

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI
ANNO ACCADEMICO 2010/2011 - PROVA DI CONCORRENZA

09 febbraio 2012

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione in tutti i fogli prima di svolgere ogni altro esercizio. Scrivere esclusivamente a penna senza abrasioni. E' vietato l'uso delle penne cancellabili, della matita, dei coprenti bianchi per la correzione (bianchetto) e la scrittura in colore rosso (riservato alla correzione). Il compito e' formato da due fogli, quattro facciate compresa questa. Le soluzioni che si vogliono sottoporre per la correzione devono essere scritte negli spazi bianchi di questi fogli. Non verranno corretti altri supporti. E' obbligatorio consegnare il compito, e' possibile chiedere che esso non venga valutato scrivendo "NON VALUTARE" in modo ben visibile nella prima facciata.

Esercizio 1: Il monitor seq da implementare deve fornire due procedure entry:

procedure entry void enter(void);

procedure entry void exit(void);

I processi chiamano i servizi di seq come segue:

....

seq.enter();

/* codice controllato da seq */

seq.exit();

....

Tutti i processi che ne facciano richiesta possono entrare nel codice controllato da seq contemporaneamente (questa NON e' una sezione critica!). Exit deve garantire che i processi escano dal codice controllato tutti insieme. Per esempio se tre processi a,b e c chiamano seq.enter ogni processo non potra' completare la exit se prima tutti gli altri due non hanno chiamato la exit. Se nel frattempo altri processi dovessero chiamare la seq.enter occorrera' attendere anche questi ultimi.

Esercizio 2: Si considerino le seguenti funzioni atomiche foo e bar e si verifichi se e' possibile usarle (o meno) al posto della test&set per il supporto di sezioni critiche (i parametri sono passati per indirizzo):

2a) foo(x,y) = <x1=y*2, y1=x/2, x=y1, y=x1> (con x, y interi)

2b) bar(x,y) = <x1=y*2, y1=y/2, x=y1, y=x1> (con x, y interi)

Esercizio 3: I semafori casual operano come normali semafori ma all'atto di una V se piu' processi sono in attesa, ne viene risvegliato uno a caso. Si ponga che nel sistema sia presente anche un contatore atomico. Questo contatore, implementato in hardware e' unico in tutto il sistema e gestisce numeri interi a precisione illimitata.

Atomic very long int globalcounter(void) { static very long int privateval; return privateval++ }

implementare le funzioni Gcsin e Gcsout, prive di argomenti, che consentano di avere un servizio di mutua esclusione globale fair (cioe' con la garanzia che i processi entrino nella mutua esclusione uno solo alla volta ma in ordine fifo).

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2010/2011 – PARTE GENERALE
 09 febbraio 2012

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione in tutti i fogli prima di svolgere ogni altro esercizio. Scrivere esclusivamente a penna senza abrasioni. E' vietato l'uso delle penne cancellabili, della matita, dei coprenti bianchi per la correzione (bianchetto) e la scrittura in colore rosso (riservato alla correzione). Il compito e' formato da due fogli, quattro facciate compresa questa. Le soluzioni che si vuole sottoporre per la correzione devono essere scritte negli spazi bianchi di questi fogli. Non verranno corretti altri supporti.

E' obbligatorio consegnare il compito, e' possibile chiedere che esso non venga valutato scrivendo "NON VALUTARE" in modo ben visibile nella prima facciata.

Per svolgere questo compito occorre solo una penna e un documento di identità valido. La consultazione o anche solo la disponibilità di altro materiale comporterà l'annullamento del compito (con la stessa penalizzazione di punteggio della grave insufficienza per la prossima esercitazione scritta).

Esercizio 1: Considerare I seguenti processi gestiti mediante uno scheduler round robin con timeslice di 3ms su una macchina SMP a due processori:

P1: cpu 5ms; I-O 4 ms; cpu 2 ms; I-O 4 ms; cpu 5ms

P2: cpu 2ms; I-O 4 ms; cpu 2 ms; I-O 1ms; cpu 5ms

P3: cpu 4ms; I-O 3 ms; cpu 3 ms; I-O 2ms; cpu 3ms

P4: cpu 15ms; I-O 1ms

l'Input-Output avviene su un'unica unita'.

Esercizio 2: Trovare una stringa di riferimenti ad una memoria di 6 frame per la quale LRU causi il triplo di page fault rispetto a MIN.

Esercizio 3: Detti rispettivamente R il numero di riga e C il numero di colonna, soddisfare la richiesta corrispondente a $(R+C)\%3$ (esercizio obbligatorio):

- 0) Confrontare la paginazione e la segmentazione di memoria
- 1) Confrontare l'allocazione concatenata e l'allocazione indicizzata per file system
- 2) Confrontare microkernel e kernel monolitici

Confrontare significa: esporre le caratteristiche delle due soluzioni, indicare le proprietà differenti e quali sono gli scenari applicativi dell'una o dell'altra.

NOTA: tutti gli esercizi verranno valutati solo se le risposte saranno corredate da motivazioni e dimostrazioni scritte in Italiano o in Inglese (corretto o almeno comprensibile). La presenza nella soluzione di un esercizio di solo codice sorgente/tabelle/scarabocchi/simboli vari comporta la non valutazione dell'esercizio.

