Programación Concurrente y de Tiempo Real
Semana Número 5-Hoja de Problemas

Resumen

Se relacionan a continuación el conjunto de ejercicios/problemas sobre los que el alumno deberá trabajar en la clase de problemas de la semana número cinco.

1. Enunciados

1. Escriba un "hola mundo concurrente ". Llámelo `holaMundo.java`.

2. Escriba un hilo que muestre pares o impares, según se indique en el constructor, un número dado de veces, que también se indicará en el constructor. Llame a la clase `ParImpar.java`. Escriba ahora un código que hago uso de la clase anterior. Llámelo `Usa_ParImpar.java`. Observe el entrelazado.

3. Ya ha desarrollado código para calcular de forma secuencial mediante un método de *Monte Carlo* aproximaciones a la integral definida de una función. Paralelice ahora su código para realizar el mismo cálculo utilizando varios *threads*. El programa debe leer desde la línea de comandos el números de hilos a utilizar y el número de puntos de prueba a emplear. Guarde su trabajo en `intMonteCarloParalelo.java`. Escriba también una tabla que compare los picos de uso de CPU (para el mismo número de puntos) entre la versión secuencial y la paralela (con diferente número de hilos). Guárdela en `tabla.pdf`.

4. Considere[1] el siguiente código. Indique qué salida produce, justificando su respuesta:

```
public class E31 extends Thread{
 private static Integer i = new Integer(1);
 private static Integer j = new Integer(1);
 private static int k=0;
 private E31 mi_thread;
 public E31(){k++;}
 public void run(){
   if(k<500)
     {mi_thread=new E31();
```

[1] Planteada una versión similar en el examen de teoría de Septiembre de 2013.

```
        i--;
        mi_thread.start();
        try{mi_thread.join();}catch (InterruptedException e){}
       }
  }

  public static void main(String[] args) throws Exception{
    E31 otro_thread = new E31();
    otro_thread.start();
    otro_thread.join();
    System.out.println(i.toString());
  }
}
```

5. Desarrolle ahora una versión paralela del programa para computar autómatas celulares unidimensionales que escribió en la segunda hoja de problemas. Llámelo `aCelularParalelo.java`. Desarrolle una tabla de picos de uso de CPU que compare las versiones secuencial y paralela para el mismo número de células y generaciones. Guárdela en `tabla2.java`.