

UNIVERSITA' DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2004/2005
COMPITO CONCORRENZA – 10 Giugno 2005

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.

Esercizio 0: Su entrambi i fogli, scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

Esercizio 1.a

Si considerino i cinque processi seguenti. Tutti semafori sono tutti inizializzati a zero.

processo 1:	processo 2:	processo 3:	processo 4:	processo 5:
s2.V()	s1.V()	s1.V()	s3.V()	s2.V()
s3.V()	s4.V()	s5.V()	s5.V()	s4.V()
s1.P()	s2.P()	s3.P()	s4.P()	s5.P()
s1.P()	s2.P()	s3.P()	s4.P()	s5.P()
....

Descrivere il comportamento di questo codice. Esistono processi che si bloccano? Esistono problemi di deadlock?

Esercizio 1.b

Si considerino i cinque processi seguenti. Tutti semafori sono tutti inizializzati a zero.

processo 1:	processo 2:	processo 3:	processo 4:	processo 5:
s2.V()	s1.V()	s1.V()	s3.V()	s2.V()
s1.P()	s2.P()	s3.P()	s4.P()	s5.P()
s3.V()	s4.V()	s5.V()	s5.V()	s4.V()
s1.P()	s2.P()	s3.P()	s4.P()	s5.P()
....

Descrivere il comportamento di questo codice. Esistono processi che si bloccano? Esistono problemi di deadlock?

Esercizio 2

Si costruisca un buffer limitato "colorato" con le seguenti chiamate:

```
void buf.insert(T elem, Color c)
T buf.extract()
bool buf.empty()
```

Il buffer è limitato con dimensione N; la chiamata extract è bloccante se il buffer è vuoto; la chiamata insert è bloccante se il buffer è pieno. Inoltre, supponendo che esistono due colori rosso e blu, la seguente condizione deve essere sempre valida: il numero di elementi di un colore non deve superare il doppio del numero di elementi dell'altro colore. Una chiamata può diventare bloccante per evitare che tale condizione sia non bloccante.

Esercizio 3

Facciamo un semaforo (stradale) con i semafori (di SO). Si consideri un incrocio con N strade (quelli che si ritrovano solo nei quiz della patente). Ad ogni strada è posto un semaforo, descritto dal codice seguente:

```
TrafficLightA[j=1,..N]:
  Accendi(Rosso)
  while (1) {
    xxxAttendi(i)
    Accendi(Verde)
    sleep(righttime(i));
    Accendi(Giallo)
    sleep(10);
    Accendi(Rosso)
    xxxOK(i);
  }
```

- 1) Una descrizione del genere è volutamente generica. E' necessario (da parte vostra) descrivere le proprietà di safety e liveness che vi aspettate da un incrocio del genere.
- 2) Scrivere le funzioni/macro **xxxAttendi** e **xxxOK** facendo uso di semafori generali, in modo tale da rispettare le proprietà di safety e liveness descritte da voi.