

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 PROVA SCRITTA DI SISTEMI OPERATIVI  
 ANNO ACCADEMICO 2013/2014  
 20 gennaio 2014

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 vogliono 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:** Scrivere il monitor lwlrb. Il monitor deve implementare le seguenti procedure entry:

```
void write(generic_type val);
generic_type read(void);
```

Il lwlrb si comporta come un bounded buffer di MAX elementi che coordina l'attività di numerosi processi produttori/scrittori e numerosi lettori/consumatori. lwlrb ammette un numero massimo (sempre MAX) di lettori e scrittori in attesa.

Se il buffer e' vuoto e ci sono piu' gia' MAX lettori in attesa, il lettore che e' in attesa da piu' tempo esce resituendo NULL.

In ugual modo se il buffer e' completamente pieno e ci sono gia' MAX scrittori che attendono di scrivere viene perduto il valore che da piu' tempo nel buffer attende di venir letto, il primo processo in attesa di scrivere puo' cosi' scrivere il suo elemento nel buffer e sbloccarsi.

**Esercizio c.2:**

a) Si consideri la funzione atomica dualshift(a,b) che opera su due operandi di tipo byte passati per indirizzo. L'operazione fa lo shift a destra dei due operandi. Il bit piu' significativo di a viene posto a zero, il bit piu' significativo di b diviene quello che all'atto della attivazione di dualshift era il bit meno significativo di a.

es. se a vale 6 e b vale 4 dopo la chiamata di dualshift(a,b) a vale 3 e b 2. Se a vale 5 e b 6 dopo la chiamata dualshift (a,b) a vale 2 e b vale 131 (128+3).

La funzione dualshift puo' essere usata al posto della test&set per la sincronizzazione fra processi? Dimostrare la risposta.

b) Si consideri ora la funzione andor(a,b) che opera su due parametri di tipo booleano passati per indirizzo e cosi' definita:

```
andor(a,b)=<c=a or b; b=a and b; a=c>
```

Puo' la funzione andor essere usata al posto della test&set per la sincronizzazione fra processi? Dimostrare la risposta.

Si ricorda che le operazioni di assegnazione di valori costanti a variabili vengono considerati atomici.

**Esercizio g.1:**

a) Sia data la seguente stringa di riferimenti: 123451234123121.

mostrare il comportamento degli algoritmi MIN e LRU quando operano su una memoria di 3 frame.

b) Data una memoria di 4 frame contenente la pagina 4 nel frame 1, la pagina 3 nel frame 2, la pagina 2 nel frame 3 e infine la pagina 1 nel frame 4. Mostrare una stringa di riferimenti di un programma che usi 5 pagine (esiste la pagina 5 non ancora mappata in memoria oltre alle 4 cariate nei frame) e che consenta alla fine dell'esecuzione di avere tutte le pagine nel frame di indice corrispondente. La pagina 1 nel frame 1, la pagina 2 nel frame 2 e cosi' via.

**Esercizio g.2:**

a) Se un sistema ha una RAM molto ampia puo' non essere utile usare la memoria virtuale. In questo caso ha senso egualmente usare la paginazione di memoria? Perche'?

b) Perche' i metodi di sincronizzazione tipo test&set (detti anche spinlock) assumono grande rilevanza nella scrittura di sistemi operativi multiprocessore?

c) Confrontare gli algoritmi di scheduling round robin e a priorita' statica. Indicare in quali casi sono da usare algoritmi di tipo round robin e quando quelli a priorita' statica.