

UNIVERSITÀ DEGLI STUDI DI BOLOGNA – CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI
 PRIMA PROVA PARZIALE – ANNO ACCADEMICO 2002/2003
 29 ottobre 2002

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: Sia dato un sistema dove oltre alle azioni atomiche di lettura e scrittura in memoria sia data la seguente operazione atomica che opera su due variabili di tipo intero:

$f(a, b) = \langle a = b = (b * 3) \% 11 \rangle$

Scrivere, se possibile, un meccanismo di supporto di sezioni critiche in modo simile al test&set usando la funzione f; in caso contrario fornire una spiegazione del perché non sia possibile scrivere tale supporto con la funzione f.

Esercizio 2: Il seguente programma concorrente, a seconda dell'interleaving fra i due processi, può stampare una o più stringhe. Indicare quali. (nota: tutti i semafori sono semafori generali inizializzati a 0)

```
P: process {
    print E;
    s2.V();
    s2.V();
    s1.P();
    print S;
}

Q: process {
    s2.P();
    print R;
    s2.P();
    s1.V();
    print P;
}
```

Esercizio 3: Considerate i seguenti processi:

```
P: process {
    print C;
    print R;
}

Q: process {
    print A;
    print P;
    print E;
}
```

Modificare il codice dei processi in modo che stampino unicamente le parole "CARPE" e "CAPRE". Le uniche modifiche ammesse sono inserimenti di invocazioni di primitive P e V sui semafori. Tutti i semafori utilizzati devono essere inizializzati a zero.

Esercizio 4: Bus Eireann è il nome del servizio di trasporti su autobus irlandese. La stazione degli autobus di Dublino offre una serie di viaggi organizzati per le mete turistiche più visitate dell'Irlanda. Gli utenti che vogliono partecipare ad un viaggio hanno già acquistato un biglietto per la meta desiderata. Esistono *NMETE* mete diverse.

Il giorno prescelto, gli utenti si recano all'autostazione, che è composta da una sala d'attesa e da un'unica piattaforma di partenza. Gli utenti entrano nella sala d'attesa e raggiungono il gruppo di partecipanti ad una delle gite. Quando il gruppo ha raggiunto la dimensione prevista (che può essere ottenuta tramite la funzione `dimgruppo(int i)`, dove `i` è l'indice

del gruppo), il gruppo esce sulla piattaforma di partenza e aspetta l'autobus. Quando l'autobus arriva, tutti i partecipanti salgono nell'autobus, che puo' finalmente partire.

La vita di un utente e di un autobus sono quindi rispettivamente:

```
utente: process {
    eireann.entraSala(int gruppo);
    eireann.esciPiattaforma (int gruppo);
    eireann.saliAutobus(int gruppo);
}

autobus: process {
    eireann.vaiPiattaforma(int gruppo);
    eireann.iniziaGita(int gruppo);
}
```

Il fatto che esista un'unica piattaforma impone alcuni accorgimenti. Quando un gruppo e' formato, tutti i partecipanti devono uscire uno dopo l'altro, uno alla volta. Non e' possibile mischiare due gruppi. Quando il primo partecipante esce, l'autobus assegnato puo' uscire dal deposito e recarsi alla piattaforma. I partecipanti non possono salire sull'autobus fino a quando non arriva alla piattaforma. Quando tutti i partecipanti sono saliti, l'autobus puo' partire e un nuovo gruppo puo' utilizzare la piattaforma. Quando un intero gruppo e' uscito sulla piattaforma, un nuovo gruppo con lo stesso indice puo' formarsi nella sala d'attesa.

- A) La definizione data non e' starvation-free. Definire un invariante che garantisca assenza di starvation.
- B) Data la definizione del problema, possibile che vi siano deadlock?
- C) Implementare il monitor eireann