

Programación Concurrente y de Tiempo Real

Grado en Ingeniería Informática

Asignación de Prácticas Número 10

Se le plantean a continuación algunos ejercicios sencillos de programación distribuida con paso de mensajes mediante MPJ-Express. Debe desarrollar los programas que se piden. **Documente todo su código con etiquetas (será sometido a análisis con javadoc). Todos los ficheros de código deben incluir las órdenes de compilación y ejecución que propone como comentarios.**

1. Enunciados

1. Escriba un programa donde el proceso master/root (`id=0`) envíe dos vectores enteros de cuatro componentes a un proceso slave (`id=1`). El proceso slave efectuará el producto interno (producto escalar) de ambos vectores, y devolverá el resultado al proceso master, que lo imprimirá en pantalla. Fije los valores de ambos vectores en el código, y utilice los métodos **Send/Recv** para comunicación de datos entre procesos. Incluya un comentario al principio del código, con las órdenes de compilación (`javac`) y ejecución (`mpjrun.bat`) que propone para gestionar su código. Guarde su trabajo en `prodInterno.java`.

2. Escriba un programa donde el proceso master/root (`id=0`) envíe utilizando **Bcast** un array de diez componentes enteros a cuatro procesos slaves (`id=1, 2, 3, 4`). Fije los valores del vector en el código. Cada slave efectúa un escalado del vector según su identificador (por ejemplo, el proceso slave con `id=3`, escala el vector multiplicando por tres todas sus componentes) y lo imprime en pantalla. Incluya un comentario al principio del código, con las órdenes de compilación (`javac`) y ejecución (`mpjrun.bat`) que propone para gestionar su código. Guarde su trabajo en `escalMultiple.java`.

3. Escriba con MPJ-Express un programa que efectúe de manera distribuida la búsqueda de números primos en el rango $[0 - 10^7]$. El proceso 0 enviará a todos los procesos del comunicador el tamaño del rango de análisis que les corresponde mediante **Bcast**; cada proceso efectuará la búsqueda de números primos en el rango que le corresponde, y volcarán resultados al proceso 0 mediante **Reduce**. Guarde su trabajo en `distributedIntegers.java`.

2. Procedimiento de Entrega

PRODUCTOS A ENTREGAR

- Ejercicio 1: `prodInterno.java`
- Ejercicio 2: `escalMultiple.java`
- Ejercicio 3: `distributedIntegers.java`

MÉTODO DE ENTREGA: Tarea de Moodle.