

Programación Concurrente y de Tiempo Real*

Grado en Ingeniería Informática

Asignación de Prácticas Número 12

La presente asignación tiene por objetivo habituarle al desarrollo de soluciones distribuidas con paso de mensajes utilizando la arquitectura RMI y control de la concurrencia cuando sea necesario, además de introducirle a la solución de problemas bajo un enfoque de paralelismo multicomputador. Documente todo su código con etiquetas (será sometido a análisis con `javadoc`). Si lo desea, puede también agrupar su código en un paquete de clases, aunque no es obligatorio.

1 Ejercicios

1. Considere de nuevo el problema de encontrar número primos en un rango dado que ya resolvió con un enfoque paralelo en la asignación de prácticas número 7. Utilizando las partes de esa solución que considere útiles, escriba ahora una solución basada en un enfoque de paralelismo con multicomputador. Ahora, cada trozo del problema se envía a distintos servidores que residen idealmente en máquinas diferentes, quedando el cliente a la espera de que las llamadas remotas se resuelvan para obtener las soluciones parciales y unificarlas en una única solución global. Utilice RMI para dar solución al problema de la búsqueda de número primos suponiendo que tiene cuatro servidores diferentes. Llame a sus ficheros `iPrimos.java`, `sPrimos.java` y `cPrimos.java`. Tome tiempos para diferentes tamaños del problema y compare el rendimiento de la versión "multicomputador" con la versión paralela que realizó en la práctica número 7. Guarde su comparación y sus conclusiones en `comparativa.pdf`.

2. Ahora va a implementar una arquitectura RMI con *callback* de cliente. Para ello, descargue el documento `guion.callback.pdf` disponible en la carpeta la práctica, junto con los ficheros de código java que en l se detallan. No se le pide que desarrolle ningún código propio en esta ocasión, pero sí que entienda el mecanismo básico de funcionamiento del *callback*.

3. Se desea disponer de un sistema de encuestas de opinión remoto. Suponga que únicamente se va a encuestar un tema concreto a través de una pregunta única, a la que se podrá responder sí, no ó ns/nc. Utilizando RMI, escriba

*©Antonio Tomeu

una aplicación servidora que acepte votos, guarde el recuento y proporcione información estadística sobre el resultado: número de votos emitidos, clasificación por categorías de respuesta, etc. Debe desarrollar la interfaz `intVotoRemoto.java`, el servidor `servVotoRemoto.java` y código de cliente en `clientVotoRemoto.java`. Ponga especial cuidado con la situación en que diferentes clientes votan concurrentemente a la hora de guardar el recuento, porque deberá proteger el recuento mediante algún protocolo de exclusión mutua.

2 Procedimiento y Plazo de Entrega

Se ha habilitado una tarea de subida en *Moodle* que le permite subir cada fichero que forma parte de los productos de la práctica de forma individual en el formato original. Para ello, suba el primer fichero de la forma habitual, y luego siga la secuencia de etapas que el propio *Moodle* le irá marcando. Recuerde además que:

- No debe hacer intentos de subida de borradores, versiones de prueba o esquemas de las soluciones. *Moodle* únicamente le permitirá la subida de los ficheros por **una sola vez**.
- La detección de plagio (copia) en los ficheros de las prácticas, o la subida de ficheros vacíos de contenido o cuyo contenido no responda a lo pedido con una extensión mínima razonable, invalidará plenamente la asignación, sin perjuicio de otras acciones disciplinarias que pudieran corresponder.
- El plazo de entrega de la práctica se encuentra fijado en la tarea de subida del Campus Virtual.
- Entregas fuera de este plazo adicional no serán admitidas, salvo causa de fuerza mayor debidamente justificadas mediante documento escrito.
- Se recuerda que la entrega de todas las asignaciones de prácticas es recomendable, tanto un para un correcto seguimiento de la asignatura, como para la evaluación final de prácticas, donde puede ayudar a superar esta según lo establecido en la ficha de la asignatura.