

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 PROVA SCRITTA DI SISTEMI OPERATIVI
 ANNO ACCADEMICO 2016/2017
 22 gennaio 2018

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: L'incrocio fra una strada e un canale è regolato da un ponte mobile come quello illustrato dall'apposito segnale.

Ovviamente le auto possono attraversare il ponte solo se è abbassato e si può alzare il ponte se non ci sono auto che lo stanno attraversando. Il ponte deve essere alzato per far passare le imbarcazioni.

Il canale in corrispondenza del ponte ha una larghezza che consente il passaggio di una imbarcazione alla volta (indipendentemente dalla direzione di provenienza) e le imbarcazioni in attesa di attraversare il ponte non possono superarsi a vicenda (l'ordine di accesso al canale sotto al ponte è FIFO).

Le auto per attraversare il ponte usano il seguente protocollo:

```
bridge.car_enter(direction)
... attraversa il ponte
bridge.car_exit(direction)
```

Le imbarcazioni usano il protocollo:

```
bridge.boat_enter(direction)
... passa sotto al ponte
bridge.boat_exit(direction)
```

dove direction vale 0 o 1 per riconoscere le due diverse direzioni delle auto o delle imbarcazioni. Occorre scrivere la soluzione in modo da evitare casi di starvation.



Esercizio c.2: Facendo uso di semafori ordinari implementare semafori ternari che possano assumere valori -1, 0, +1.

L'invariante dei semafori ternari di questo esercizio è:

$$nP - 1 \leq nV + \text{init} \leq nP + 1$$

dove nP è il numero di operazioni P completate, nV il numero delle operazioni V completate e init è il valore iniziale del semaforo.

Esercizio g.1: Considerare i seguenti processi gestiti mediante uno scheduler round robin con timeslice di 4ms su una macchina SMP:

```
P1: cpu 5 ms; I-O 4 ms; cpu 2 ms
P2: cpu 2 ms; I-O 4 ms; cpu 2 ms
P3: cpu 3 ms; I-O 3 ms; cpu 3 ms
P4: cpu 10 ms; I-O 1 ms
```

l'Input-Output avviene su un'unica unità. Calcolare il tempo necessario a completare l'esecuzione dei 4 processi al variare del numero di processori presenti nel sistema.

Esercizio g.2: Rispondere alle domande seguenti:

- A cosa serve partizionare un disco? Fornire esempi pratici di uso di dischi partizionati.
- Quale è la differenza fra TLB miss e page fault nella gestione della memoria virtuale?
- Fornire esempi di file system con allocazione contigua, e spiegare perché sarebbe inefficiente usare altri metodi di allocazione nei casi d'uso tipici di questi file system.
- Come funziona la tecnica denominata *aging* e come riesce ad evitare la starvation negli scheduler a priorità?