SISTEMI OPERATIVI
21 settembre 2011

Cognome:_____ **Nome:**_____
Matricola:_____

1. Ricordate che non potete usare calcolatrici o materiale didattico.
2. Ricordate che potete consegnare al massimo tre prove scritte per anno accademico.

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

ESERCIZIO 1 (5 punti)

a) Si consideri il problema dei lettori e scrittori visto a lezione, dove i codici del generico scrittore e del generico lettore sono riportati qui di seguito.

Inserite le operazioni di wait e signal mancanti necessarie per il funzionamento del sistema secondo la soluzione vista a lezione, indicando anche il semaforo mancante ed il suo valore di inizializzazione.

semafori e variabili condivise necessarie con relativo valore di inizializzazione:
semaphore mutex = 1;
semaphore scrivi = 1;
int numlettori = 0;

"scrittore"
{
wait(scrivi);
Esegui la scrittura del file
signal(scrivi)
}

"lettore"
{
wait(mutex);

numlettori++;

if numlettori == 1 **wait(scrivi);**

signal(mutex);

... leggi il file ...

wait(mutex);

numlettori--;

if numlettori == 0 **signal(scrivi);**

signal(mutex);

b) La soluzione del problema dei lettori e scrittori vista a lezione garantisce l'assenza di starvation? (motivate la vostra risposta)

No, infatti un qualsiasi processo scrittore potrebbe dover attendere all'infinito senza riuscire a ad entrare in sezione critica. Al contrario i processi lettori sono liberi da starvation (in altre parole, non è garantita l'attesa limitata

c) Quali sono le tre condizioni fondamentali che deve rispettare una soluzione corretta al problema della sezione critica, e quali problemi possono portare se non vengono rispettate?

Mutua esclusione (se non rispettata può portare ad avere due o più processi contemporaneamente in sezione critica), attesa limitata (se non rispettata produce starvation), progresso (se non rispettato produce deadlock).

d) Descrivete brevemente un pregio ed un difetto dei semafori, in quanto strumenti di sincronizzazione, implementati come visto a lezione.

Pregio: evitano il busy waiting
Difetto: non sono primitive di sincronizzazione strutturate, per cui un loro uso scorretto può portare alla violazione della mutua esclusione, a starvation o deadlock.

e) Riportate lo pseudocodice che descrive l'implementazione dell'operazione di Wait, e dite che informazione fornisce il valore corrente del semaforo

Per lo pseudocodice si vedano i lucidi della sezione 6.5.2

Se il S->valore < di zero il suo valore assoluto ci dice quanti processi sono sospesi su quel semaforo. In caso contrario il valore ci dice quanti processi possono eseguire la wait su quel semaforo senza venire sospesi.

ESERCIZIO 2 (5 punti)

Un sistema con memoria paginata usa un TLB con un hit-ratio del 95%, e un tempo di accesso di 20 nanosecondi. Un accesso in RAM richiede invece 0,08 microsecondi.

a) Qual è, in nanosecondi, il tempo medio di accesso in RAM (esplicitate i calcoli che fate)?

Tmedio = 0,95 * (80+20) + 0,05 * (2*80 + 20) = 95 + 9 = 104 nanosecondi

b) Il sistema viene ora dotato di memoria virtuale, usando come algoritmo di rimpiazzamento quello della *seconda chance*.

E' possibile che un processo del sistema si veda aumentare il numero di frame che usa e contemporaneamente aumenta anche il numero di page fault generati dal processo?

Si, perché nel caso peggiore l'algoritmo si comporta come FIFO, che soffre dell'anomalia di Belady.

c) Quali informazioni conterrà ciascuna entry di una page table di un sistema che usa l'algoritmo della seconda chance?

Numero di un frame, bit di validità, bit di riferimento.

d) E' corretto dire che in un sistema che implementa la memoria virtuale, i processi partono più velocemente (in media)?

Si, perché un processo può partire anche se non tutto il suo codice e i dati sono stati caricati in RAM.

e) Descrivete brevemente come avviene la prevenzione del thrashing nel sistema Windows.

Windows fissa una soglia minima di frame liberi che devono sempre essere presenti nel sistema. Ad ogni processo sono associati un insieme di lavoro minimo e massimo. Se nel sistema il numero di frame liberi scende sotto la soglia minima, ad ogni processo vengono rimosse tutte le pagine in eccesso rispetto al suo insieme di lavoro minimo (le pagine vengono scelte con l'algoritmo della seconda chance, o con FIFO).

ESERCIZIO 3 (4 punti)

Un hard disk ha la capienza di 2^{38} byte, ed è formattato in blocchi da 1024 byte.

a) Quanti accessi al disco sono necessari per leggere l'ultimo blocco di un file A della dimensione di 4096 byte, assumendo che sia già in RAM il numero del primo blocco del file stesso e che venga adottata una allocazione concatenata dello spazio su disco? (motivate la vostra risposta)

5. Ogni blocco infatti memorizza 1020 byte di dati più 4 byte di puntatore al blocco successivo (infatti, $2^{38}/2^{10} = 2^{28}$), per cui sono necessari 5 blocchi per memorizzare l'intero file.

b) Qual è lo spreco di memoria dovuto alla frammentazione interna nella memorizzazione di A (motivate la risposta)?

L'hard disk è suddiviso in $2^{38}/2^{10} = 2^{28}$ blocchi, sono necessari 4 byte per memorizzare un puntatore al blocco successivo, e ogni blocco contiene 1020 byte di dati. Il quinto blocco memorizzerà quindi 16 byte del file, e la frammentazione interna corrisponde a 1024 – 16 = 1008 byte (1004 se si considerano non sprecati i 4 byte del quinto blocco che contengono il puntatore, non utilizzato, al blocco successivo)

c) Se si adottasse una allocazione indicizzata dello spazio su disco, quanti accessi al disco sarebbero necessari per leggere l'ultimo byte di un file B grande 400k byte (specificate quali assunzioni fate nel rispondere a questa domanda e motivate la vostra risposta)?

Poiché sono necessari 4 byte per scrivere il numero di un blocco, in un blocco indice possono essere memorizzati 256 puntatori a blocco, e con un blocco indice possiamo indirizzare in tutto $2^8 * 2^{10} = 2^{18} =$ 256K byte. Un solo blocco indice non è quindi sufficiente a memorizzare B. Assumendo una allocazione indicizzata concatenata (ma si ottengono gli stessi risultati con una allocazione indicizzata gerarchica) usando un secondo blocco indice è possibile memorizzare l'intero file. Se il numero del primo blocco indice è già in RAM, per leggere l'ultimo carattere del file sono necessari 3 accessi al disco: lettura del primo blocco indice, lettura del secondo blocco indice, lettura dell'ultimo blocco del file.

d) in quale caso l'allocazione indicizzata comporta un grande spreco di spazio su disco?

Nel caso di file molto piccoli, perché il blocco indice rimane quasi completamente inutilizzato.