

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2013/2014
 03 giugno 2014

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 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.
 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: Nel giorno di sabato in un albergo ad Haifa funziona un solo ascensore in modo automatico. L'ascensore si ferma a ogni piano a partire dal piano terra all'ultimo piano e poi inverte la marcia fermandosi ancora ad ogni piano e così torna al piano terra. Per tutto il giorno di sabato l'ascensore funziona in questo modo.

Il programma dell'ascensore e' il seguente.

```
up,down=0,1
process elevator:
  while True:
    for floor in range(0,N):
      ...chiude le porte, va al piano floor, apre le porte...
      sabelev.atffloor(floor,up)
    for floor in range(N,0,-1):
      ...chiude le porte, va al piano floor, apre le porte...
      sabelev.atffloor(floor,down)
```

Gli utenti quindi aspettano l'ascensore, salgono quando arriva l'ascensore che va nella direzione desiderata e scendono al piano dove sono diretti. Per usare l'ascensore da un piano gli utenti richiamano questa funzione:

```
def changeffloor(from, to):
  if from != to and 0<=from<N and 0<=to<N:
    sabelev.enter(from, to)
    sabelev.exit(from, to)
```

dove *from* e' il piano di partenza e *to* il piano di arrivo. Quando l'ascensore arriva ad un piano non può ripartire se non sono scese tutte le persone che volevano giungere a quel piano. Per semplicità assumeremo che la capienza della cabina non sia limitata.

Esercizio c.2: Siano dati 4 processi:

```
prchar="IATL"
Process i: i=range(4)
while True:
  sync(i)
  print(prchar[i])
```

Scrivere la funzione *sync* in modo che in un sistema di processi a memoria privata, usando message passing asincrono, l'unico output possibile sia la stringa di lunghezza infinita: "ITALIAITALIAITALIA...."

Esercizio g.1: Costruire un Grafo di Holt con 6 processi e 6 classi di risorse in modo che valgano contemporaneamente le seguenti proprietà:

- la situazione sia di Deadlock
- non sia sufficiente eliminare un processo perché il Deadlock venga risolto
- ogni processo abbia esattamente una richiesta in sospeso
- il grafo sia connesso (ci sono archi fra nodi di partizioni diverse comunque si partizioni l'insieme dei nodi)

Esercizio g.2: Rispondere alle seguenti domande:

1. Quali sono le differenze fra un device driver di una unità con funzionalità di DMA e quello di una unità senza DMA?
2. L'algoritmo del Banchiere e il grafo di Holt servono entrambi per gestire i Deadlock. Quali sono le differenze, quando e perché si usa l'uno e quando e perché l'altro?
3. Quali problemi risolve la paginazione e quali la segmentazione?