

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2011/2012

24 gennaio 2013

Esercizio -1: Essere iscritti su AlmaEsami 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 tre fogli, sei 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 (verrà automaticamente valutato gravemente insufficiente).

Esercizio c.1: (a) Scrivere un monitor `nmbb` che realizzi un buffer limitato (di ampiezza `BUFSIZE`) che consenta alle chiamate `write` (inserimento nel buffer) e `read` (lettura da buffer) di operare su vettori di più elementi. In particolare l'interfaccia delle procedure `entry` da implementare e' la seguente:

```
procedure entry write(int n, struct elem *v);
procedure entry read(int m, struct elem *w);
```

se n o m sono maggiori di `BUFSIZE` le funzioni non devono fare nulla (caso di errore).

La funzione `write` deve attendere che ci sia spazio nel buffer per inserire n elementi (il vettore v conterra' n elementi). Solo quando e' possibile completare l'operazione vengono inseriti tutti gli elementi di v nel buffer

La funzione `read` attende che vi siano almeno m elementi nel buffer quindi estrae dal buffer (in ordine FIFO) m elementi e li copia nel vettore w .

(b) sono possibili casi di deadlock? (motivare dettagliatamente la risposta).

Esercizio c.2: Considerare le seguenti funzioni atomiche (le variabili sono passate per indirizzo)

```
foo(x,y): <x=2*y, y=2*x>
bar(z,t): <z=z xor t, t=t xor z>
```

Quali fra queste funzioni possono essere usate per implementare un servizio di sezioni critiche (in modo simile a quanto visto a lezione con la `Test&Set`)?

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

```
P1: cpu 5 ms; I-O 4 ms; cpu 2 ms; I-O 4 ms; cpu 5 ms
P2: cpu 2 ms; I-O 4 ms; cpu 2 ms; I-O 1 ms; cpu 5 ms
P3: cpu 4 ms; I-O 3 ms; cpu 3 ms; I-O 2 ms; cpu 3 ms
P4: cpu 13 ms; I-O 1 ms
```

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

Esercizio g.2:

Nel corso abbiamo incontrato concetti che possono apparire simili, o diversi approcci di soluzione per lo stesso problema.

Il candidato scelga due dei tre confronti proposti e indichi per ognuno: in quale ambito dei sistemi operativi si applicano i concetti espressi, quali sono i punti in comune e quali le differenze fra gli elementi da confrontare.

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