

Programación Paralela y Sistemas Distribuidos

Trabajo Práctico N°1: Conceptos básicos para la construcción de Sistemas Distribuidos

1) Escriba un servidor que, usando sockets, reciba un mensaje de texto y lo repita a su cliente. Programe también el cliente para verificar y probar el comportamiento del servidor. Realice la implementación en TCP y comente los resultados.

Al conectarse el cliente con el servidor, este muestra que el cliente se conectó con su IP y puerto. El cliente le envía un mensaje y el servidor responde, luego el cliente se desconecta.

Al intentar conectar muchos clientes el servidor atiende de a un cliente por vez ya que este no fue programado para recibir más de una petición.

2) Modifique el programa anterior para que pueda atender varios clientes a la vez.

En este caso el servidor acepta las conexiones de todos los clientes que se conecten, encolando las respuestas por orden de llegada.

5) Escribir un servicio que devuelva información de clima del lugar donde reside el servidor. Esta información podrá generarse de forma aleatoria. Deberá ser realizado con RMI. Para ello considere la interface remota, la clase (lado servidor) que implementa esa interface, el servidor, y un cliente que permita probar este funcionamiento.

Se ejecuta un servidor que crea una instancia remota de un objeto que provee, de manera aleatoria, el estado del clima. El cliente se conecta al servidor, hace referencia a esta instancia y llama al método que devuelve el estado del clima. El cliente tiene que conocer este método, de otra manera no es posible la conexión.

6) Escribir un servidor utilizando RMI, que ofrezca la posibilidad de sumar y restar vectores de enteros. Introduzca un error en su código que modifique los vectores recibidos por parámetro. ¿Qué impacto se genera? ¿Qué conclusión saca sobre la forma de pasaje de parámetros en RMI? Mostrar claramente los valores de los vectores del lado cliente, antes y después de la ejecución de la suma o resta.

Cuando se pasan los vectores sin error en el servidor, el resultado es el esperado y los vectores luego tienen los mismos valores pasados. En cambio cuando se introduce el error, el resultado no es el esperado pero los valores de los vectores siguen siendo los originales ya que los parámetros se pasan por valor y no por referencia.

7) Implemente un servidor RMI que resuelva tareas genéricas. Para ello tener en cuenta la interface Tarea, que tendrá un método ejecutar(). El objetivo es que desde el cliente se puedan escribir objetos (que implementen la interface Tarea) que hagan un cálculo concreto (calcular un número aleatorio, un primo, el valor de Pi con cierta precisión, etc.), y que esa tarea se pase al servidor RMI, quien hará el cálculo y devolverá el valor al cliente.

Es un caso parecido al ítem 5 con la salvedad que al servidor que implementa la interface IRMI tiene un método “ejecutar” el cual recibe como parámetro un objeto de tipo genérico (en nuestro caso denominado ITask) que es una clase con alguna operación y devuelve un objeto genérico también. De esta manera podemos ir agregando funcionalidades al sistema y mientras el cliente sepa de estas va a poder acceder a todas las propiedades que el sistema brinde.