

# Sistemi Operativi

## Compitino Basic 18 Marzo 2005

Si risponda ai seguenti quesiti in modo sintetico (evidenziando quindi tutti gli aspetti necessari per rispondere completamente alla domanda evitando di fornire dettagli insignificanti o non richiesti)

1. Si consideri la seguente tabella dei segmenti: Si calcolino gli indirizzi fisici corrispondenti ai seguenti indirizzi logici: A.  $\langle 0, 0x230 \rangle$ ; B.  $\langle 1, 0x80 \rangle$ ; C.  $\langle 2, 0x220 \rangle$ ; D.  $\langle 3, 0x430 \rangle$ ; E.  $\langle 5, 0x20 \rangle$ .

Segm.	Base	Limite
0	0xAAE01100	0x200
1	0x00012330	0x100
2	0x07000080	0x320
3	0x01001234	0x800

- A. errore (fuori segmento)  
 B. 0x000123B0 C. 0x070002A0  
 D. 0x01001664 E. errore (segmento inesistente)

2. Un computer ha quattro frame. La seguente tabella riporta per ogni frame l'istante di caricamento, di ultimo riferimento e il reference bit. Dite quale pagina verrebbe liberata dall'algoritmo FIFO; da LRU e da CLOCK (il puntatore all'ultimo frame controllato sia sul 2).

Frame	Car.	Rif.	R
0	26	279	1
1	130	280	1
2	110	272	0
3	170	280	0

FIFO: 0; LRU: 2 e CLOCK: 2

3. I sistemi Unix usano (gran) parte della memoria principale come buffer e cache per le operazioni di I/O. In un sistema a memoria virtuale paginata:

- (a) Quale dimensione è consigliabile usare per i buffer di I/O da programma?

la stessa del frame fisico, in quanto i buffer vengono allocati prendendo frame dalla free list.

- (b) Da quale struttura del Sistema Operativo viene preso lo spazio per allocare i buffer?

dalla free list, appunto.

4. (a) Si descrivano i passi eseguiti da un driver per attivare una operazione di I/O sincro con un dispositivo basato su interrupt.

I passi principali eseguiti dal driver sono i seguenti:

1. Verificare la validità dei parametri passatigli dagli strati superiori del kernel;
2. Accodare la richiesta alla coda delle richieste (se implementata anche lo scheduling delle operazioni)
3. Accedere al controller per attivare le operazioni (agendo sui registri come necessario).
4. Passare il processo corrente nello stato wait, e richiamare lo scheduler per il context switch.
5. Verificare il risultato delle operazioni, accedendo al controller (eventualmente, leggendo i dati ivi contenuti).
6. Restituire il risultato allo strato superiore.

- (b) Quale parte del sottosistema di I/O comunica al processo l'avvenuta terminazione dell'operazione, e in seguito a quale evento?

La routine di gestione degli interrupt, invocata dal controller stesso sollevando l'interrupt associatogli.

5. Descrivete sinteticamente le tecniche di comunicazione DMA e DVMA. Quali vantaggi e svantaggi comportano? Si dia un esempio di DVMA nei PC dei giorni nostri.

DMA è l'accesso diretto alla memoria fisica da parte delle periferiche. Alleggerisce la CPU dalla gestione diretta dell'I/O, ma occupa il bus in modo asincrono e solleva il problema del *buffer interlock*, ossia la necessità di escludere dagli algorit-

# Sistemi Operativi

## Compitino Basic 18 Marzo 2005

mi di paginazione i frame coinvolti nella comunicazione DMA.

DVMA è l'accesso diretto alla memoria virtuale da parte delle periferiche. Come il DMA, alleggerisce la CPU dalla gestione diretta dell'I/O, e i frame coinvolti nell'I/O diretto sono gestiti secondo le normali strategie della memoria virtuale. La complicazione è che bisogna dotare questi canali DVMA di hardware di rilocalizzazione dinamica paragonabili ad una MMU, con il conseguente costo di gestione da parte del kernel (sia per manipolarle, sia per gestire i page fault che genererebbero). Un esempio di architettura DVMA è AGP, dove la MMU dedicata si chiama GART (Graphic Address Relocation Table).

6. Dite in quale strato del sottosistema di I/O di Unix, viene svolta ognuna delle seguenti azioni:

- (a) Convertire i codici che giungono dalla tastiera nei corrispondenti caratteri ASCII.

codice kernel indipendente dai device

- (b) Attivare il trasferimento via DMA dal controller del disco alla memoria.

driver specifici di ogni dispositivo

- (c) Sostituire i %d, %s, ..., della stringa di formattazione di `printf` con i corrispondenti valori.

programmi e librerie in user space

- (d) Alla terminazione dell'operazione di I/O, portare dallo stato *wait* allo stato *ready* il processo in attesa dei dati.

routine di gestione degli interrupt

- (e) Controllare se l'utente ha i permessi necessari per accedere al dispositivo.

File System

7. Come è implementato il sistema delle directory in HFS? Quali vantaggi pone questo sistema riguardo alla ricerca di files all'interno dell'intero File System?

È implementato con (una variante di) B-tree. Il vantaggio consiste nell'aver tempo di ricerca di un qualsiasi file logaritmico nel numero totale di files presenti nel File System.

8. Quale strategia di allocazione dei file è consigliabile per supporti di sola lettura, come CD-ROM e i DVD? Discutetene i vantaggi e svantaggi in tale situazione.

In tali situazioni, l'allocazione contigua è sicuramente da preferire. Importanti vantaggi: flusso ad (alta) velocità costante dei dati in lettura sequenziale (adatto a dati multimediali), semplicità nel calcolo dell'accesso diretto. Gli svantaggi tipici dell'allocazione contigua (difficoltà di allocazione dinamica e riallocazione degli spazi, frammentazione interna ed esterna) qui non sussistono, perché il file system non viene modificato una volta creato.

9. (a) Si descriva sinteticamente il concetto di *matrice di accesso*.

La matrice di accesso è un modello generale di protezione. Le righe rappresentano i domini, le colonne gli oggetti; ogni casella contiene i diritti che quel dato dominio esercita sul dato oggetto.

- (b) Qual è il suo svantaggio principale?

La dimensione: generalmente tale matrice è così grande che non può essere mantenuta in memoria principale, per cui il suo accesso richiede (lente) operazioni di I/O.

- (c) Quali sono le due principali implementazioni di tale modello?

Le liste di controllo di accesso (ACL) e le liste di abilitazioni (capability lists).

10. (a) A livello kernel, quale problema sorge nel passare da una macchina monoprocesso ad una macchina SMP?

# Sistemi Operativi

## Compitino Basic 18 Marzo 2005

L'accesso contemporaneo da parte di più processori alle stesse strutture dati del kernel. Questo è solitamente un problema lettori-scrittori.

- (b) Qual è la soluzione ovvia? È anche quella più efficiente? Cosa comporta in termini di latenza?

Limitare ad 1 processore per volta la possibilità di eseguire codice kernel (ad esempio con uno spinlock). Non è efficiente, perché processori diversi potrebbero voler operare su sezioni di kernel disgiunte. Inoltre la latenza aumenta parecchio (perché per gestire un I/O che termina bisogna aspettare che l'altro processore eventualmente in kernel finisca il suo lavoro).

11. (a) Si descriva sinteticamente il concetto di *middleware*.

Il middleware è uno strato tra le applicazioni e i sistemi operativi, che ha l'obiettivo di implementare un qualche modello di calcolo (strutture dati e operazioni) omogeneo e coerente tra macchine differenti.

- (b) Si dia un esempio di middleware per ognuno dei seguenti modelli: documenti distribuiti, file system distribuito, computazioni distribuite.

Nell'ordine: HTML + HTTP, XDR + NFS, RPC o RMI o CORBA+IIOP.