

SISTEMI OPERATIVI IDT/IEL
Seconda prova intermedia, 19 giugno 2007

Nome _____ Cognome _____

Preferenza per l'orale

29 giugno 2007 (appello regolare)

4 luglio 2007 (data riservata)

11 luglio 2007 (data riservata)

13 luglio 2007 (appello regolare)

1 agosto 2007 (data riservata)

Esercizio 1

Sia dato un sistema che prevede risorse di tipo R_A , R_B , R_C , e R_D , delle quali sono presenti rispettivamente 4, 3, 1, e 1 istanze. Sul sistema si trovano in esecuzione 3 processi (P_1 , P_2 , e P_3), le cui richieste massime possibili per le suddette risorse sono indicate nella tabella sottostante:

	R_A	R_B	R_C	R_D
P_1	3	3	1	1
P_2	3	2	1	0
P_3	1	1	0	1

All'istante t_0 alcune istanze delle suddette risorse sono allocate ai processi, come indicato di seguito:

	R_A	R_B	R_C	R_D
P_1	2	2	0	0
P_2	1	0	0	0
P_3	0	1	0	1

- Si determini se il sistema si trova in uno stato sicuro, ed eventualmente si individui una sequenza sicura;
- analogamente, si verifichi se il sistema si trova in uno stato sicuro nell'ipotesi che sul sistema siano presenti 2 istanze (anziché 1) della risorsa R_C .

Esercizio 2

Si definisca una classe Java denominata `Generatore` che, dato il riferimento ad un oggetto di classe `CodaIllimitata` descritto meglio nel seguito, ed in un flusso di esecuzione dedicato, inserisca periodicamente (periodo **T** secondi) nella coda un numero compreso tra **A** e **B**.

Si definisca quindi una seconda classe Java denominata `Moltiplicatore` che, dati i riferimenti a 2 oggetti di classe `CodaIllimitata`, sempre in un flusso di esecuzione dedicato, calcoli il prodotto

- tra i primi due elementi della prima coda, se questa contiene già almeno due elementi
- tra il primo elemento della prima coda ed il primo elemento della seconda, altrimenti.

e quindi mostri il risultato sullo schermo. Si tenga presente che in nessun caso si dovrà calcolare il prodotto se non sono disponibili entrambi gli operandi.

Si realizzi quindi un programma Java che, dati due oggetti di classe `CodaIllimitata`, provveda tramite opportune istanze della classe `Generatore` a riempire il primo con tasso **T1** ed il secondo con tasso **T2**. Parallelamente dovrà essere eseguito il calcolo specificato nella classe `Moltiplicatore`. Il programma dovrà terminare dopo che sono stati calcolati **K** prodotti.

La classe `CodaIllimitata` è così definita:

```
public class CodaIllimitata {  
    public void deposita(double d);  
    public double preleva();  
    public int dimensione();  
}
```

Per la risoluzione di problematiche di sincronizzazione è consentito solo l'utilizzo del costrutto nativo Java `synchronized(...)` (tipo regione critica).