

Corso di Laurea in Informatica — Esame di Sistemi Operativi

1° dicembre 2003

1. (a) A che cosa serve il meccanismo delle *chiamate di sistema*?
- (b) In che cosa differisce dalle chiamate di procedura?

Risposta:

- (a) (2pt) Esso realizza l'interfaccia tra il nucleo ed i processi utente, e permette ai processi utente di *chiedere* al sistema operativo di svolgere delle operazioni per conto dei processi.
 - (b) (2pt) Rispetto alla chiamata di procedura, la chiamata di sistema è basata su un'interruzione software, e quindi include anche: modifica della modalità di funzionamento del processore (a supervisore), modifica dei livelli di priorità degli interrupt (*mascheramento*), invocazione di una routine di gestione dell'interruzione. E al termine dell'attività richiesta, tali passi sono eseguiti al contrario.
2. Si consideri un sistema con scheduling a priorità con due code H (a priorità maggiore) e L (a priorità minore). La coda L è round robin con quanto di 20msec; la coda H è FCFS.

Entrano quattro processi come segue:

- A arriva sulla coda L all'istante 0, e ha burst di CPU pari a 25 msec;
- B arriva sulla coda L all'istante 20 msec, e ha burst di CPU pari a 10 msec;
- C arriva sulla coda H all'istante 15 msec, e ha burst di CPU pari a 8 msec;
- D arriva sulla coda H all'istante 30 msec, e ha burst di CPU pari a 8 msec;

Si determini

- (a) l'ordine di terminazione dei 4 processi;
- (b) il tempo di attesa medio;
- (c) il tempo di reazione medio;

Si ignori il tempo di latenza del kernel.

Risposta: Si deve disegnare il diagramma GANTT e si vede che

- (a) (2pt) l'ordine di terminazione è C D B A;
- (b) (3pt) il tempo di attesa dei quattro processi è $A=8+18=26\text{ms}$, $B=8+8=16\text{ms}$, $C=0\text{msec}$, $D=0\text{msec}$. La media è $42/4=10,5$ msec.
- (c) (3pt) il tempo di reazione dei quattro processi è $A=0\text{msec}$, $B=8\text{msec}$, $C=D=0\text{msec}$; la media è $8/4=2\text{msec}$;

3. Si consideri un sistema con n risorse di uno stesso tipo (p.e., n stampanti), e dove ogni processo può utilizzare al massimo m risorse di tale tipo.

- (a) Senza alcun sistema di assegnazione delle risorse, quanti processi possiamo eseguire contemporaneamente garantendo comunque l'assenza di deadlock?
- (b) Usando l'algoritmo del banchiere, quanti processi potremmo mandare in esecuzione contemporaneamente?

Risposta:

- (a) (3pt) La situazione peggiore è quando tutti gli x processi detengono $m - 1$ risorse (per un totale di $x(m - 1)$ risorse allocate), e per continuare tutti ne chiedono ancora una. Per garantire l'assenza di deadlock, almeno una di queste richieste deve essere soddisfatta, e quindi deve essere $x(m - 1) \leq n - 1$, ossia $x \leq \frac{n-1}{m-1}$. Il valore massimo ammissibile è quindi $x = \left\lfloor \frac{n-1}{m-1} \right\rfloor$.
- (b) (3pt) Non c'è alcun limite: l'algoritmo automaticamente blocca i processi che dovessero richiedere troppe risorse rispetto a quante disponibili, e questo indipendentemente dal numero di processi in esecuzione.

4. (a) Descrivere la politica di allocazione di memoria *next-fit*.
- (b) Dite un vantaggio di questa politica rispetto alla *best-fit*.
- (c) Potrebbe essere applicata alla memoria paginata a due livelli?

Risposta:

- (a) (2pt) Nel contesto di un'allocazione contigua della memoria in partizioni multiple, la strategia next-fit identifica l'area da allocare cercando nella lista circolare delle aree libere la prossima sufficientemente grande da esaudire la richiesta.
 - (b) (2pt) In media basta scorrere metà lista, e non tutta.
 - (c) (2pt) No, perché non è allocazione di memoria contigua.
5. (a) Si descrivano i passi eseguiti da un driver per attivare una operazione di I/O bloccante con un dispositivo basato su interrupt.
- (b) Quale parte del sottosistema di I/O sblocca il processo e consente al driver di completare l'esecuzione, e in seguito a quale evento?

Risposta:

- (a) (3pt) I passi principali eseguiti dal driver sono i seguenti:
 - i. Verificare la validità dei parametri passati dagli strati superiori del kernel;
 - ii. Accodare la richiesta alla coda delle richieste (se implementa anche lo scheduling delle operazioni)
 - iii. Accedere al controller per attivare le operazioni (agendo sui registri come necessario).
 - iv. Passare il processo corrente nello stato wait, e richiamare lo scheduler per il context switch.
 - v. Verificare il risultato delle operazioni, accedere al controller (eventualmente, leggendo i dati ivi contenuti).
 - vi. Restituire il risultato allo strato superiore.
 - (b) (2pt) La routine di gestione degli interrupt, invocata dal controller stesso sollevando l'interrupt associatogli.
6. (a) Perché la mappa di bit di allocazione dei blocchi di un disco deve essere mantenuta sul disco stesso, e non in memoria principale?
- (b) 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.
- (a) (3pt) Perché è una informazione permanente, ossia sopravvive al sistema operativo stesso (deve essere ritrovata quando il file system viene rimontato). In teoria potrebbe essere mantenuta in memoria principale e copiata in memoria secondaria solo al momento dell'unmount, ma in tal caso un blocco della macchina, o uno smontaggio improprio (p.e. rimozione del supporto prima dell'unmount) lascerebbe il file system in uno stato inconsistente.
 - (b) (3pt) 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.
7. (a) Qual è il problema principale della matrice d'accesso? Come si risolve?
- (b) È possibile proteggere efficacemente i file di un sistema in cui gli utenti possono eseguire direttamente le operazioni di I/O su disco? Perché?

Risposta:

- (a) (3pt) La matrice d'accesso di norma è molto grande e sparsa. Si risolve usando rappresentazioni più compatte, come la lista di controllo degli accessi (ACL), o la lista delle abilitazioni (capability list).
 - (b) (2pt) No, non è possibile, perché gli utenti possono scavalcare qualsiasi protezione andando a leggere/scrivere direttamente i blocchi su disco (a meno di non usare tecniche crittografiche, e solo in lettura, o steganografiche).
8. (a) A livello kernel, quale problema sorge nel passare da una macchina monoprocesso ad una macchina SMP?
- (b) Qual è la soluzione ovvia? È anche quella più efficiente? Cosa comporta in termini di latenza?

Risposta:

- (a) (2pt) L'accesso contemporaneo da parte di più processori alle stesse strutture dati del kernel. Questo è solitamente un problema lettori-scrittori.

Risposta:

- (b) (3pt) 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).
9. La chiamata di sistema *accept()*, usata normalmente da un server per mettersi in attesa delle richieste dei client, richiede, oltre alla socket su cui ascoltare, anche (l'indirizzo di) una `struct sockaddr`.
- (a) Quale informazione contiene tale struttura all'uscita di *accept()* (ossia, dopo che la connessione è stata stabilita)?
- (b) Come può essere impiegata tale informazione dal server?

Risposta:

- (a) (2pt) La struttura contiene l'indirizzo della socket controparte, ossia del client. Ad esempio, nel dominio `AF_INET` (IPv4), questa struttura contiene l'indirizzo IP e la porta da cui si sta collegando il client.
- (b) (3pt) Questa informazione può essere usata dal server a scopo di controllo (verifica della legittimità del client), di logging (l'indirizzo, ora e data vengono scritti in un file di log), etc. Non serve per dare la risposta, come invece è necessario nelle collegamenti `connection-less`.