

UNIVERSITÀ DEGLI STUDI DI BOLOGNA – CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI
III APPELLO (STRAORDINARIO) AUTUNNALE – ANNO ACCADEMICO 2000/2001
20 ottobre 2001

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

Esercizio 0: Scrivere correttamente il proprio nome, cognome e numero di matricola in ogni foglio prima di svolgere ogni altro esercizio seguente.

Esercizio 1: Un semaforo ternario è un semaforo che soddisfa il seguente invariante $0 \leq n_{tV} + t_{init} - n_{tP} \leq 2$, dove n_{tV} è il numero di operazioni tV completate sul semaforo e n_{tP} è il numero di operazioni tP . Scrivere un monitor che implementi le funzionalità di t_{init} , tP , tV su un semaforo ternario.

Esercizio 2: Dato un servizio di semafori generali che fornisca le chiamate *init*, *P*, *V* implementare i semafori ternari come definiti nell'esercizio 1.

Esercizio 3: In un servizio di message passing completamente asincrono nè l'operazione send nè l'operazione receive sono bloccanti. Se non vi sono messaggi in coda al quando un processo chiede di fare la receive, viene restituito un messaggio NULL ma la funzione comunque immediatamente ritorna il controllo al chiamante. Sia dato un servizio di message passing asincrono operante con le primitive *asend* e *arecv*, implementare un servizio di message passing completamente asincrono implementando le chiamate *casend* e *carecv* facendo uso di un processo gestore. (per semplicità si pu' implementare per una singola coda di messaggi)

Esercizio 4a. Cosa fa il seguente codice? Descriverne il comportamento.

```
bool b1=false;b2=false;
cond c1,c2;

procedure entry p1():
    if (b1) c1.wait;
    c1.signal;
    b1=false; b2=true;
    c2.signal;
end;

procedure entry p2():
    if (b2) c2.wait;
    c2.signal;
    b2=false; b1=true;
    c1.signal;
end;
```

Esercizio 4b. Il codice dell'esercizio 4a può causare situazioni di *deadlock*?