

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2008/2009
PROVA DI CONCORRENZA – 20 GENNAIO 2009

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.

Esercizio 1:

Scrivere un monitor vbb che implementi un buffer limitato che mantenga bloccati al piu' NSCR produttori/scrittori. Nel caso ci siano gia' NSCR scrittori in attesa, e un nuovo scrittore tenti di inserire un messaggio il primo messaggio nel buffer viene perso e il relativo scrittore sbloccato.

La "vita" dei processi che usano il monitor puo' essere cosi' schematizzata:

```
producer[i, i=1,...,NPRODUCER]: process
    while (1):
        ptmp=produce();
        vbb.enqueue(ptmp);

consumer[j, i=1,...,NCONSUMER]: process
    while (1)
        ctmp=vbb.dequeue();
        consume(ctmp);
```

Esercizio 2:

Con quali delle seguenti funzioni e' possibile implementare un meccanismo di entrata e uscita da una sezione critica? Motivare la risposta con un frammento di codice o una spiegazione per l'impossibilita'.

1. $\text{minmax1}(x,y) = \langle xx=x, yy=y, x=\min(xx,yy), y=\max(xx,yy) \rangle$
2. $\text{minmax2}(x,y) = \langle yy=y, y=\max(x,y), x=\min(y,yy) \rangle$
3. $\text{minmax3}(x,y) = \langle x=\min(x,y), y=\max(x,y) \rangle$

Esercizio 3:

Con due soli semafori binari globali e una funzione getpid che restituisca un intero univoco che identifichi il processo chiamante e' possibile realizzare un servizio di semafori generali. I due semafori binari globali si accedono tramite le operazioni P0() V0() P1() e V1();

Scrivere il codice in grado di implementare le primitive dei semafori generali P() e V() usando I due semafori binari globali (Nota: l'implementazione non sara' efficiente ma *non* deve prevedere busy wait).