

SISTEMI OPERATIVI IDT/IEL - Prima prova intermedia, 3 giugno 2004

Esercizio 1

Sia dato un sistema nel quale a ciascun processo è assegnato, in maniera statica, uno di due possibili livelli di priorità ("ALTA" e "BASSA"). Per ogni livello di priorità è prevista una coda dei processi pronti. In particolare, la coda dei processi pronti ad alta priorità è gestita in modalità Round Robin (con quanto di tempo $q=2$), mentre quella dei processi pronti a bassa priorità è gestita secondo uno schema FCFS. Inoltre, per la gestione della priorità fra le due code è prevista la prelazione.

Supponendo di avere 4 processi P1, P2, P3 e P4, con i tempi di arrivo (T_A), i tempi di elaborazione richiesti per il loro completamento (T_{CPU}) e le priorità riportati nella tabella sottostante, si determinino i tempi di attesa (*waiting*) e di completamento (*turn around*) per i singoli processi.

Processo	T_A	T_{CPU}	Priorità
P1	0	6	BASSA
P2	1	3	BASSA
P3	4	5	ALTA
P4	5	3	ALTA

Esercizio 2

Si definisca una prima classe, denominata *Riempimento*, che realizzi un thread in grado di riempire con valori casuali un vettore di elementi di tipo `double`. Si definisca poi una seconda classe, denominata *Massimo*, che determini il massimo valore contenuto in un vettore di elementi di tipo `double`.

Utilizzando le due classi di cui sopra, si realizzi quindi un programma Java che istanzia un vettore di `double` di dimensione N (eventualmente passato da linea di comando), e che quindi provvede a riempirlo con valori casuali in un thread separato; una volta completato il riempimento, il programma determina il valor medio ed il valore massimo degli elementi del vettore: mentre la determinazione del valor medio avviene nel thread principale, quella del massimo avviene in un thread separato. Quando entrambi i valori sono stati determinati, il programma li stampa sullo schermo.

Si ricorda che, in Java, per ogni vettore è definito un attributo `length` di tipo `int` che indica il numero di elementi contenuto nel vettore.

Infine, per generare i valori casuali nella classe *Riempimento*, si suggerisce l'utilizzo del metodo `random()` della classe `Math`:

```
static double random(): resituisce un valore non negativo di tipo  
double, appartenente all'intervallo [0.0, 1.0)
```