Nome/cognome _____ N. di matricola (10 cifre) _____ Posizione: Riga _____ Col _____

UNIVERSITA' DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004
COMPITO CONCORRENZA - 13 Febbraio 2004

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.
Esercizio 0: Su entrambi i fogli, scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1:
Un gestore di dispositivo di memoria secondaria è così organizzato. I processi utente **P[i]**, con **i** compreso fra **1** e **N,** così come il processo del kernel **GM** che gestisce la memoria, possono richiedere la lettura o la scrittura di una particolare traccia del dispositivo stesso. Le richieste assumono la forma di un oggetto **Req**, che comprende varie informazioni fra cui il richiedente (**req.origin**) e l'identificatore della traccia (**req.track**).

- Scrivere un gestore **mng** utilizzando monitor
- Le tracce devono essere eseguite nell'ordine indicato da una delle versioni dell'algoritmo dell'ascensore (a scelta, SCAN,C-SCAN,LOOK,C-LOOK).
- Le richieste del processo **GM** sono privilegiate, cioè vengono gestite prima delle altre.
- Il processo **GM** è unico, quindi in ogni istante può esserci al più una richiesta da **GM.**
- Dopo aver eseguito la richiesta di **GM**, il processo torna ad eseguire le richieste dei processi utente, se ve ne sono.
- E' possibile utilizzare array di condizioni o altre strutture dati.

. La vita dei vari componenti è illustrata di seguito. Implementate i metodi sottolineati.

```
process Dispositivo {
  while (true) {
    Req req = mng.getRequest();
    Resp resp = esegui(req);
    mng.addResponse(req, resp);
  }
}
```

```
process P[i] {
  while (true) {
    Req req = genera();
    mng.addRequest(req);
    Resp resp =
      mng.getResponse();
  }
}
```

```
process GM {
  while (true) {
    Req req = genera();
    mng.addPrivilegedRequest(req);
    Resp resp =
      mng.getResponse();
  }
}
```

Esercizio 2

E' possibile realizzare un servizio di mutua esclusione tipo test&set utilizzando le seguenti funzioni? Se la riposta è positiva, illustrate come. Se la risposta è negativa, spiegate perchè.

a) rswap(a,b,c) = <if(random()%2) then a=b,b=c else c=b,b=a>dove random produce un numero random intero a 32bit.
b) asqrt(x,y)= < x=y=sqrt(x) > dove x,y sono numeri floating point e sqrt è la funzione radice quadrata.

Esercizio 3

Sia data l'astrazione di semaforo **piuomeno**. Un semaforo **piuomeno** può essere rappresentato dalla seguente interfaccia:

```
interface sempiuomeno  {

  /**
   * init_v: il valore iniziale del semaforo
   * max_v e' il valore massimo
   */
  public sempiuomeno (int init_v, int max_v);

  /*
   * val puo' assumere valori positivi o negativi
   */
  op (int val);
}
```

L'invariante del semaforo è **0<=init_v + somma val_i <=max_v**. Cioè la **op** si comporta come una o più **P** se val è negativo, come una o più **V** se **val** è positivo. Dimostrare che i semafori piuomeno hanno lo stesso potere espressivo dei semafori generali.