Nome/cognome ___________________________ N. di matricola (10 cifre) _______________ Posizione: Riga ___ Col ___

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2006/2007
CONCORRENZA - 16 Gennaio 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: In un noto supermercato bisogna prendere il numero per essere serviti al banco del pesce o a quello degli affettati. I clienti prendono i numeri dei banchi a cui sono interessati e si mettono in attesa. Alla chiamata del loro numero fanno la loro richiesta al banco e vengono serviti. Se nel frattempo vengono chiamati anche all'altro banco perdono il turno e debbono prendere un altro numero per il banco corrispondente.

Scrivere il monitor **Supermercato** con i metodi

```
prendi_numero(int banco);   /* prende un numero */
int attendi();              /* ritorna -1 se il cliente non ha piu' numeri validi;
                                 altrimenti ritorna il banco disponibile;
                                 deve essere invocata una volta per ogni numero preso */
servi_il_prossimo();        /* invocata dal commesso per gestire il cliente successivo */
```

sapendo che la vita dell'operatore e dei clienti consiste in

```
commesso() {while(1) { servi_il_prossimo(); <<gestisce la richiesta>> } }

cliente(int serve_pesce, int serve_pane) {
  if(serve_pesce) prendi_numero(PESCE);
  if(serve_pane) prendi_numero(PANE);
  while (serve_pesce || serve_pane) {
    int banco = attendi();
    if (banco == PESCE) { <<prendi il pesce>>; serve_pesce = 0; }
    else if (banco == PANE) { << prendi il pane>>; serve_pane = 0; }
    else if (banco == -1) {
      if(serve_pesce) prendi_numero(PESCE);
      if(serve_pane) prendi_numero(PANE);
    }
  }
```

Esercizio 2:

```
sem t[2] = {0,1};
sem p[2] = {3,3};
int g = 0;
player[i=0,1]() {
   while(1) {
     P(t[i]);
     if (random(3) == g) {
         V(t[(i+1)%2);
     } else {
         V(p[(i+1)%2];
         P(p[i]);
         V(t[i]);
     }
  }
}
```

Due giocatori player[0] e player[1] giocano secondo le regole qui a fianco.
La partita termina se il sistema entra in deadlock. Il perdente e' il processo che effettua l'ultima P() che determina il deadlock.
Sfortunatamente per il vincitore, il SO interviene e uccide entrambe i contendenti.

a) qual'e' il grado di parallelismo del gioco?
b) descrivere con l'ausilio di un grafo diretto tutte le possibili evoluzioni di una partita
c) puo' una partita durare in eterno?

Esercizio 3: Al fine di gestire le vostre uscite con gli amici il sabato sera volete implementate le seguenti primitive sincrone di coordinazione:

```
int propose(msg m, id ID);  /* propone la proposta m a ID; ritorna 1 se la proposta e' stata accettata
                                 e 0 se e' stata rifiutata */
msg accept(int n);          /* attende la ricezione di n proposte identiche da parte di n processi.
                                 Quando questo evento si verifica le proposte “vincitrici” vengono
                                 accettate e tutte le altre rifiutate */
```

Mostrare tre differenti implementazioni utilizzando primitive di message passing sincrone, asincrone e totalmente asincrone. In cosa differiscono le tre implementazioni? Ordinarle da quella piu' soddisfacente a quella meno soddisfacente (motivando le risposte) nei seguenti due casi a) processi in esecuzione sulla stessa macchina; b) processi remoti.

Nome/cognome ________________________ N. di matricola (10 cifre) __________________ Posizione: Riga __ Col___

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2006/2007
PARTE GENERALE - 16 Gennaio 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:

a) Mostrare un grafo di Holt che non contenga un knot e che sia riducibile in un passo a un grafo che contiene un knot di 4 nodi. Cosa possiamo dire sul deadlock in un grafo del genere?

b) Mostrare lo stato dell'algoritmo del banchiere che rappresenta il grafo di Holt trovato al punto a. E' uno stato safe?

Esercizio 2: 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	15	14
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	Nel mezzo	0110010	0110010	Noi leggiav	. 1	/local/tmp
.. 0	.. 0	.. 2	del cammin	1001000	0110010	amo un gio	.. 0	
tmp 3	tmp 4	A 8	di nostra	...	11001	rno per dile	conf 20	
local 2			vita			tto		
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 0; 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(“/tmp/A”,”a”);    write(fd,buffer,800);  close(fd); }
P2() { fd = open(“/etc/conf”,”r”); read(fd,buffer,2000); close(fd); }
```

Esercizio 3: Sia x l'ultima cifra del numero di matrica e y la penultima cifra. Descrivere il componente del sistema operativo numero (y*10+x)%4, senza dimenticarsi di elencare gli interrupt e le trap di pertinenza del componente e le interazioni dirette del componente con altri componenti e con parti hardware della macchina.

0) Memory managr
1) Disk manager
2) File system manager
3) Scheduler