SISTEMI OPERATIVI
9 febbraio 2011

Cognome:_____Nome:_____
Matricola:_____

1. Ricordate che non potete usare calcolatrici o materiale didattico.
2. Ricordate che se prendete un voto inferiore a 10 dovete saltare lo scritto del 24 febbraio 2011.

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 quale di queste tre, se non viene rispettata, produce starvation?

Mutua esclusione, attesa limitata, progresso. La mancanza di attesa limitata produce starvation.

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 il diagramma di stato della vita di un processo, e indicate quale stato, secondo voi, può coinvolgere l'uso di semafori.

Per il diagramma di stato si veda il lucido della sezione 3.1.2
Lo stato di Waiting

ESERCIZIO 2 (5 punti)

In un sistema la memoria fisica è divisa in 2^{22} frame, un indirizzo logico è scritto su 32 bit, e all'interno di una pagina, l'offset massimo è 3FF.

a) Quanti frame occupa la la page table più grande del sistema? (motivate numericamente la vostra risposta)

Un frame/pagina è grande 2^{10} = 1024 byte, e quindi la page table più grande può avere $2^{(32-10)} = 2^{22}$ entry. Nel sistema vi sono 2^{22} frame, per cui sono necessari tre byte per scrivere il numero di un frame, e quindi la page table più grande occupa $(2^{22} \cdot 3) / 2^{10}$ frame = $2^{12} \cdot 3$ frame = 12K frame

b) Quale dimensione minima dovrebbero avere le pagine di questo sistema per essere certi di non dover ricorrere ad una paginazione a più livelli? (motivate la vostra risposta, e per questa domanda assumete di usare 4 byte per scrivere il numero di un frame all'interno di una page table)

Poniamo 32 = m + n (m = bit usati per scrivere un numero di pagina, n = bit usati per scrivere l'offset). Allora il numero di entry della PT più grande, moltiplicato per la dimensione di una entry deve poter essere contenuto in una pagina/frame, ossia: $2^m \cdot 2^2 \leq 2^n$
Da cui: $m + 2 \leq n$. Poiché $m = 32 - n$; risolvendo il semplice sistema si ha $n = 17$, ossia le pagine devono almeno essere grandi 2^{17} = 128 Kbyte.

c) Questo sistema può soffrire del problema del thrashing?

Non si può dire a priori, poiché dipende dal fatto che sia implementata o meno la memoria virtuale.

ESERCIZIO 3 (4 punti)

a) Dove sono memorizzati, da un punto di vista logico (ovviamente, fisicamente gli attributi sono in memorizzati in modo permanente in qualche blocco del disco), gli attributi di un file?

Nella directory che "contiene" il file, oppure in una struttura interna del sistema puntata da un puntatore contenuto della directory che "contiene" il file.

b) Considerate la seguente sequenza di comandi Unix (assumete che tutti i comandi lanciati possano essere correttamente eseguiti):

```
1:   cd /tmp
2:   mkdir mynewdir
3:   cd mynewdir
4:   echo "ciao" > pippo          // crea un nuovo file di nome pippo contenente la stringa ciao
5:   ln pippo pluto
6:   ln –s pippo paperino
7:   ln pluto topolino
8:   rm pippo
9:   cat topolino                 // cat stampa il contenuto del file passato come argomento
10:  cat paperino
```

Dopo l'esecuzione di tutti i comandi:
qual è il valore del link counter nell'index-node associato al link fisico *topolino*? 2
qual è il valore del link counter nell'index-node associato al link fisico *mynewdir*? 2
cosa possiamo dire del link counter dell'index-node associato al link fisico *tmp*? Che è aumentato di 1, a causa dell'entry ".." inserita dentro la nuova sottocartella mynewdir.
Qual è l'output del comando numero 10? "no such file or directory"

c) Descrivete brevemente l'allocazione indicizzata dello spazio su disco, e la variante degli index-node (se preferite, usate degli opportuni disegni)

Vedere i lucidi della sezione 11.4.3

d) In quale caso la lettura dei dati di un file è più veloce se il file è memorizzato su un sistema raid anziche su un normale hard disk?

Quando i blocchi del file appartengono a strip memorizzati su dischi diversi del sistema raid, e possono quindi essere letti in parallelo.