

SISTEMI OPERATIVI E LABORATORIO
(Indirizzo Sistemi e Reti)
16 settembre 2009

Cognome: _____ Nome: _____
Matricola: _____

ESERCIZIO 1 (5 punti)

Tre processi P_A , P_B e P_C eseguono il seguente codice:

Shared **Var** semaphore mutex = 1; (valore iniziale)
 semaphore done = 0; (valore iniziale)

P_A :
repeat forever:
wait(done)
wait(done)
wait(mutex)
<A>
signal(mutex)

P_B :
repeat forever:
wait(done)
wait(mutex)

signal(mutex)
signal(done)

P_C :
repeat forever:
wait(mutex)
<C>
signal(mutex)
signal(done)

a)

L'esecuzione concorrente di P_A , P_B e P_C produce una sequenza (di lunghezza indefinita) di chiamate alle procedure A, B e C. Quale o quali delle sequenze qui sotto riportate può essere la porzione iniziale di sequenze prodotte dall'esecuzione concorrente di P_A , P_B e P_C ? (marcate la o le sequenze che scegliete con una croce nello spazio apposito)

1. ☐ C,B,B,A,C,C,A,B,C...
2. ☒ C,C,B,A,C,C,A,C,C ...
3. ☐ C,C,A,C,A,C,B,B,C ...
4. ☒ C,C,A,C,C,B,A,C,B ...

b)

Durante l'esecuzione dei tre processi P_A , P_B e P_C , è possibile dire qual è *il valore più piccolo* che può assumere la variabile semaforica di "done"? E' possibile dire qual è *il valore massimo* che può assumere la variabile semaforica di "done"? (motivare la vostra risposta, assumendo che P_A , P_B e P_C siano i soli processi ad usare "done")

Il valore più piccolo è -2: si verifica se P_A e P_B eseguono la wait sul semaforo uno di seguito all'altro.

Il valore massimo è indefinito (in realtà, in linea di principio può essere pari al più grande valore intero rappresentabile nel sistema su cui girano i processi): non si può infatti stabilire a priori quante volte P_C può ciclare prima che gli altri due processi riescano ad entrare in CPU.

c)

Descrivete brevemente i problemi che si possono presentare se non viene correttamente risolto il problema della sezione critica

Lucidi 6.2 nn. 15-17

d)

Che cosa significa che un sistema operativo ha un *kernel con diritto di prelazione*?

Lucidi 6.2 nn. 21-23

e)

Un sistema operativo usa un algoritmo di scheduling Round Robin e lascia che i processi utente si sincronizzino fra loro usando un meccanismo di sincronizzazione via hardware (ad esempio implementato attraverso l'istruzione macchina Test&Set) basato su busy waiting. Cosa succede quando un processo cerca di entrare in una sezione critica già occupata?

L'intero quanto di tempo assegnato a quel processo viene sprecato. (lucido 38 sez. 6.4)

ESERCIZIO 2 (5 punti)

In un SO la tabella delle pagine può contenere al massimo 256 (decimale) entry, e l'offset massimo all'interno di una pagina è FFF (esadecimale).

a) Il SO potrebbe dover adottare un sistema di paginazione a due livelli (motivate la vostra risposta)?

No. Una tabella di 256 entry che usi un intero frame da 4096 byte permetterebbe di assegnare a ciascuna entry $4096/256 = 16$ byte = 128 bit in cui scrivere il numero di un frame. In questo caso lo spazio di indirizzamento fisico sarebbe di 2^{128} frame da 4 Kbyte, ben superiore a quello delle attuali macchine reali. Sicuramente quindi una tabella delle pagine di questo sistema occupa meno di un frame.

b) Quale dimensione può avere lo spazio di indirizzamento fisico di questo sistema per poter dire con certezza che il sistema deve implementare la memoria virtuale?

Lo spazio di indirizzamento fisico deve essere inferiore a quello logico. Poiché lo spazio di indirizzamento logico è di $2^8 * 2^{12} = 2^{20}$ byte, lo spazio di indirizzamento fisico dovrà essere al massimo grande 2^{19} byte (in realtà anche con uno spazio fisico di 2^{20} byte sarebbe necessario implementare la memoria virtuale, in quanto una parte della RAM sarebbe comunque occupata dal SO)

c) Descrivete brevemente i vantaggi e gli svantaggi fondamentali della paginazione della memoria:

Vantaggi: si sfrutta al meglio la RAM, evitando la frammentazione esterna e riducendo a mezza pagina (in media) quella interna. Si realizza una forma automatica di protezione dello spazio di indirizzamento di ciascun processo.

Svantaggi. Il meccanismo di traduzione degli indirizzi è complesso e introduce dei ritardi. Per limitarli è necessario un supporto hardware (TLB). Inoltre le tabelle delle pagine occupano spazio in RAM

d1) Un sistema che implementa la memoria virtuale e che adotta una politica di rimpiazzamento delle pagine locale può soffrire di trashing? (motivate la risposta)

d2) Un sistema che implementa la memoria virtuale e che ha uno spazio di indirizzamento fisico maggiore di quello logico può soffrire di trashing? (motivate la risposta)

d1) Sì. Se ci sono troppo processi nel sistema, ciascuno avrà pochi frame a disposizione e produrrà continuamente dei page fault, sostituendo una propria pagina che sarà molto probabilmente riferita nell'immediato futuro.

d2) Sì. Se un sistema implementa la memoria virtuale può soffrire di trashing indipendentemente dalla dimensione degli spazi di indirizzamento logici e fisici.

e)

Perché l'uso delle librerie dinamiche è preferibile all'uso delle librerie statiche?

Perché permettono di risparmiare spazio in RAM. Infatti, una sola copia della libreria può essere condivisa da tutti i programmi che la usano, e la libreria viene caricata in RAM solo se il programma che la usa chiama una delle subroutine della libreria stessa.

ESERCIZIO 3 (4 punti)

Un hard disk ha la capienza di 2^{33} byte, ed è formattato in blocchi da 512 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)

9. Ogni blocco infatti memorizza 509 byte di dati più 3 byte di puntatore al blocco successivo (infatti, $2^{33}/2^9 = 2^{24}$), per cui sono necessari 9 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 $2^{33}/2^9 = 2^{24}$ blocchi, sono necessari tre byte per memorizzare un puntatore al blocco successivo, e ogni blocco contiene 509 byte di dati. Il nono blocco memorizzerà quindi 24 byte del file, e la frammentazione interna corrisponde a $512 - 24 = 488$ byte (485 se si considerano non sprecati i tre byte del quinto blocco che contengono il puntatore, non utilizzato, al blocco successivo)

c) Quante operazioni di lettura/scrittura sono necessarie per cancellare il terzo blocco del file, sempre assumendo che il numero del primo blocco sia già in RAM (motivate la risposta)?

4. Occorre leggere il primo, il secondo, e il terzo blocco del file (leggere anche il terzo blocco serve per sapere il numero del quarto blocco). Occorre poi modificare il valore del puntatore della copia in RAM del secondo blocco (che ora deve puntare a quello che era il quarto blocco, ora divenuto il terzo) e riscriverlo sull'hard disk. Se si conta anche l'operazione di aggiornamento sull'hard disk degli attributi del file le operazioni diventano 5. Se si conta anche l'aggiornamento sull'hard disk della struttura che tiene traccia dei file liberi, le operazioni diventano almeno 6.

Parte di Unix