

nome e cognome _____ numero di matricola 16 74 _____

CORSO DI SISTEMI OPERATIVI
CORSO DI LAUREA IN INFORMATICA - UNIVERSITA' DI BOLOGNA
PRIMO APPELLO DELLA SESSIONE ESTIVA AA 1999/2000

Esercizio 0. Scrivere correttamente il proprio nome e cognome e numero di matricola in tutti i fogli.

Quattro robot si muovono in uno spazio bidimensionale soddisfacendo richieste di posizionamento provenienti da numerosi client.

(si assuma lo spazio limitato, di forma quadrata, con coordinate intere, e le richieste dei clienti si assumano coerenti con le dimensioni).

I clienti richiedono posizionamenti con il seguente protocollo:

```
robotcontrol.moveto(X,Y)
// operate at X,Y
robotcontrol.go()
```

La vita del robot puo' essere rappresentata dal seguente processo:

```
robot[i, i=0..3]: process
  // move the robot to the starting point
  while (true)
  {
    (X,Y)=robotcontrol.getrequest(i);
    // move the robot to the requested position
    robotcontrol.requestcompleted(i);
  }
```

Il robot deve attendere prima di muoversi dalla posizione assunta che il cliente abbia terminato le operazioni che desiderava compiere a quelle coordinate.

Esercizio 1. implementare il monitor robotcontrol. I robot devono utilizzare la politica del minimo spostamento (SMF, shortest move first) per la scelta della successiva posizione da assumere.

Esercizio 2. implementare le funzioni robotcontrol.x facendo uso di message passing sincrono e di un processo controllore.

nome e cognome _____ numero di matricola 16 74 _____

Esercizio 3. Il problema con politica SMF e' soggetto a casi di starvation?
Motivare la risposta.

4) Con le premesse del problema molti sviluppatori sceglierebbero una suddivisione dello spazio in quattro aree quadrate dimezzando i lati del quadrato originale e assegnerebbero staticamente ogni robot ad un'area. Discutere dal punto di vista della concorrenza pro e contro di questa scelta.

5) Considerare le seguenti implementazioni di mutex_in e mutex_out

```
mutex_in:
    var xin;
    xin=0;
    do
        swap(xin,y)
    while (xin=0);
```

```
mutex_out;
    var xout;
    xout=1;
    do
        swap(xout,y)
    while (xout=1);
```

dove swap e' una istruzione atomica.

Questa implementazione risolve il problema della mutua esclusione?
perche'?