

Programación Concurrente y de Tiempo Real^{*}

Grado en Ingeniería Informática

Asignación de Prácticas Número 12

Resumen

Se le plantean a continuación un conjunto de ejercicios sencillos de programación sobre paralelismo comparado Java-C++. Para cada uno, debe desarrollar un programa independiente que lo resuelva. **Documente todo su código con etiquetas (será sometido a análisis con javadoc).**

1. Ejercicios

1. Ya ha desarrollado soluciones basadas en el procedimiento de *Monte-Carlo*. En esta ocasión se le pide aplicarlo al cálculo de una aproximación numérica iterativa al número π , y paralelizarla para determinar la aceleración óptima que puede obtenerse. Para ello:

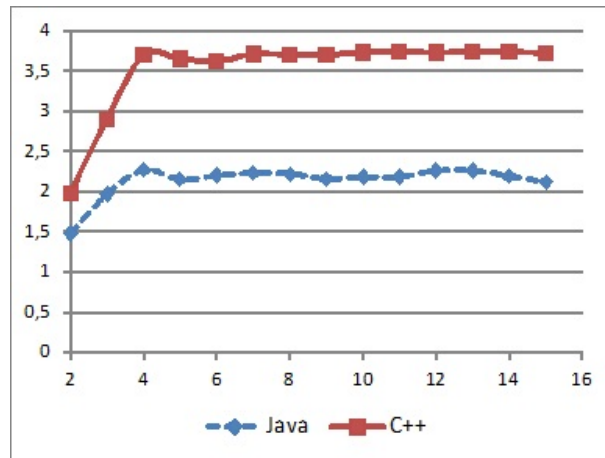
1. Escriba una solución de hebra única para el problema utilizando los lenguajes Java y C++ y guárdela en los ficheros `piSecuencial.java` y `piSecuencial.cpp` respectivamente.
2. Escriba ahora versiones paralelas de los códigos anteriores, y guárdelas en `piParalelo.java` y `piParalelo.cpp`. **El número de tareas paralelas se introducirá en ambos casos por la línea de comandos**, junto con el número de puntos con los que realizar la aproximación.

Utilice en ambos lenguajes objetos atómicos, para proveer la exclusión mutua que necesite sobre los contadores de puntos válidos.

2. Desarrolle sobre Linux una curva $Speed-Up = f(tareas)$ y otra $Tiempo = f(tareas)$, que permitan comparar el desempeño de ambos lenguajes para resolver el problema. Cada curva integrará en una sola imagen los datos para ambos lenguajes, como en la figura¹ siguiente:

^{*}©Antonio Tomeu

¹La curva de la imagen muestra la apariencia deseada que debe obtener en su producto, y no presupone nada acerca de sus propios resultados.



Desarrolle ahora las mismas curvas sobre Windows. Finalmente, Realice un análisis de los resultados obtenidos, justifíquelos en el contexto de lo aprendido en la asignatura, y guarde sus curvas y sus comentarios en un fichero `piAnalisis.pdf`.

3. Repita todas las tareas anteriores utilizando de nuevo ambos lenguajes para el conocido problema del productor-consumidor, utilizando el API de alto nivel en el caso de Java. Cada productor y consumidor realizará un ciclo de n inserciones y extracciones sobre un array de 100 ranuras. Habrá 5 tareas productoras y cinco consumidoras. El buffer, en ambos lenguajes, estará encapsulado en un monitor, y la nomenclatura de ficheros será la siguiente: monitor productor-consumidor Java residirá en `pcMonitor.java` y en C++ residirá en `pcMonitor.cpp`; los diseños de hebra respectivos se guardarán en los ficheros `pcHebras.java` y `pcHebras.cpp`

4. Desarrolle de nuevo, para sistemas operativos Linux y Windows una curva $Tiempo = f(n)$ con un número de ítems n como variable, comenzando en 10^3 y acabando en 10^4 , que permita comparar el desempeño de ambos lenguajes para procesar los ciclos de producción y consumo. La curva integrará en una sola imagen los datos para ambos lenguajes, como en la figura anterior. Realice un análisis de los resultados obtenidos, justifíquelos, y guarde sus curvas y comentarios en un fichero `pcAnalisis.pdf`.

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:

- Los documentos escritos que no sean ficheros de código deben generarse **obligatoriamente** utilizando Latex, a través del editor Overleaf, disponible en la nube. Tiene a su disposición en el Campus Virtual un manual que le permitirá desarrollar de forma sencilla y eficiente documentos científicos de alta calidad. Puede encontrar el citado manual en la sección dedicada a Latex en el bloque principal del curso virtual. El url de Overleaf es: <https://www.overleaf.com/>
- 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 o 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.