

Sistemas Operativos Avanzados

Semestre 2-2016 :: Proyecto 1 :: Opción 2

Implementación de un mecanismo de RPC

1. Planteamiento

Se desea que implementen un mecanismo de RPC que permita ejecutar procedimientos definidos en programas escritos en diferentes lenguajes de programación. Su solución debe cumplir con los siguientes requisitos:

1.1. Requisitos generales

- Deben implementar un proceso servidor y un proceso cliente interoperables en dos lenguajes de programación diferentes.
 - Es decir, si escogieron C++ y Python como lenguajes, entonces deben implementar un servidor y un cliente en Python y un servidor y un cliente en C++.
- El protocolo de comunicación a utilizar debe ser JSON-RPC 2.0
 - Implementar las extensiones de JSON-RPC 2.0 es opcional.
- El protocolo de transporte entre el servidor y el cliente debe ser únicamente HTTP/1.1, utilizando el método POST.
- La solución debe compilar y ejecutarse correctamente en sistemas operativos basados en Debian.

1.2. Requisitos del lado del servidor

- El servidor debe ser capaz de recibir peticiones JSON-RPC mediante HTTP.
- Al recibir una petición, el servidor debe verificar lo siguiente:
 - Que el método a ejecutar existe y es un método que el servidor efectivamente exporta al cliente.
 - Es decir, se debe utilizar un mecanismo de despacho para la ejecución del método del lado del servidor.

- Que los parámetros recibidos sean válidos para la ejecución del método seleccionado. Específicamente, se debe verificar que:
 - El tipo de cada parámetro es el esperado.
 - Si el paso de parámetros es por lista, entonces los parámetros deben estar en el orden correcto.
 - Si el paso de parámetros es por nombre, entonces los nombres deben coincidir con los de la firma del método a ejecutar.
- El servidor debe enviar el resultado del método ejecutado al cliente en un mensaje JSON-RPC mediante HTTP.

1.3. Requisitos del lado del cliente

- Se debe implementar un *client stub* en cada lenguaje de programación implementado.
 - Por ejemplo, mediante una biblioteca de enlace dinámico/estático en C++, un JAR externo en Java, o un módulo en Python.
- El cliente debe usar los métodos definidos en el *client stub* como llamadas a procedimientos locales.

1.4. Métodos a implementar

Los métodos a implementar en los servidores deben corresponder a una calculadora científica avanzada con las siguientes funcionalidades, según disponibilidad en los lenguajes de programación seleccionados.:

- Operaciones aritméticas:
 - Suma.
 - Resta.
 - Multiplicación.
 - División.
 - Potencia.
 - Raíz cuadrada.
- Operaciones trigonométricas:
 - Seno.
 - Coseno.
 - Tangente.
 - Tan2.

- Arcoseno.
- Arcocoseno.
- Arcotangente.
- Funciones Trascendentes:
 - Seno hiperbólico.
 - Coseno hiperbólico.
 - Tangente hiperbólica.

- Al menos dos operaciones sobre números complejos.

Las operaciones aritméticas deben ser implementadas de forma polimórfica tanto para números enteros como para números punto flotante de manera separada. Las operaciones trigonométricas solo deben definirse para números punto flotante.

2. Herramientas

- Especificación de JSON-RPC 2.0 <http://www.jsonrpc.org/specification>
- Para el lenguaje de programación C++ se recomiendan estas bibliotecas:
 - [cURL](#) para el uso del protocolo HTTP.
 - [Json-spirit](#) para procesar texto en JSON.

3. Evaluación

La evaluación consistirá en la ejecución de sus proyectos para verificar lo siguiente:

- La correcta y completa implementación de todos los requisitos planteados.
- La interoperabilidad entre todos sus clientes y servidores.