

SISTEMI OPERATIVI E LABORATORIO
(Esonero e Scritto - Indirizzo Sistemi e Reti)
16 aprile 2008

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)

Si consideri un sistema in cui la tabella delle pagine di un processo può avere al massimo 1024 entry. Un indirizzo fisico generato dal sistema è scritto su 20 bit, e la RAM è suddivisa in 512 frame.

(a) Quanto è grande lo spazio di indirizzamento logico del sistema (esplicitate i calcoli che fate)?

Un numero di frame è scritto su 9 bit, per cui la dimensione di un frame, e quindi di una pagina, è di 2^{11} byte. Poiché un processo può essere formato al massimo da $1024 = 2^{10}$ pagine, lo spazio di indirizzamento logico è di $2^{10} \times 2^{11}$ byte (pari a circa 2 megabyte).

(b) Per ciascuna entry di una tabella delle pagine di questo sistema, è necessario memorizzare anche il bit di validità della pagina corrispondente? (motivate la vostra risposta)

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

(c) Per ciascuna entry di una tabella delle pagine di questo sistema, è necessario memorizzare anche il dirty bit della pagina corrispondente? (motivate la vostra risposta)

Si, nel caso in cui venga utilizzato un algoritmo di rimpiazzamento delle pagine che ne fa uso, come ad esempio l'algoritmo della seconda chance migliorato.

(d) Quale soluzione viene comunemente usata per limitare l'inefficienza dell'accesso in RAM tipica dei sistemi che adottano la paginazione della memoria?

Viene usato un TLB, una memoria associativa che limita il tempo necessario alla traduzione di un indirizzo logico in fisico.

(e) E' possibile che un certo programma P abbia un turnaround migliore in un sistema che implementa la memoria virtuale piuttosto che in un sistema che non implementa la memoria

virtuale (ovviamente a parità di tutte le altre condizioni). Motivate la vostra risposta con un esempio esemplificativo.

Il tempo di risposta di un programma P è normalmente migliore in un sistema che implementa la memoria virtuale, dato che deve essere caricato meno codice dall'hard disk prima che P possa iniziare ad eseguire. se P non genera mai page fault, avrà quindi un turnaround migliore in un sistema che implementa la memoria virtuale.

(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 dell'operazione di *wait* su un semaforo.

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

b) Che effetto ha la system call *block()* usata nello pseudocodice?

Toglie la CPU al processo che la invoca e manda in esecuzione uno dei processi in Ready Queue. Il processo che ha chiamato *block* non viene rimesso nella Ready Queue

c) Per le seguenti affermazioni, dite se sono vere e false, motivando la vostra risposta.

- I. L'uso dei semafori evita il problema della starvation e/o del deadlock
falso, perché questo dipende da come vengono usati
- II. L'uso dei semafori evita il problema del busy waiting
Vero perché permette di eliminare l'attesa attiva nei processi che li usano

d) Riportate lo pseudocodice di due processi A e B che devono avere il seguente comportamento: il processo A esegue, nell'ordine, le procedure <A1> e <A2>. Il processo B esegue la procedura <B1>.

La procedura <B1> deve essere eseguita **dopo** che è stata eseguita la procedura <A1> e **prima** che venga eseguita la procedura <A2>. Indicate il valore di inizializzazione dei semafori che usate per ottenere l'effetto richiesto.

Semaphore sync1 = 0; sync2 = 0;

A:

...
<A1>
signal(sync1)
wait (sync2)
<A2>
...

B:

...
wait(sync1)
<B1>
signal(sync2)
...
...

ESERCIZIO 3 (9 punti)

In un sistema si sa che la quasi totalità dei file (diciamo il 99,99% dei file) hanno una dimensione inferiore a quella di un blocco dell'hard disk.

a) Quale (o quali) metodo di allocazione dello spazio su disco è più conveniente adottare per quel particolare sistema, e perché?

Sicuramente una allocazione contigua o concatenata (che nel caso di file lunghi un blocco praticamente coincidono), poiché gli svantaggi di queste modalità di allocazione non si manifestano per file che stanno in un solo blocco (difficoltà di trovare spazio contiguo sufficiente e tempo di accesso proporzionale alla lunghezza del file).

b) Quale (o quali) metodo di allocazione dello spazio su disco è meno conveniente adottare per quel particolare sistema, e perché?

Sicuramente l'allocazione indicizzata, perché per ogni file lungo un blocco è necessario usare un blocco indice quasi completamente inutilizzato

c) Usando opportuni disegni esemplificativi, illustrate (anche solo graficamente) la modalità di allocazione dello spazio su disco adottata da:

- I. allocazione indicizzata
- II. allocazione indicizzata concatenata
- III. allocazione indicizzata a più livelli
- IV. allocazione indicizzata nella variante adottata dagli i-node Unix

Si vedano i lucidi della sezione 11.4

d) Commentate la seguente considerazione: ***Quando un processo deve leggere il contenuto di un file (ad esempio usando delle fscanf), il tempo necessario ad accedere al file dipende dalla lunghezza del pathname assoluto del file.***

Il primo accesso al file richiede l'uso di una system call open. Se la open fa uso del pathname assoluto del file, allora l'intero pathname deve essere percorso per raggiungere il file, con diversi accessi all'hard disk. Successivi accessi al file tuttavia, useranno le informazioni contenute nella open file table.

e) Un sistema RAID è composto da 10 hard disk da 100 Megabyte ciascuno. Dite qual è l'effettiva capacità di memorizzazione del sistema a seconda che venga adottata la seguente configurazione:

RAID livello 0: 1 gigabyte

RAID livello 1: 500 megabyte

RAID livello 4: 900 megabyte

RAID livello 5: 900 megabyte