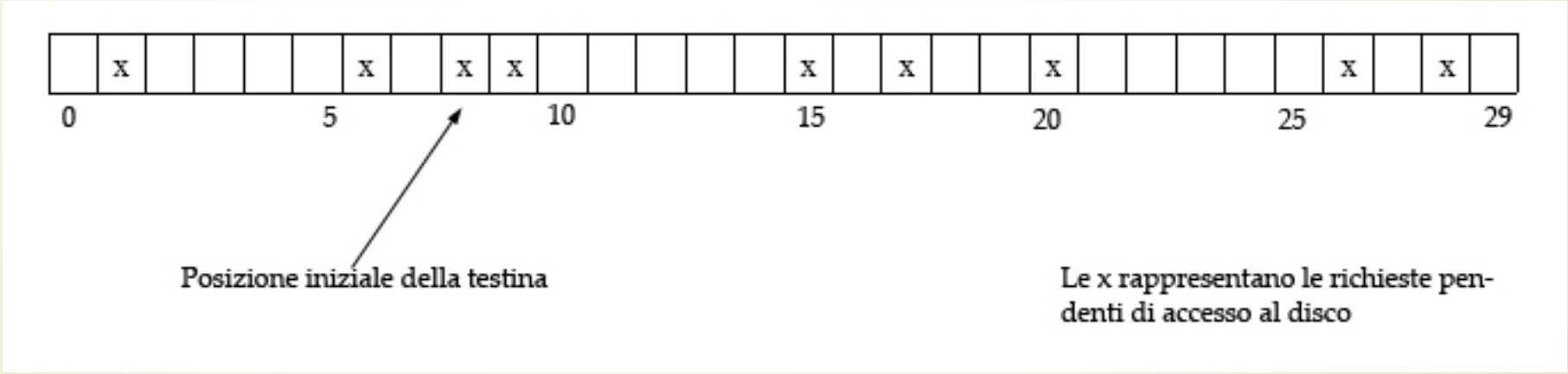
PAG 2

ID	0	1	2	3
0	0	1	0	0
1	0	0	0	0
2	1	1	0	1
3	1	1	0	0

Pagina da estrarre: Pagina 1, con valore binario 0000

Gestione della memoria, paginazione e gestione del disco (5)

5) 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 memoria, paginazione e gestione del disco (5) - Soluzioni (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*.

Gestione della memoria, paginazione e gestione del disco (5) - Soluzioni (2)

- Algoritmo **SSF:** soluzione:
  - 8, 9, 6, 1, 15, 17, 20, 26 e 28.
- Algoritmo dell'**ascensore**: soluzione:
  - 8, 9, 15, 17, 20, 26, 28, 6, 1