nome e cognome _____ numero di matricola 16 74 _____

CORSO DI SISTEMI OPERATIVI
CORSO DI LAUREA IN INFORMATICA - UNIVERSITA' DI BOLOGNA
PRIMO APPELLO DELLA SESSIONE AUTUNNALE AA 1999/2000
22 settembre 2000

Renzo Davoli

Esercizio -1. Essersi iscritti correttamente per sostenere questa prova.
Esercizio 0. Scrivere correttamente il proprio nome, cognome e numero di matricola in tutti i fogli.

Un database è composto da N file.

I programmi clienti, quando devono effettuare un aggiornamento, devono accedere in modo esclusivo a un sottoinsieme S degli N file (il sottoinsieme viene specificato al momento della richiesta).

Il tipico programma-cliente ha la seguente struttura algoritmica:

```
DBclient:
      while true do
      {
            ... compute or ask the operation to do and the set S...
            DBmanager.lock(S)
            ... DB update operations ...
            DBmanager.unlock(S)
      }
```

Ogni file deve essere acceduto al più da un processo e quindi se all'atto dell'operazione lock uno dei file di S è stato richiesto e non ancora rilasciato da un altro processo, la chiamata lock deve essere bloccante. Al contrario, due processi devono essere in grado di accedere al DB senza attesa se i sottoinsiemi specificati sono ad intersezione vuota.

Esercizio 1. Scrivere il monitor DBmanager.

nome e cognome _____ numero di matricola 16 74 _____

Esercizio 2. Scrivere le funzioni della classe DBclass (della quale DBmanager è un'istanza) facendo uso di semafori.

Esercizio 3a. Date le specifiche del problema indicate prima del testo dell'esercizio 1 sono possibili casi di deadlock?

Esercizio 3b. Sono possibili casi di starvation?

Esercizio 4. Un sistema ha la seguente istruzione atomica:

```
int incr(int &x)
{
      return(x++);
}
```

Con questa istruzione viene tentata l'implementazione di un meccanismo per realizzare le sezioni critiche come segue:

```
class mutex {
private:
      int counter;
      int sharedval;

public:
      mutex() {counter=sharedval=0;}

      enter()
      {
            int x;
            x=incr(counter);
            while (x > sharedval)
                  ;
      }

      leave()
      {
            incr(sharedval);
      }

}
```

4a. La classe mutex realizza la mutua esclusione?

4b. Se alla 4a avete risposto sì: la soluzione è fair?
Se alla 4a avete risposto no: come è possibile modificare le funzioni enter e exit per realizzare un meccanismo di supporto per la mutua escluzione?