

Cognome: _____ Nome: _____ Matricola: _____

A

Sistemi Operativi — A.A. 2005-2006, prova scritta del 28 aprile 2006

Libri e appunti chiusi. Vietato comunicare con chiunque. Vietato l'uso di cellulari, calcolatrici, palmari e affini.

Tempo a disposizione: 60 minuti. Le domande sono etichettate con 1,2 o 3 asterischi:

* = domanda semplice, valutazione alta, rispondi a queste prima delle altre

** = domanda di media difficoltà

*** = domanda difficile, valutazione bassa, rispondi dopo aver risposto alle altre

Domanda 1

1. * Descrivi l'architettura hw per memoria virtuale con tabella delle pagine a due livelli. Spiega anche in che senso tale architettura risolve il problema delle tabelle delle pagine molto grandi.

Vedi materiale didattico.

Tale tecnica permette di avere la tabella delle pagine in memoria virtuale e quindi non necessariamente essa occupa memoria reale per l'intera grandezza. In memoria reale deve essere sempre presente la tabella di livello più alto, che occupa tipicamente una sola pagina.

2. ** Considera l'istruzione macchina "zero 0x007FE800 0x3000" che azzerava 0x3000 bytes a partire dall'indirizzo 0x007FE800. Considera una architettura con page table a due livelli pentium-like a 32 bit (pagine di 4KB, 4 byte per entry). Quanti page faults può generare al più tale istruzione durante la fase di esecuzione (non considerare eventuali page faults per la fase di fetch)?

offset: 12 bit, pagina: 20 bit (10 primo livello, 10 secondo livello)

Indirizzi acceduti da 0x007FE800 a 0x008017ff.

pagine accedute: 0x007FE, 0x007FF, 0x00800, 0x00801 (possono provocare 4 page faults)

pagine della page table accedute: 0 e 1 (possono provocare 2 page faults)

Totale: max 6 page faults.

Domanda 2

1. * Che cosa è un "system call"? Elenca le categorie di funzionalità che ritieni più importanti tra quelle fornite dalle system call.

system call: vedi materiale didattico.

categorie: i/o, gestione processi, richiesta memoria

2. ** Supponi che in un sistema esistano due processi P e Q: P puramente cpu bound (nessun I/O) e Q I/O bound che usa pochissimo la cpu. Considera una politica di scheduling della cpu che dia priorità a Q e sia preemptive. Descrivi come cambiano gli stati dei due processi e che relazione c'è in ogni istante tra lo stato dell'uno e dell'altro.

P oscilla tra gli stati ready e running. Q cicla tra gli stati ready running e blocked e passa la maggior parte del tempo in blocco, poco tempo in running (fino alla prossima system call) e pochissimo in ready (il tempo che lo scheduler faccia preemption su P). Q è in blocco implica P è running, Q running implica P ready.

Domanda 3

1. ** Considera un sistema che è destinato ad eseguire sia processi cpu bound sia processi i/o bound. Quali pensi siano i processi che devono ottenere la cpu con priorità? Perché?

I processi I/O bound devono avere priorità poiché occupano la cpu per poco tempo ma il valore ai fini dell'avanzamento del processo è molto. Inoltre questo permette, con poca spesa di tempo di cpu, di ottenere attività da fare per gli altri dispositivi aumentando l'utilizzo di tutto il sistema.

2. *** Considera la strategia di scheduling feedback con 5 code (0..4) in cui la priorità di un processo diminuisce di 1 (cioè migliora) ogni volta che il processo va in blocco e aumenta di 1 (cioè peggiora) ogni volta che il quanto di tempo q scade. Supponi che nel sistema vi siano n processi i/o bound Q_i ($i=1..n$) e un processo cpu bound P , tutti di durata infinita. Dai un esempio di comportamento dei processi Q_i che mette P in starvation (a regime).

Si ha starvation se i processi sono tutti sempre in coda migliore di quella di P e tra gli n processi ce n'è sempre almeno uno che non è in blocco. Ad esempio si ha starvation per P se ciascun cpu burst dei Q_i dura x con $x < q$ (quindi tutti i Q_i sono sempre in coda 0) e ciascun i/o burst dei Q_i dura y con $y < (n-1)x$ (quindi ciascun Q_i diviene ready prima che gli altri $n-1$ abbiano finito i loro cpu burst).

Domanda 4

1. ** Descrivi la tecnica denominata RAID 5. Quale è la sua tolleranza ai guasti? Quale è il collo di bottiglia dell'architettura? in quali casi si manifesta?

Vedi materiale didattico.

Domanda 5

1. * Descrivi l'algoritmo di page replacement denominato CLOCK nella sua variante che considera il bit di modifica.

Vedi materiale didattico.

2. ***Mostra un esempio in cui aging si comporta in modo diverso da LRU nella gestione di 3 frame. Per l'esempio fornisci una sequenza di accessi, i parametri scelti per aging e una giustificazione del motivo per cui c'è il comportamento è diverso dai LRU.

Parametri di aging: scansione ogni 2 istanti, 3 bit.

1 1 (scansione) 2 2 (scansione) 3 3 (scansione) 3 2 (scansione) 1 1 (scansione) 4

Alla momento di richiedere 4 gli stimatori di aging per le tre pagine sono:

1: 100

2: 010

3: 011

Viene quindi tolto il frame 2 anche se LRU sceglierebbe 3. Infatti aging non ha abbastanza informazione per dire che 2 è stato acceduto più recentemente di 3.

Cognome: _____ Nome: _____ Matricola: _____

B

Sistemi Operativi — A.A. 2005-2006, prova scritta del 28 aprile 2006

Libri e appunti chiusi. Vietato comunicare con chiunque. Vietato l'uso di cellulari, calcolatrici, palmari e affini.

Tempo a disposizione: 60 minuti. Le domande sono etichettate con 1,2 o 3 asterischi:

* = domanda semplice, valutazione alta, rispondi a queste prima delle altre

** = domanda di media difficoltà

*** = domanda difficile, valutazione bassa, rispondi dopo aver risposto alle altre

Domanda 1

1. * Descrivi l'architettura hw per memoria virtuale con tabella delle pagine invertita. Spiega anche in che senso permette di evitare di avere tabelle delle pagine molto grandi.

Vedi materiale didattico.

Tale tecnica permette di avere una entry per ciascun frame, quindi l'occupazione e' proporzionale alla ampiezza della memoria fisica a disposizione del sistema e non alla grandezza dello spazio di indirizzamento virtuale dei processi.

1. ** Considera l'istruzione macchina "zero 0x017FE800 0x4000" che azzerava 0x4000 bytes a partire dall'indirizzo 0x017FE800. Considera una architettura con page table a due livelli pentium-like a 32 bit (pagine di 4KB, 4 byte per entry). Quanti page faults può generare al più tale istruzione durante la fase di esecuzione (non considerare eventuali page faults per la fase di fetch)?

offset: 12 bit, pagina: 20 bit (10 primo livello, 10 secondo livello)

Indirizzi acceduti da 0x017FE800 a 0x018027ff.

pagine accedute: 0x017FE, 0x017FF, 0x01800, 0x00801, 0x00802 (possono provocare 5 page faults)

pagine della page table accedute: 0 e 1 (possono provocare 2 page faults)

Totale: max 7 page faults.

Domanda 2

1. * Mostra il diagramma di stato di un processo in cui viene considerata la possibilità di sospensione del processo stesso (quello a 7 stati). Perché un processo dovrebbe essere sospeso? quando può essere riattivato?

vedi materiale didattico.

2. ** Supponi che in un sistema i processi $P_1 \dots P_{10}$ sono puramente cpu bound (nessun I/O) e il processo Q è I/O bound e usa pochissimo la cpu. Considera una politica di scheduling della cpu di tipo round robin semplice. Descrivi come cambia lo stato di Q e quanto tempo permane nello stato di in ready.

Q cicla tra gli stati ready running e blocked e passa la maggior parte del tempo in blocco, poco tempo in running (fino alla prossima system call). Passa in ready 10 quanti di tempo ogni volta che esce dal blocco.

Domanda 3

1. ** Considera un sistema che è destinato ad eseguire sia processi cpu bound sia processi interattivi. Quali pensi siano i processi che devono ottenere la cpu con priorità? Perché?

I processi interattivi devono avere priorità poiché occupano la cpu per poco tempo ma il valore ai fini dell'avanzamento del processo è molto, infatti, questo permette, con poca spesa di tempo di cpu, di mettere in grado l'utente di interagire ancora con la macchina.

2. *** Considera la strategia di scheduling feedback con 6 code (0..5) in cui la priorità di un processo diminuisce di 1 (cioè migliora) ogni volta che il processo va in blocco e aumenta di 1 (cioè peggiora) ogni volta che il quanto di tempo t scade. Supponi che nel sistema vi siano p processi i/o bound Z_i ($i=1..p$) e un processo cpu bound W , tutti di durata infinita. Dai un esempio di comportamento dei processi Z_i che mette W in starvation (a regime).

Si ha starvation se i processi sono tutti sempre in una coda migliore di W e tra i p processi Z_i ce n'è sempre almeno uno che non è in blocco. Ad esempio si ha starvation per W se ciascun cpu burst dei Z_i dura x con $x < t$ (quindi tutti gli Z_i sono sempre in coda 0) e ciascun i/o burst dei Z_i dura y con $y < (p-1)x$ (quindi ciascun Z_i diviene ready prima che gli altri $p-1$ abbiano finito i loro cpu burst).

Domanda 4

2. ** Descrivi la tecnica denominata RAID 4. Quale è la sua tolleranza ai guasti? Quale è il collo di bottiglia dell'architettura? in quali casi si manifesta?

Vedi materiale didattico.

Domanda 5

3. * Descrivi l'algoritmo di page replacement denominato CLOCK. Perché è considerato una approssimazione di LRU.

Vedi materiale didattico.

4. ***Mostra un esempio in cui aging si comporta in modo diverso da LRU nella gestione di 4 frame. Per l'esempio fornisci una sequenza di accessi, i parametri scelti per aging e una giustificazione del motivo per cui c'è il comportamento è diverso dai LRU.

Parametri di aging: scansione ogni 2 istanti, 4 bit.

1 4 (scansione) 2 2 (scansione) 3 3 (scansione) 3 2 (scansione) 1 4 (scansione) 5

Alla momento di richiedere 4 gli stimatori di aging per le tre pagine sono:

1: 1000

2: 0101

3: 0110

4: 1000

Viene quindi tolto il frame 2 anche se LRU sceglierebbe 3. Infatti aging non ha abbastanza informazione per dire che 2 è stato acceduto più recentemente di 3.