

Sistemi Operativi

7 luglio 2011

Compitino II A

Si risponda ai seguenti quesiti, giustificando le risposte.

1. Nella fase di creazione di nuovi processi che vantaggi offre la tecnica del Copy-on-Write (COW)?

Risposta: (4 punti) Il meccanismo del copy-on-write consente di condividere sia il codice che i dati a due processi. Ciò risulta utile in particolare in ambiente Unix in seguito ad una `fork` dato che sia il processo padre che il figlio inizialmente condividono sia il codice che i dati. Ciò avviene facendo in modo che le page table dei due processi puntino alle stesse pagine in memoria (in questo modo non è necessario duplicare le pagine dei dati al momento della `fork`). Tuttavia le pagine contenenti i dati vengono marcate come pagine di sola lettura. Quando uno dei due processi tenta di modificare una pagina dati, avviene una violazione della protezione di sola lettura e si solleva una trap al sistema operativo. La routine di gestione della trap a questo punto copia effettivamente la pagina ed aggiorna le page table in modo che ogni processo punti correttamente alla propria copia (viene anche tolto il flag di sola lettura ovviamente). In questo modo la dispendiosa attività di duplicare le pagine dei dati viene ritardata fino al primo tentativo di modifica da parte di uno dei processi coinvolti, facendo in modo che la `fork` venga eseguita molto velocemente.

2. (a) Si descriva il concetto di *working set* $WS(t, \Delta)$, all'istante t con intervallo Δ .
 (b) Si consideri la seguente stringa di riferimenti (partendo con $t = 0$):

6 4 2 7 6 5 5 5 1 6 4

Cosa è $WS(10, 6)$, ossia dopo l'ultimo accesso?

- (c) Nel precedente esempio quanti page fault ci sono complessivamente con $\Delta = 3$ (supponendo che in ogni istante si mantenga in memoria esattamente il solo working set)?

Risposta:

- (a) (3 punti) Il working set è un'approssimazione della località del processo, ossia è l'insieme di pagine "attualmente" riferite. In generale $WS(t, \Delta)$ = insieme delle pagine riferite negli accessi $[(t - \Delta + 1), t]$.
 (b) (3 punti) $WS(10, 8) = \{1, 4, 5, 6\}$.
 (c) (3 punti) Si verificano 9 page fault, come risulta dalla seguente simulazione dell'algoritmo:

6	4	2	7	6	5	5	5	1	6	4
6	4	2	7	6	5	5	5	1	6	4
	6	4	2	7	6	6		5	1	6
	6	4	2	7				5	1	
p	p	p	p	p	p			p	p	p

3. Si consideri un processo che generi la seguente stringa di riferimenti alle pagine virtuali:

0 2 1 0 4 0 2 1 1 0 5 3 2

- (a) Se il processo ha 4 frame, gestiti LRU, quanti page fault vengono generati?
 (b) Qual è il numero minimo di frame necessario per minimizzare i page fault?

Risposta:

- (a) (3 punti) Simuliamo il funzionamento di LRU nel caso della reference string data:

	0	2	1	0	4	0	2	1	1	0	5	3	2
		0	2	1	0	4	0	2	2	1	0	5	3
			0	2	1	1	4	0	0	2	1	0	5
					2	2	1	4	4	4	2	1	0
											4	2	1
												4	4
	P	P	P		P						P	P	P

Si verificano quindi sette page fault.

Sistemi Operativi

7 luglio 2011

Compitino II A

- (b) (3 punti) Il minimo numero di page fault è 6 page fault (perché il processo accede a 6 pagine). Per determinare il numero minimo di frame per avere solo 6 page fault, si può sfruttare la *distance string*, che nel caso in questione risulta essere la seguente:

$\infty \ \infty \ \infty \ 3 \ \infty \ 2 \ 4 \ 4 \ 1 \ 3 \ \infty \ \infty \ 5$

Si ricorda che la *distance string* rappresenta la distanza fra la posizione di una pagina nel modello e la prima posizione, ovvero, quella nella prima riga della matrice (contando anche la casella di partenza) nel momento in cui la pagina stessa viene riferita. Se una pagina non è presente nella matrice, allora la sua distanza, quando viene riferita è ∞ .

Indichiamo ora con C_i il numero di volte che il numero i compare nella *distance string*; nel caso in questione abbiamo: $C_1 = 1$, $C_2 = 1$, $C_3 = 2$, $C_4 = 2$, $C_5 = 1$, $C_\infty = 6$. Indicando poi con m il numero di frame e con n il numero più grande che compare nella *distance string*, indichiamo con $F_m = \sum_{k=m+1}^n C_k + C_\infty$ il numero di page fault che si verificano con m frame e con la *reference string* data. L'intuizione è la seguente: se ho a disposizione m frame i page fault saranno provocati dai riferimenti a pagine che "distano" almeno $m + 1$ dal top della matrice e dal numero di ∞ (ovvero da riferimenti a pagine non ancora presenti nel modello). Nel nostro caso abbiamo: $F_1 = 12$, $F_2 = 11$, $F_3 = 9$, $F_4 = 7$, $F_5 = 6$; quindi il numero minimo di frame che minimizza i page fault è 5.

4. Si illustrino i passi eseguiti dai driver dei dispositivi. Cosa significa che i driver devono essere rientranti?

Risposta: (4 punti) Le operazioni eseguite dai driver di un dispositivo di I/O sono le seguenti:

1. Controllare i parametri passati.
2. Accodare le richieste in una coda di operazioni (soggette a scheduling).
3. Eseguire le operazioni, accedendo al controller.
4. Passare il processo in modo wait (I/O interrupt-driven), o attendere la fine dell'operazione in busy-wait.
5. Controllare lo stato dell'operazione nel controller.
6. Restituire il risultato.

I driver devono essere rientranti, ovvero, nel mezzo di un'esecuzione, può essere lanciata una nuova esecuzione.

5. Si spieghi cos'è un file e quali sono le principali operazioni su di esso che un S.O. deve implementare.

Risposta: (4 punti) Un file è insieme di informazioni correlate a cui è stato assegnato un nome. Esso è inoltre la più piccola porzione unitaria di memoria logica secondaria allocabile dall'utente o dai processi di sistema. Le operazioni fondamentali sui file che un sistema operativo deve implementare sono le seguenti:

Creazione: allocazione dello spazio sul dispositivo e collegamento di tale spazio al file system.

Cancellazione: staccare il file dal file system e deallocare lo spazio assegnato al file.

Apertura: caricare alcuni metadati dal disco nella memoria principale, per velocizzare le chiamate seguenti.

Chiusura: deallocare le strutture allocate nell'apertura.

Lettura: dato un file e un puntatore di posizione, i dati da leggere vengono trasferiti dal media in un buffer in memoria.

Scrittura: dato un file e un puntatore di posizione, i dati da scrivere vengono trasferiti sul media.

Append: versione particolare di scrittura.

Riposizionamento (seek): non comporta operazioni di I/O.

Troncamento: azzerare la lunghezza di un file, mantenendo tutti gli altri attributi.

Lettura dei metadati: leggere le informazioni come nome, timestamp, ecc.

Scrittura dei metadati: modificare informazioni come nome, timestamps, protezione, ecc.

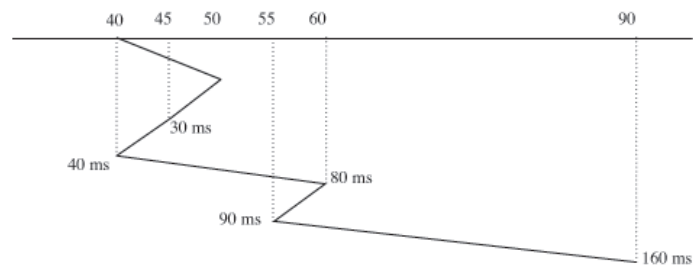
Sistemi Operativi
7 luglio 2011
Compitino II A

6. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 40; lo spostamento ad una traccia adiacente richiede 2 ms. Al driver di tale disco arrivano richieste per i cilindri 90, 45, 40, 60, 55, rispettivamente agli istanti 0 ms, 20 ms, 30 ms, 40 ms, 80 ms. Si trascuri il tempo di latenza.

1. In quale ordine vengono servite le richieste?
2. Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le cinque richieste in oggetto?

Risposta:

1. (3 punti) Le richieste vengono servite nell'ordine 45, 40, 60, 55, 90:



2. (2 punti) Il tempo di attesa medio per le cinque richieste in oggetto è
$$\frac{(160-0)+(30-20)+(40-30)+(80-40)+(90-80)}{5} = \frac{160+10+10+40+10}{5} = \frac{230}{5} = 46 \text{ ms.}$$