

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
PROVA SCRITTA DI SISTEMI OPERATIVI  
ANNO ACCADEMICO 2012/2013  
21 febbraio 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 `bbwl` che fornisca il servizio di bounded buffer with logging. Ci sono tre tipi di processi che interagiscono con questo servizio: i produttori, i consumatori e i logger. I produttori, quando hanno generato un elemento lo inseriscono nel buffer chiamando la procedure `entry`:

```
bbwl.write(etype elem)
```

e quando i consumatori possono elaborare un ulteriore elemento chiamano la funzione (`entry`):

```
etype bbwl.read()
```

il buffer ha una capacità massima limitata a `MAXELEM` elementi.

I processi logger chiamano la funzione:

```
etype bbwl.log()
```

per poter registrare una traccia dei pacchetti transitati nel buffer. I pacchetti devono essere sempre tracciati (forniti a un logger) prima di poter essere ricevuti da un consumatore e i pacchetti devono essere forniti ai processi logger e ai processi consumatori in ordine FIFO (nell'ordine col quale i processi produttori li hanno inseriti nel buffer chiamando la funzione `write`).

**Esercizio c.2:** Considerare le seguenti funzioni atomiche (le variabili sono passate per indirizzo)

```
alpha(x,y): <x=4, y=sqrt(x)>
```

```
bravo(x,y): <y=sqrt(x), x=4>
```

```
charlie(x,y): <y=sqrt(x), x=4*y>
```

```
delta(z,t): <z=z xor t, t=z xor t>
```

Quali fra queste funzioni possono essere usate per implementare un servizio di sezioni critiche (in modo simile a quanto visto a lezione con la `Test&Set`)? fornire il codice di una implementazione o una dimostrazione di impossibilità.

**Esercizio g.1:** Sia dato l'algoritmo di rimpiazzamento `MINNUM`. Come pagina da rimpiazzare `MINNUM` sceglie sempre quella con l'indirizzo logico più basso (numero di pagina minore).

a. mostrare una stringa di riferimenti di lunghezza infinita e che generi infiniti page fault tale che `MIN` e `MINNUM` si comportino esattamente nello stesso modo

b. mostrare una stringa di riferimenti di lunghezza infinita tale che `MINNUM` generi un page fault ad ogni accesso in memoria mentre `MIN` generi un numero finito di page fault (in entrambi i punti l'insieme delle pagine utilizzate nelle stringhe di riferimenti deve essere finito)

**Esercizio g.2:** Rispondere alle seguenti domande:

1. La paginazione non elimina completamente né la frammentazione interna né quella esterna ma le rende trascurabili. Perché?
2. Il file system `ext2` è più efficiente nell'accesso diretto a file di piccole dimensioni rispetto a quelli di grandi dimensioni. Perché?
3. In quali casi l'algoritmo `C-Look` è preferibile all'algoritmo `Look`? Perché?
4. Nella memorizzazione delle capability per il controllo di accesso ai file può essere usato un algoritmo di crittografia a singola chiave (chiave privata). Perché?