Compito di Sistemi Operativi, Proff. Mancini e Massaioli

3 giugno 2003

1. (punti: -1,4)
   Il meccanismo di traduzione degli indirizzi nella gestione della memoria virtuale con paginazione:

   (a) utilizza la Tabella delle Pagine, che riporta le corrispondenze tra indirizzi virtuali e indirizzi fisici di inizio delle pagine;

   (b) utilizza la Tabella delle Pagine che riporta le corrispondenze tra indirizzi virtuali e indirizzi astratti di inizio delle pagine;

   (c) utilizza la Tabella delle Pagine che riporta le corrispondenze tra indirizzi astratti e indirizzi fisici di inizio delle pagine;

   (d) è velocizzato con l'utilizzo di un dispositivo *hardware*, detto TLB (*Translation Lookaside Buffer*), che contiene i dati di traduzione per le pagine utilizzate più di recente;

   (e) è velocizzato con l'utilizzo di un algoritmo *software*, detto TLB (*Translation Lookaside Buffer*), che recupera rapidamente i dati delle pagine utilizzate più di recente;

   (f) richiede un accesso a disco per ogni accesso ad una locazione in una pagina di memoria principale;

   (g) richiede un accesso a disco per ogni accesso ad una locazione di memoria principale non utilizzata da più di 1024 cicli di *clock*;

   (h) è indipendente dalla nozione di processo, essendo lo spazio degli indirizzi virtuali unico per tutti i processi;

   (i) è fortemente legato alla nozione di processo, essendo lo spazio degli indirizzi virtuali unico per ogni processo;

   (j) è incompatibile con un modello segmentato della memoria;

   (k) è combinabile con un modello segmentato della memoria.

2. (punti: -1,4)
   In un sistema di comunicazione basato su indirizzamento indiretto:

   (a) la primitiva *send* richiede un identificatore del processo destinatario;

   (b) la primitiva *receive* richiede un identificatore del processo destinatario;

   (c) la primitiva *send* richiede un identificatore del processo mittente;

   (d) la primitiva *receive* richiede un identificatore del processo mittente;

   (e) la primitiva *send* non consente un'identificazione del processo mittente;

   (f) la primitiva *send* consente un'identificazione del processo mittente;

   (g) è possibile realizzare solo comunicazioni "uno-a-molti";

   (h) è possibile realizzare anche comunicazioni "uno-a-uno";

   (i) nessuna delle affermazioni precedenti è corretta.

3. (punti: -1,4)
   Ciascuna delle seguenti quattro condizioni: (1) le risorse sono seriali (mutua esclusione), (2) le risorse sono non prerilasciabili; (3) le risorse sono gestite con richiesta bloccante; (4) il sistema raggiunge uno stato di attesa circolare; è:

   (a) sufficiente affinché si verifichi lo stallo (o *deadlock*);

(b) necessaria affinché si verifichi lo stallo (o *deadlock*);

(c) necessaria e sufficiente affinché si verifichi lo stallo (o *deadlock*);

(d) sufficiente affinché si verifichi l'attesa indefinita (o *starvation*);

(e) necessaria e sufficiente affinché si verifichi l'attesa indefinita (o *starvation*);

(f) nessuna delle affermazioni precedenti è corretta.

4. (punti: -1,4)
I meccanismi per la mutua esclusione:

(a) permettono ai programmatori di evitare l'accesso contemporaneo da due processi a risorse che possono essere accedute da un solo processo alla volta;

(b) consentono ai programmatori di far accedere contemporaneamente due processi a risorse che altrimenti potrebbero essere accedute da un solo processo alla volta;

(c) permettono ai programmatori di evitare l'attesa indefinita (o *starvation*);

(d) permettono ai programmatori di evitare lo stallo (o *deadlock*);

(e) se mal implementati, possono causare attesa indefinita (o *starvation*);

(f) indipendentemente dalla loro implementazione, possono comunque causare attesa indefinita (o *starvation*);

(g) se mal implementati, possono contribuire alla condizione di stallo (o *deadlock*);

(h) indipendentemente dalla loro implementazione, contribuiscono alla condizione di stallo (o *deadlock*);

(i) nessuna delle affermazioni precedenti è corretta.

5. (punti: -1,4)
Quali fra le sequenti affermazioni sono corrette:

(a) le transizioni dei processi dallo stato di attesa allo stato di pronto competono alla funzione di gestione del processore. Questa funzione designa di volta in volta il processo pronto selezionandolo tra i processi in attesa;

(b) le transizioni dei processi dallo stato di attesa allo stato di pronto competono alla funzione di gestione del processore. Questa funzione designa di volta in volta il processo in esecuzione selezionandolo tra i processi pronti;

(c) le transizioni dei processi dallo stato di pronto allo stato di esecuzione competono alla funzione di gestione del processore. Questa funzione designa di volta in volta il processo pronto selezionandolo tra i processi in attesa;

(d) le transizioni dei processi dallo stato di pronto allo stato di esecuzione competono alla funzione di gestione del processore. Questa funzione designa di volta in volta il processo in esecuzione selezionandolo tra i processi pronti;

(e) il prerilascio del processore dovuto alla riattivazione di un processo con priorità superiore a quella del processo in esecuzione fa passare il processo in esecuzione nello stato di pronto;

(f) il prerilascio del processore dovuto alla riattivazione di un processo con priorità superiore a quella del processo in esecuzione fa passare il processo in esecuzione nello stato di attesa;

(g) nessuna delle affermazioni precedenti è corretta.

6. (punti: 6)
In al più 70 parole, descrivere le politiche di *scheduling First-Come-First-Served* ("in ordine di arrivo") e *Shortest-Process-Next* ("precedenza ai lavori brevi") ed indicare il principale svantaggio di ognuna.

7. (punti: 6)
Illustrare in al più 70 parole il concetto di stato sicuro introdotto nell'algoritmo del banchiere.

8. (punti: 6)
Considerare un insieme di sei processi P_1, P_2, P_3, P_4, P_5, e P_6 con i seguenti tempi di arrivo e tempi di esecuzione in millisecondi:

Processo	Tempo di arrivo	Tempo di esecuzione
P_1	2	2
P_2	3	14
P_3	7	25
P_4	14	29
P_5	15	13
P_6	18	6

Assegnare questo insieme di processi ad un processore in base alla politica Round Robin considerando un quanto di tempo di 7 millisecondi.

Calcolare il valor medio del tempo di attesa ed il valor medio del tempo di turnaround dei processi.

9. Considerare la seguente stringa di riferimenti di un processo alla memoria in un sistema con memoria virtuale
S = 2 7 1 8 2 8 1 8 2 8 4 5 9 0 4 5 2 3 5 4

   (a) (punti: 3)
   Illustrare il comportamento dell'algoritmo LRU di sostituzione delle pagine per una memoria fisica di 3 blocchi. Calcolare il numero di page fault che si verificano.

   (b) (punti: 3)
   Illustrare il comportamento dell'algoritmo Second Chance di sostituzione delle pagine per una memoria fisica di 3 blocchi. Calcolare il numero di page fault che si verificano.

10. (punti: 16)
Il processo P_A invia dati al processo P_B tramite un sistema a doppio *buffer*, costituito da due *array* di lunghezza N, B_1 e B_2, di cui i due processi condividono l'accesso: mentre P_B legge da un array, P_A può scrivere nell'altro.
Scrivere i codici dei processi P_A e P_B, utilizzando i semafori per la sincronizzazione (suggerimento: è una versione semplificata del problema lettori-scrittori).

Risposte Scritto di Sistemi Operativi - Data,

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

1. (punti: -1,4) **a** **b** **c** **d** **e** **f** **g** **h** **i** **j** **k** **l**
2. (punti: -1,4) **a** **b** **c** **d** **e** **f** **g** **h** **i** **j** **k** **l**
3. (punti: -2,4) **a** **b** **c** **d** **e** **f** **g** **h** **i** **j** **k** **l**
4. (punti: -1,4) **a** **b** **c** **d** **e** **f** **g** **h** **i** **j** **k** **l**
5. (punti: -1,4) **a** **b** **c** **d** **e** **f** **g** **h** **i** **j** **k** **l**
6. (punti: 6)

7. (punti: 6)

8. (punti: 6)

9. (a) (punti: 3)

F_0																				
F_1																				
F_2																				
F_3																				
F_4																				
F_5																				

(b) (punti: 3)

F_0																				
F_1																				
F_2																				
F_3																				
F_4																				
F_5																				