

SISTEMI OPERATIVI IDI - Prima prova intermedia, 22 marzo 2005

Esercizio 1

In un sistema tempo-reale che adotta uno scheduling di tipo *rate-monotonic* (RMS) all'istante $t=0$ si presentano tre processi periodici, aventi le caratteristiche riportate in tabella:

Processo	T (periodo)	C (carico)
P1	4	1
P2	6	3
P3	8	0,5

Per il sistema di processi sopraindicato si determini:

- la rappresentazione grafica della traccia di esecuzione per $t \leq 21$
- se la traccia è periodica o meno e, in caso affermativo, il periodo;
- come cambia la traccia nel caso in cui il sistema comprenda anche i due processi descritti nella sottostante tabella

Processo	T (periodo)	C (carico)
P4	8	1
P5	30	2

Esercizio 2

Si realizzi un programma che:

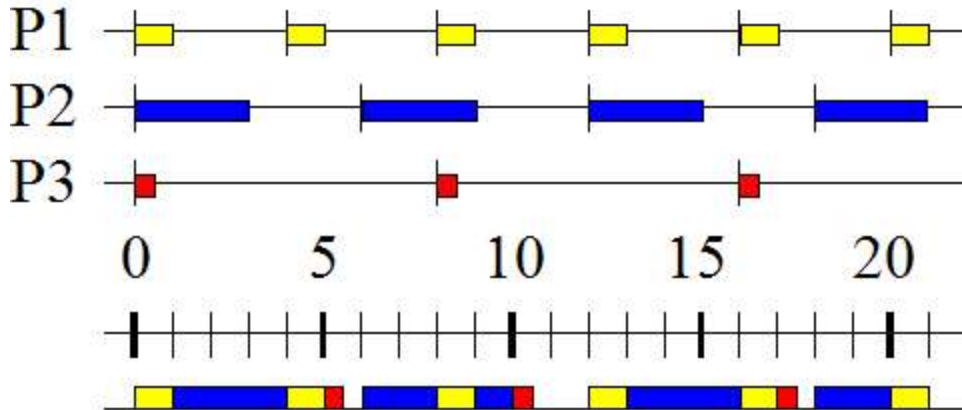
- provvede a riempire con valori casuali un vettore di elementi di tipo `double` ed avente lunghezza `N`. Per la determinazione dei valori casuali si usi il metodo statico `random()` della classe `Math`, che restituisce un valore non negativo di tipo `double`, nell'intervallo `[0.0, 1.0)`;
- una volta completato il riempimento del vettore di cui sopra avvia, in un thread separato, il calcolo della somma degli elementi del vettore;
- per tutta la durata del calcolo di cui al precedente punto, visualizza periodicamente (con periodo pari a `T ms`) la somma parziale fino a quel momento determinata;
- al termine visualizza la somma totale.

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.

soluzione

Esercizio 1

a)



b) il sistema di processi è periodico, di periodo 24

c) il sistema di processi P1, P2, P3, P4, P5 non è schedulabile

Esercizio 2

```
public class Somma extends Thread
{
    public Somma( double[] v ) { vettore = v; }
    public void run() {
        for ( int i = 0; i < vettore.length; i++ )
            somma += vettore[ i ];
        finito = true;
    }
    public double somma() { return somma; }
    public boolean finito() { return finito; }

    private double[] vettore = null;
    private double somma = 0;
    private boolean finito = false;
    public static final int N = 10000;
    public static final int T =100;

    public static void main( String[] args ) {
        double[] v = new double[ N ];
        for ( int i = 0; i < v.length; i++ )
            v[ i ] = Math.random();
        Somma s = new Somma( v );
        s.start();
        while ( ! s.finito() ) {
            try {
                Thread.sleep( T );
            } catch ( InterruptedException ie ) {}
            System.out.println( "somma parziale: " + s.somma() );
        }
        System.out.println( "somma totale: " + s.somma() );
    }
}
```