

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010  
CONCORRENZA – 08 settembre 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:** Si costruisca il monitor 'dualbb' che gestisca due bounded buffer. Le funzioni entry da implementare sono:

```
enum buf {0,1};  
void put(buf t, T item);  
T get(buf t);  
dove T e' un tipo generico.
```

Entrambi i buffer hanno la capacita' massima di MAX elementi. I due buffer non sono indipendenti. La *put* inserisce l'elemento nel buffer indicato come parametro (0 o 1). La *get* preleva un elemento dal buffer indicato se il buffer non e' vuoto. Se il buffer e' vuoto e ci sono elementi nell'altro buffer viene prelevato un elemento dall'altro buffer, se infine entrambi i buffer sono vuoti *get* attende il primo elemento inserito nel solo buffer corrispondente al parametro.

**Esercizio 2:** Sia dato un servizio di message passing sincrono con timeout in ricezione cosi' definito:

`void srx_send(pid_t dest, Msg msg)` si comporta come la `send` sincrona

`Msg srx_rcv(pid_t sender, time timeout)` riceve un messaggio dal processo indicato come sender (che in questo servizio non puo' essere \*) ma se non arriva alcun messaggio, allo scadere del timeout indicato la `srx_rcv` restituisce NULL.

Usando il servizio cosi' definito scrivere (senza processi server) un servizio di message passing sincrono con timeout in trasmissione che abbia le seguenti due funzioni:

`void stx_send(pid_t dest, Msg msg, time timeout)` si comporta come la `send` sincrona, ma se il ricevente non richiede di poter ricevere un messaggio, allo scadere del timeout la `stx_send` ritorna.

`Msg stx_rcv(pid_t sender)` riceve un messaggio dal processo indicato come sender (che in questo servizio non puo' essere \*), la `stx_rcv` non ha alcun timeout: attende fino a che non giunga un messaggio trasmesso tramite la `stx_send`.

**Esercizio 3:** Sia dato un programma composto da due processi concorrenti P[0] e P[1].

```
semaphore S[2] init {1,0} /* S[0] e' inizializzato a 1, S[1] a 0 */  
int val[2]={0,1};
```

```
process P[2]:  
  S[i].P();  
  print val[i];  
  val[i] += val[1-i];  
  S[1-i].V();
```

Quale e' la sequenza stampata dal programma? E' una sequenza finita?

NOTA: tutti gli esercizi verranno valutati solo se le risposte saranno corredate da motivazioni e dimostrazioni scritte in Italiano o in Inglese (corretto o almeno comprensibile). La presenza nella soluzione di un esercizio di solo codice sorgente/tabelle/scarabocchi/simboli vari comporta la non valutazione dell'esercizio.



UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2009/2010  
 PARTE GENERALE – 08 settembre 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 identità valido. La consultazione o anche solo la disponibilità di altro materiale comporterà l'annullamento del compito (con la stessa penalizzazione di punteggio della grave insufficienza per la prossima esercitazione scritta).

**Esercizio 1:** Si consideri il seguente bubble sort:

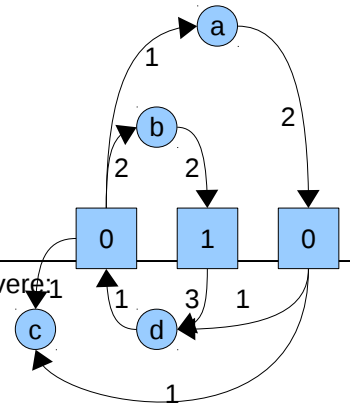
```
struct page data[5]={ /* dati iniziali */ };
struct page tmp;
int i,j;
for (i=4; i>0; i--)
    for (j=0;j<i;j++)
        if (data[j] > data[j+1]) {
            tmp=data[j]; data[j]=data[j+1]; data[j+1]=tmp;
        }
}
```

Considerato che il codice e' caricato nella pagina 10, che sizeof(struct page)=2k (l'ampiezza di una pagina) e che i dati (tutti globali) sono caricati a partire dalla pagina 20.

- i) dare una rappresentazione delle possibili stringhe di riferimento relative all'esecuzione del bubble sort
- ii) qual e' il numero massimo e il numero minimo di page fault che genera il bubble sort quando viene eseguito su un sistema teorico con 5 frame di memoria che usa l'algoritmo MIN?

**Esercizio 2:** Considerare il grafo di Holt qui a lato:

- i) e' completamente riducibile?
- ii) Scrivere lo stato di un algoritmo del Banchiere che rappresenti esattamente la situazione del grafo di holt.



- Esercizio 3:** Se N e' il numero di matricola calcolare  $M=N\%5$ . Prendere l'oggetto M e descrivere:
- a) cosa e', a cosa serve
  - b) esempi di funzionamento.

- oggetto0: Memory Manager
- oggetto1: Interrupt Handler
- oggetto2: Scheduler
- oggetto3: Disk Cache
- oggetto4: File System Manager

NOTA: tutti gli esercizi verranno valutati solo se le risposte saranno corredate da motivazioni e dimostrazioni scritte in Italiano o in Inglese (corretto o almeno comprensibile). La presenza nella soluzione di un esercizio di solo codice sorgente/tabelle/scarabocchi/simboli vari comporta la non valutazione dell'esercizio.



