

SISTEMI OPERATIVI E LABORATORIO
(Esonero e Scritto - Indirizzo Sistemi e Reti)
17 aprile 2009

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

Ricordate che non potete usare calcolatrici o materiale didattico. Siate sintetici nelle vostre risposte, anche quando è richiesto di motivarle, sono sufficienti poche righe per rispondere correttamente. (Si ricorda che gli studenti degli anni precedenti devono sostenere l'intero scritto).

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

(il punteggio conseguito farà media con quello ottenuto nella parte di laboratorio. E' comunque necessario prendere almeno 18 punti per considerare passata la parte di teoria.)

ESERCIZIO 1 (9 punti)

Un sistema adotta una Inverted Page Table con 1024 entry (tutte usate). Un indirizzo logico è scritto su 3 byte (usando tutti i bit) e lo spazio di indirizzamento logico di un processo può usare al massimo 1000 (esadecimale) pagine. (nel seguito motivate tutte le risposte che date)

(a) Su quanti bit è scritto un indirizzo fisico del sistema (esplicitate i calcoli che fate)?

Poiché la IPT tiene traccia dello stato di allocazione dei frame della RAM, e ha 1024 entry, vi sono 2^{10} frame nel sistema. Lo spazio di indirizzamento logico è formato da 1000 pagine, per cui il numero di una pagina va da 000 ad FFF, ed essendo usati 24 bit per scrivere un indirizzo logico, 12 bit sono usati per l'offset all'interno di una pagina e di un frame. Di conseguenza un indirizzo fisico è scritto $10+12 = 22$ bit.

(b) Possiamo dire se questo sistema adotta un algoritmo di rimpiazzamento delle pagine?

Sì, ne fa uso. Infatti lo spazio di indirizzamento logico è più grande di quello fisico, e il sistema deve implementare anche la memoria virtuale.

(c) Perché si usa una Inverted Page Table al posto di normali page table?

Per limitare la quantità di memoria primaria occupata dalle page table dei processi.

(d) L'uso di una Inverted Page Table richiede anche l'uso di un Translation Look-aside Buffer?

Sì, poiché altrimenti la traduzione di un indirizzo logico in uno fisico richiede in media di leggere la metà delle entry della IPT

(e) Come è fatta una generica i -esima entry di una Inverted Page Table, e qual è il significato della sua posizione nella IPT (ossia che cosa indica il valore "i"?)

Ogni entry è una coppia di valori: “Process ID – Numero della pagina dell’indirizzo logico da tradurre”. Se quella coppia si trova all’*i*-esima entry vuol dire che la pagina specificata nella entry si trova nel frame numero *i*.

- (f) Che tipo di codice genera il compilatore di un moderno sistema operativo che implementa la memoria virtuale, e perché?

Codice dinamicamente rilocabile, in modo che il codice possa essere spostato, se necessario, da un punto all’altro della RAM senza dover ricalcolare gli indirizzi usati dalle istruzioni del codice stesso.

ESERCIZIO 2 (9 punti)

- a) riportate lo pseudocodice che descrive correttamente l’implementazione delle operazioni di *wait* e di *signal* su un semaforo.

Wait(semaphore S) :

```
S.value--;  
if S.value < 0 then  
    aggiungi questo processo a S.waiting  
    block();  
end;
```

signal(semaphore S):

```
S.valore++;  
if S.valore <= 0  
    toglì un processo P da S.waiting;  
    wakeup(P);  
end;
```

- b) perché nell’implementazione di una sezione critica l’uso dei semafori permette di evitare lo spreco di tempo di cpu dovuto al busy waiting?

Perché il processo che si addormenta su un semaforo non viene messo in coda di ready, ma sulla coda di attesa del semaforo, e non può quindi essere selezionato dallo scheduler. Il processo tornerà in coda di ready solo quando il potrà accedere alla sezione critica (perché un altro processo, uscendo dalla sezione critica lo avrà svegliato attraverso una signal).

- c) Di un sistema si sa quanto segue:

- Il sistema usa un algoritmo di scheduling Round Robin con quanto di tempo pari a *Q*.

- Nel sistema sono presenti in tutto N processi utente, i quali condividono una sezione critica (ossia, devono sincronizzarsi per poter accedere in modo mutuamente esclusivo alla sezione critica).
- I processi usano un meccanismo di sincronizzazione basato su busy waiting (implementato ad esempio attraverso la test&Set o la Swap).
- Un processo P è appena entrato in sezione critica, e deve rimanerci per una quantità di tempo T pari a: $Q < T < 2Q$

Qual è la quantità massima di tempo di CPU sprecata a causa dell'implementazione di un meccanismo di sincronizzazione basato sul busy waiting, dal momento in cui P entra in sezione critica al momento in cui ne esce (motivate la vostra risposta, specificando eventuali assunzioni che fate)?

$Q(N-1)$. Infatti dopo essere entrato in sezione critica il processo P subisce il context switch prima di esserne uscito. Se tutti gli altri processi tentano di entrare in sezione critica la trovano occupata e passano il quanto di tempo loro assegnato a fare busy waiting (si veda il lucido 38 del capitolo 6)

ESERCIZIO 3 (9 punti)

In un sistema si sa che la quasi totalità dei file (diciamo il 99,99% dei file) hanno una dimensione molto superiore a quella di un blocco dell'hard disk, e devono poter essere acceduti in modo diretto

a) Quali metodi di allocazione dello spazio su disco non verranno sicuramente adottati per quel particolare sistema, e perché?

L'allocazione contigua, per la difficoltà di trovare spazio contiguo sufficiente, e l'allocazione concatenata, a causa del tempo di accesso proporzionale alla lunghezza del file.

b) Si supponga che il sistema usi una allocazione dello spazio su disco simile a quella dei sistemi unix, usando blocchi da 1024 byte e 32 bit per scrivere il numero di un blocco. Quanti accessi su disco sono necessari per leggere il byte numero 200.000 di un file, supponendo già presente in memoria principale il numero dell'i-node associato al file? (motivate la vostra risposta)

3. Innanzitutto, un accesso in lettura è necessario per leggere l'i-node del file. Poiché il file è lungo almeno 200.000 byte, i primi 10 puntatori a blocco dell'i-node non sono sufficienti ad indirizzare tutti i blocchi del file, ed è necessario usare il puntatore di indirezione semplice. Infatti un blocco può contenere 256 puntatori a blocco, tra i quali vi sarà anche il blocco che contiene il byte numero 200.000. Altre due letture sono quindi necessarie per leggere prima il blocco di indirezione semplice e poi il blocco contenente il byte 200.000.

c) spiegate la differenza tra un pathname relativo e il pathname assoluto di un file

Un pathname relativo è sempre relativo ad una directory diversa dalla radice del file system, e non inizia mai con il simbolo "/", mentre il pathame assoluto di un file inizia sempre dalla radice del file system.

d) commentate questa la seguente asserzione: *se viene perso il pathname assoluto di un file non è più possibile accedere ai dati del file*

L'asserzione è senza senso, in quanto normalmente il pathname di un file non è memorizzato da nessuna parte

e) Dite quale configurazione RAID è opportuno scegliere per le seguenti situazioni:

1. Il sistema deve essere estremamente veloce nell'accesso ai dati e sfruttare al massimo lo spazio di memorizzazione disponibile, mentre l'affidabilità non è particolarmente importante: RAID di livello 0
2. Il sistema deve garantire una ragionevole affidabilità ed efficienza nell'accesso ai dati, sfruttando al meglio lo spazio disponibile: RAID di livello 5
3. Il sistema deve garantire la massima affidabilità: RAID di livello 1