Nome_____ Cognome _____N. di matricola (10 cifre) _____

Riga ___ Col _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2006/2007
CONCORRENZA - 12 Luglio 2007

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1: Il gioco dell'oca. N giocatori giocano al gioco dell'oca eseguendo il seguente codice:

```
Pi [i=0,..,N] {
  int fine_partita;
  do { fine_partita = oca.muovi(i,random(1,6)) } while (!fine_partita);
  if (fine_partita == 1) exit(VINTO);  else exit(PERSO);
}
```

Il gioco procede per turni. Tutti i giocatori iniziano dalla casella 0. Il tabellone contiene 99 caselle.

A ogni turno il giocatore che invoca oca.muovi(i,n) muove di n caselle per poi passare la mano al giocatore successivo (che e' P(i+1)%2 a meno che quest'ultimo non stia saltando il turno; in tal caso si passa al primo giocatore successivo che non salta il turno). Se la partita non ha avuto termine, la p.e. restituisce 0. Inizia il giocatore P0.

Se un giocatore finisce su una casella multipla di 13 salta tutti i turni seguenti (ovvero **rimane sospeso su una condition all'intero del monitor**) fino a quando un altro giocatore non lo libera finendo sulla stessa casella e rimanendo intrappolato al suo posto.

Il primo giocatore che arriva o supera la casella 100 ha vinto e scatena la fase di terminazione della partita. In tale fase tutti i processi sospesi nel monitor vengono risvegliati e lasciano la p.e. ritornando 2. Ogni successiva invocazione della p.e. ritornera' immediatamente 2 senza altri effetti. Il valore di ritorno della p.e. per il processo vincitore e' invece 1.

a) Implementare il monitor oca, basandosi sull'usuale politica signal-urgent.

b) Discutere perche' mai, per risolvere questo particolare problema, la politica signal&continue sarebbe stata preferibile.

Esercizio 2: Siano dati i seguenti processi:

P1: S1.P(); S2.P(); S1.V();

P2: S1.V(); S1.P();

P3: S1.P(); S2.V(); S1.V();

Discutere tutte le possibili storie esecutive al variare dei valori iniziali assegnati ai semafori S1 e S2.

Esercizio 3: Message passing.

adoublesend(msg,dst1,dst2)

msg=adoublerecv(sender)

adoublesend spedisce il messaggio in modo asincrono a dst1 e dst2 ma dst2 deve riceverlo solo dopo che dst1 ha ricevuto il messaggio. Implementare il servizio di messaggi a salto doppio basandosi su un servizio di message passing asincrono (asend arecv). Non occorre alcun processo gestore.

Nome_____ Cognome _____N. di matricola (10 cifre) _____

Riga ___ Col _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2006/2007
PARTE GENERALE - 12 Luglio 2007

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1: Un sistema operativo gestisce un file system basato su i-nodes mantenendo in memoria principale la tabella degli i-node (precaricata al boot). L'i-node 0 descrive la directory radice. La dimensione di un blocco fisico e' di 1KB. La tabella degli i-node corrente e':

	inode[0]	inode[1]	inode[2]	inode[3]	inode[4]	inode[8]	inode[20]
size:	210	90	40	11	45	1094	2048
type:	d	d	d	l	d	f	f
...							
direct[0]:	3	20	6	31	8	14	15
direct[1]:	0	0	0	0	0	16	19
direct[2]:	0	0	0	0	0	0	0
...							
indirect[0]:	0	0	0	0	0	0	0

Il contenuto corrente del disco e':

block[3]	block[6]	block[8]	block[14]	block[15]	block[16]	block[19]	block[20]	block[31]
. 0	. 2	. 4	0110010	bg=red	0110010	style=bold	. 1	/local/bin
.. 0	.. 0	.. 2	1001000	fg=blue	0110010	font=serif	.. 0	
bin 3	bin 4	a.out 8	...	...	11001	...	conf 20	
local 2								
etc 1								

Mostrare il comportamento della testina del disco supponendo: 1) che venga utilizzato l'algoritmo dell'ascensore; 2) che inizialmente la testina sia sul blocco 5; 3) che all'istante 0 vengano eseguiti concorrentemente i seguenti due processi; 4) che il tempo di CPU per i processi sia istantaneo.

P1() { fd = open("/etc/conf","r"); read(fd,buffer,1000); close(fd); }

P2() { fd = open("/bin/a.out","w"); write(fd,buffer,1800); close(fd); }

Esercizio 2: Sia dato il seguente stato di un banchiere multivaluta, ove x,y,z,w sono variabili naturali.

COH (3,4+x)

	massime	assegnate	da richiedere
P1:	(6,5)	(0,3+x)	(6,y)
P2:	(5,10)	(3,5)	(2,5)
P3:	(7,7)	(1,5)	(6+y,2)
P4:	(7+x+z,8+w)	(x+z,z)	(7,8+z)

Sotto l'ipotesi che tale stato sia safe, calcolare il range di valori possibili per il capitale iniziale del banchiere, in funzione di x,y,z,w.

Esercizio 3: Sia x l'ultima e y la penultima cifra del vostro numero di matricola. Elencare e descrivere (separatamente) tutti gli interrupt e le trap gestite dal componente (y*10+x)%4.

0. Memory manager
1. Disk manager
2. File system manager
3. Scheduler