

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010
 CONCORRENZA – 16 giugno 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: 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':

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

Scrivere un monitor che implementi un semaforo cartesiano planare sghembo.

Esercizio 2: Siano dati N processi P_0, P_1, \dots, P_{N-1} e un servizio di message passing asincrono.

Scrivere la funzione barrier(i) che implementi un servizio di sincronizzazione a barriera fra tutti i processi: ogni processo può proseguire oltre la funzione solo se tutti i processi hanno chiamato la barrier. Il parametro e' l'indice del processo.

Esercizio 3: Siano dati quattro processi di nome I, T, A, L cosi' strutturati

Process I: <pre>while (true) { ... print("I"); ... }</pre>	Process T: <pre>while (true) { ... print("T"); ... }</pre>	Process A: <pre>while (true) { ... print("A"); ... }</pre>	Process L: <pre>while (true) { ... print("L"); ... }</pre>
---	---	---	---

Completare il codice in corrispondenza dei punti di sospensione (...) per fare in modo che l'esecuzione concorrente stampi: ITALIAITALIAITALIA.... all'infinito.

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010
PARTE GENERALE – 16 giugno 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: Si consideri un array bidimensionale di interi allocato all'indirizzo 1024 in una macchina a 32 bit, con pagine di 256 byte e una memoria di 3 frame:

```
int A[16][16];
```

Si consideri il programma che segue (caricato all'indirizzo 256, il codice eseguibile e' di 48 byte):

```
int i;  
int j;  
for (i=0; i<16; i++)  
    for (j=0; j<16; j<16)  
        A[i,j]=i*j;
```

a) Mostrare quale sia il numero minimo di page fault che questo codice può generare in un sistema a memoria virtuale.

b) Considerare ora il codice:

```
int i;  
int j;  
for (j=0; j<16; j++)  
    for (i=0; i<16; i<16)  
        A[i,j]=i*j;
```

Qual e' in questo caso il numero minimo di page fault per l'esecuzione di questo codice?

Esercizio 2: Considerare i metodi di allocazione contigua, concatenata e indicizzata (a un livello) per file system.

Sia dato un file che occupi N blocchi. Mostrare il numero minimo e massimo di accessi (specificando quanti in lettura e quanti scrittura) per aggiungere un blocco al file (in posizione casuale). Ovviamente le risposte devono essere opportunamente motivate.

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: buddy list del kernel Linux

oggetto1: linker dinamico

oggetto2: sistemi di controllo di versione distribuiti (ad esempio: cvs (concurrent version system), svn (subversion), git (fast version control system))

oggetto3: sottosistema di ambiente nei sistemi Windows

oggetto4: autotools (autoconf, automake)



