

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 PROVA SCRITTA DI SISTEMI OPERATIVI  
 ANNO ACCADEMICO 2011/2012

16 luglio 2012

**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 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 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:** Per lo scopo di questo esercizio si ipotizzi la presenza di infiniti processi rossi e infiniti processi blu.

Red: process	Blue: process
limcol.enter(RED)	limcol.enter(BLUE);
....sezione limcol....	....sezione limcol....
limcol.exit(RED)	limcol.exit(BLUE)

Nella sezione limcol possono entrare sia processi rossi sia processi blu, ma il numero dei processi di uno dei due colori deve sempre superiore o uguale al 75% del totale dei processi nella sezione limcol.

(per esempio: 1 rosso e 3 blu oppure 3 rossi e uno blu oppure 5 rossi e uno blu sono casi possibili, 3 rossi e 3 blu oppure 2 rossi e 5 blu no)

1- Scrivere il monitor limcol (che ha due procedure entry: enter e exit). I processi in attesa devono venir riattivati in modo FIFO.

2- Sono possibili casi di starvation?

**Esercizio c.2:** Siano date le seguenti 16 funzioni  $foo_n$  che hanno due parametri booleani passati per indirizzo.

$foo_n(x, y) : \langle x=bool_n(x, y), y=1 \rangle$  dove  $bool_n$  e' definita come segue:

x	y	bool <sub>0</sub>	bool <sub>1</sub>	bool <sub>2</sub>	bool <sub>3</sub>	bool <sub>4</sub>	bool <sub>5</sub>	bool <sub>6</sub>	bool <sub>7</sub>	bool <sub>8</sub>	bool <sub>9</sub>	bool <sub>10</sub>	bool <sub>11</sub>	bool <sub>12</sub>	bool <sub>13</sub>	bool <sub>14</sub>	bool <sub>15</sub>
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Per quali indici n le funzioni foo possono essere utilizzate al posto della test&set?

**Esercizio g.1:** Scrivere una stringa di riferimenti infinita per la quale gli algoritmi MIN, LRU e FIFO si comportino esattamente allo stesso modo se applicati ad una memoria di 3 frame. La stringa deve fare riferimento a un numero finito di pagine e deve generare un numero infinito di page fault.

**Esercizio g.2:** Rispondere ad almeno due delle seguenti domande:

- A cosa serve l'algoritmo di Dekker e come funziona?
- Access Control List/Capability, a cosa servono? Quali sono le differenze?
- Cosa significa "partizionare" un disco? A cosa serve?