Programación Concurrente y de Tiempo Real
Grado en Ingeniería Informática
Examen Final de Prácticas
Junio de 2021

1. Notas de Procedimiento

1. Dispone de 180 minutos para completar el ejercicio.
2. Puede utilizar el material bibliográfico (libros) y copia de API que estime convenientes. Se prohíbe el uso de apuntes, *pen-drives* y similares.
3. Todos los ficheros de código generados deben incluir un comentario que incluya nombre, apellidos y D.N.I.
4. Entregue sus productos, utilizando la tarea de entrega disponible en el Campus Virtual, en un fichero (`.rar, .zip`) de nombre

   `Apellido1_Apellido_2_Nombre`

   que contendrá una subcarpeta por cada enunciado del examen, la cual contendrá a su vez el conjunto de ficheros que den solución a ese enunciado en particular. Cada fichero de código debe incluir un comentario con su nombre, apellidos y D.N.I.
5. **No se aceptarán bajo ningún concepto entregas efectuadas por medios diferentes a la tarea de subida, ni fuera del plazo de examen establecido. Por tanto, debe asegurarse de entregar sus productos en el plazo establecido y mediante la tarea habilitada a tal efecto.**

2. Criterios de Corrección

Veáse la ficha de la asignatura, apartado Sistema de Evaluación, subapartado Criterios Generales de Evaluación.

3. Enunciados de Examen

1. (Procesamiento Paralelo de Array 1-D, 3.5 puntos) Se desea procesar la evolución de un array `r` de 10^6 componentes numéricos en el tiempo, a

partir de la siguiente ecuación de estado, donde $r^{(t)}[i]$ es el valor de la i-ésima componente del array en el instante t:

$$r^{(t+1)}[i] = (r^{(t)}[i-2] - r^{(t)}[i-1]) \cdot 10 + r^{(t)}[i+1]$$

Se pide escribir un programa en Java que efectúe de forma secuencial y paralela el cálculo indicado, a fin de cuantificar la mejora de rendimiento temporal, de acuerdo a las siguientes especificaciones:

- tareas paralelas soportadas por implementación de la interfaz `Runnable`.
- división automática del cálculo en tareas paralelas.
- los valores son números de tipo entero entre 0 y 5.
- condición de frontera a imponer: completar con ceros.
- procesamiento de las tareas mediante un ejecutor de clase `ThreadPoolExecutor`.
- el array original se carga con 10^6 números aleatorios en el rango indicado, y se procesa a lo largo de 200 instantes de tiempo discreto, para los siguientes modos de cálculo:
  - modo secuencial: se procesa el array secuencialmente (hebra única) mostrando el tiempo total de ejecución con precisión de nanosegundo.
  - modo paralelo: se procesa el array en paralelo (múltiples hebras) mostrando el tiempo total de ejecución, con precisión de nanosegundo y con división automática del dominio de datos. Una vez iniciado el programa, solicitará por teclado el número de hilos a emplear en la versión paralela, mostrará el tiempo empleado en el cálculo de la versión secuencial, a continuación el tiempo empleado en el cálculo de la versión paralela y, por último, también mostrará el speedup alcanzado.

2. (Cálculo Distribuido del Máximo, 3.5 puntos) Desarrolle una aplicación distribuida, empleando la biblioteca MPJ Express, que permita obtener de forma remota el mayor de los valores de una lista de números enteros. La aplicación constará de una única clase denominada Maximo y de solo dos procesos concurrentes. En proceso emisor generará de forma aleatoria una lista de cinco números enteros en el rango [0, 10) y la enviará al proceso receptor. El proceso receptor determinará el mayor de los valores de esa lista y lo remitirá al proceso emisor, que lo imprimirá como resultado por pantalla. Durante la ejecución de la aplicación han de imprimirse exactamente cuatro mensajes por pantalla (uno por línea) con el siguiente formato:

```
Emisor envía la lista: 5, 5, 6, 9, 3
Receptor recibe la lista: 5, 5, 6, 9, 3
Receptor devuelve: 9
Emisor recibe: 9
```

Los valores numéricos podrán variar, evidentemente, en cada ejecución. El primero y el último de los mensajes deberán imprimirse desde el proceso emisor. El segundo y el tercero desde el proceso receptor. Incluya en forma de comentario al comienzo del archivo las órdenes de compilación y ejecución. **Asegúrese de incluir como comentarios de su código las órdenes de línea de comando que propone para compilar y ejecutar el mismo**.

AYUDA: Puede emplear la siguiente función para convertir un array de enteros a una cadena de caracteres:

```
public static String bufferToString(int[] bufer) {
StringBuilder out = new StringBuilder();
for(int i = 0 ; i < bufer.length-1 ; i++) {
out.append(bufer[i] + ", ");
}
out.append(bufer[bufer.length-1]);
return out.toString();
}
```

Puede emplear el siguiente fragmento de código para generar la lista de valores aleatorios:

```
int range = 10;
for(int i = 0 ; i < bufer.length ; i++) {
bufer[i] = (int)(Math.random() * range);
}
```

3. (Integración Remota con RMI, 3.0 puntos) Se desea disponer de una arquitectura distribuida bajo el framework Java-RMI que calcule la integral definida de una función real de variable real f dada por $f(x) = x^2$ en $[0, 1]$ utilizando la regla del punto medio, en su versión compuesta (ver Figura 1). Desarrolle tal arquitectura teniendo en cuenta las especificaciones siguientes:

   - Debe redactar la interfaz, el servidor y el cliente en los ficheros `interfaz.java, servidor.java` y `cliente.java`.
   - La interfaz dispondrá de un método integral que servirá para indicar al servidor desde el cliente el número n de subintervalos donde se aplicará la regla del punto medio. El método devolverá al cliente el valor aproximado de la integral definida.
   - Dispondrá también de un método de reinicio que servirá para que el servidor inicie un nuevo cálculo.

   Ejemplo de métodos de la interfaz:

   ```
   public float integral(int n);
   public void reinicio();
   ```

El servidor implementará los métodos especificados en la interfaz, dividiendo el intervalo $[0, 1]$ en n subintervalos, aplicando la regla del punto medio en cada uno (ver Figura), y sumando las aproximaciones parciales para construir la aproximación total, que retornará al cliente. El cliente leerá desde teclado el número de subintervalos n, y pedirá al servidor computar la integral buscada de forma aproximada. Una vez recibida, la imprimirá en pantalla.

AYUDA: La regla del punto medio establece la aproximación buscada mediante la ecuación:

$$\int_a^b f(x)\,dx \approx (b-a) f(\frac{a+b}{2})$$

que se ilustra gráficamente en su versión compuesta en la Figura 1, aplicada en cuatro subintervalos del intervalo principal:

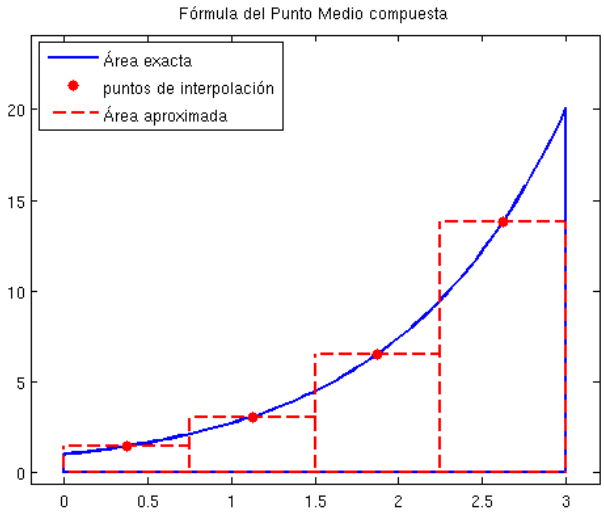

Figura 1: Regla del Punto Medio Compuesta