

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010  
CONCORRENZA – 05 maggio 2010

**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:** Due nazioni limitrofe che chiameremo 0 e 1 hanno stabilito un numero massimo MAXIMM di immigrati che soggiornano nell'altra nazione.

Quando un cittadino di una nazione vuole andare nell'altra, alla frontiera si rivolge agli ufficiali di immigrazione dell'altra nazione e viene posto in attesa se il numero massimo di immigrati e' stato gia' raggiunto **in una delle due nazioni**.

Pertanto il cittadino di 0 che vuole andare in 1 viene messo in stato di attesa sia se ci sono gia' MAXIMM cittadini di 0 in 1 sia se ci sono MAXIMM cittadini di 1 in 0.

La "vita" dei cittadini e' quindi:

```
cittadinoi,j, i ∈ {0,1}, j ∈ {1,..., Ncittadinii};
while(1) {
    immigration01.emigrate(i); ....
    immigration01.returnhome(i); ....
}
```

Il problema (non la soluzione) presenta problemi di deadlock o di starvation?

Scrivere il monitor **immigration01** con due procedure entry emigrate(int citizenship) e returnhome(int citizenship).

(il parametro e' la propria cittadinanza, I cittadini di 0 mettono 0 come parametro e quindi vogliono andare in 1, e viceversa).

La soluzione chiaramente non deve inserire problemi di deadlock o di starvation che non siano propri del problema.

**Esercizio 2:** Un semaforo cartesiano planare sghembo prevede due operazioni:  $P_{\eta\beta}(x,y)$  e  $V_{\eta\beta}(x,y)$ . Il valore iniziale di tali semafori e' una coppia (xinit, yinit). X, y, xinit e yinit sono numeri reali non negativi.

L'invariante del semaforo cartesiano planare sghembo e':

$$\left( x_{init} + \sum_{i \in Completed V_{\eta\beta}} x_i - \sum_{j \in Completed P_{\eta\beta}} x_j \right) \left( y_{init} + \sum_{i \in Completed V_{\eta\beta}} y_i - \sum_{j \in Completed P_{\eta\beta}} y_j \right) \geq 0$$

I semafori ordinari e quelli cartesiani planari sghembi hanno lo stesso potere espressivo? Produrre una dimostrazione.

**Esercizio 3:** La seguente classe twostep puo' essere usata al posto della Test&Set?

```
Class twostep {
    private int a;
    private int b;
    void twostep(void) {a=1;b=0}
    atomic int op (int x) {
        int result=b;
        b=a;
        a=x;
        return result;
    }
}
```

Produrre una dimostrazione.



UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010  
 PARTE GENERALE – 12 maggio 2010

**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.

Per svolgere questo compito occorre solo una penna e un documento di identita' valido. La consultazione o anche solo la disponibilita' di altro materiale comportera' l'annullamento del compito (con la stessa penalizzazione di punteggio della grave insufficienza per la prossima esercitazione scritta).

**Esercizio 1:** Data una memoria di 3 frame e un unico processo in esecuzione che necessita di 4 pagine, si costruisca una stringa di riferimenti infinita  $s$  tale che, detto  $N_{MRU}(s, t)$  il numero di page fault generati dall'algorithmo MFU (most recently used) al tempo  $t$  (dopo  $t$  riferimenti) e similmente  $N_{LRU}(s, t)$  il numero di page fault generati dall'algorithmo LFU (least recently used) si abbia:

- $\forall t \in \mathbb{N} \quad N_{MRU}(s, t) \leq N_{LRU}(s, t)$
- $\lim_{x \rightarrow \infty} (N_{LRU}(s, t) - N_{MRU}(s, t)) = \infty$

**Esercizio 2:** Siano dati una macchina monoprocesso e uno scheduler per processi periodici, i.e. programmi che vengono riattivati a intervalli regolari di tempo (detti periodi). Lo scheduler gestisce tre processi  $p_1$ ,  $p_2$ ,  $p_3$  che hanno prioritá statica:  $p_1$  ha la prioritá massima,  $p_2$  media,  $p_3$  minima.  $p_1$  ha un periodo di 3ms e la sua esecuzione occupa la CPU per 1ms,  $p_2$  ha un periodo di 5 ms e deve usare la CPU per 2 ms,  $p_3$  ha un periodo di 10ms.

L'esecuzione (periodica) di  $p_1$ ,  $p_2$  e  $p_3$  viene iniziata nello stesso istante.

Si chiede quale sia l'utilizzo massimo di processore per il processo  $p_3$  che consente a tutti i processi di completare la propria esecuzione. La risposta deve essere corredata da un diagramma di GANNT (teoricamente infinito) che mostri come i processi possano venir eseguiti nei tempi opportuni.

**Esercizio 3:** Se  $N$  e' il numero di matricola calcolare  $M=N\%5$ . Prendere l'oggetto  $M$  e descrivere:

- a) a cosa serve
- b) esempi reali di utilizzo

oggetto0: sottosistema di ambiente

oggetto1: autotools (autoconf, automake)

oggetto2: buddy list

oggetto3: linker dinamico

oggetto4: sistemi di controllo di versione distribuiti (ad esempio: cvs (concurrent version system), svn (subversion), git (fast version control system))



