



Llamadas a Procedimientos Remotos - RPC

RPC (DEL INGLES REMOTE PROCEDURE CALL)

Introducción

Es un protocolo que le permite a un programa ejecutar procedimientos (código) situados en el espacio de direcciones de otro proceso, el cual puede residir en otra máquina.

- ▶RPC generan un modelo cliente-servidor.
- ▶Pretende distribuir un programa ordinario a través de nodos conectados mediante una red (aprovechar servicios no disponibles de manera local sin perder la modularidad del diseño).

Introducción

- ▶ Le hace parecer al programador que está ocurriendo una simple llamada local.
- ▶ Cuando es combinado con el protocolo compilador RPCGEN, en forma transparente genera llamadas remotas.

Las implementaciones RPC más populares son:

- ▶ La desarrollada por Sun Microsystems denominada ONC-RCP (Open Network Computing, ONC-RCP).
- ▶ La desarrollada por Microsoft en línea con el Ambiente de Computación Distribuida (DCE, Distributed Computing Environment).

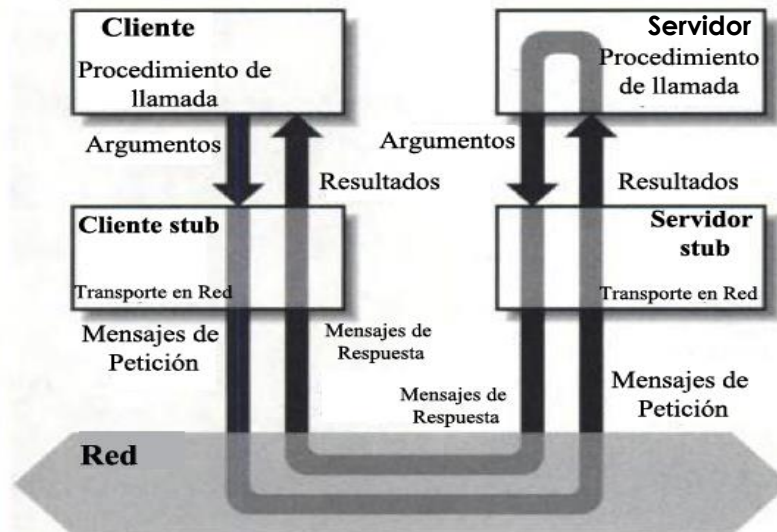
Características de RPC

- ▶ Simplifica el desarrollo de aplicaciones distribuidas.
 - ▶ Evita que el programador interactúe directamente con la interfaz de Sockets.
 - ▶ Oculta los detalles de la implementación de llamadas remotas.
 - ▶ Implementa un dialogo de petición - respuesta.
-
- ▶ **Mensaje de petición:** Identifica al procedimiento que se esta llamando y además contienen a los parámetros de la llamada.
 - ▶ **Mensaje de respuesta:** Contiene los valores devueltos y se encarga de gestionar los contenidos de los mensajes empaquetando y formateando los datos

Funcionamiento General de RPC



Llamada a Procedimiento Local



Llamada a Procedimiento Remoto

En una llamada a procedimiento remoto, el cliente y el servidor se ejecutan como procesos separados. Por tanto, no necesariamente corren sobre la misma máquina

Los dos procesos se comunican a través de stubs, existe uno en el cliente y otro en el servidor. Los stubs son bloques de código que contienen funciones que mapean llamadas a procedimientos locales en una serie de llamadas a procedimientos remotos

RFC (Request for Comments)

- ▶ **RFC:** Es una serie de documentos técnicos acerca de internet, los cuales incluyen protocolos, procedimientos, programas y conceptos, así como opiniones, etc. Estos documentos técnicos y normativos son producidos por la IETF (Internet Engineering Task Force), la IAB (Internet Architecture Board) o la IRTF (Internet Research Task Force).
 - ▶ Son numeradas consecutivamente lo cual les proporciona un ID.
 - ▶ Son publicadas en línea a través de diferentes repositorios.
 - ▶ Más referencias en : www.rfc-editor.org/RFCoverview.html.

ONC-RPC (Open Network Computing – Remote Procedure Call)

- ▶ Actualmente se encuentra en la versión 2 descrita en la RFC 1831 (Agosto 1995). La RFC 1832 provee la descripción de XDR (Agosto,1995).

Sus componentes son:

- ▶ **RPCGEN:** Un compilador que toma la definición de la interfaz de un procedimiento remoto y genera el stub del cliente y el stub del servidor.
- ▶ **XDR (eXternal Data Representation):** Estandar para la descripción y codificación de datos que garantiza portabilidad entre sistemas de arquitecturas diferentes.
- ▶ Una biblioteca para el manejo de detalles (comunicación entre otras).

Características de XDR

- ▶ Su sintaxis es similar a la del lenguaje de programación C.
- ▶ Empleado para la transferencia de información entre arquitecturas (Capa de Presentación – OSI).
- ▶ Empleado para la definición de Interfaces.
- ▶ Solo envía el valor de la variable por la red (Implicit Typing).
- ▶ No es un lenguaje de programación (Lenguaje de descripción de datos).

Tipos de datos manejados por XDR

- ▶ **Int**
- ▶ **Unsigned int**
- ▶ **Enum**
- ▶ **Bool**
- ▶ **Hyper**
- ▶ **Unsigned hyper**
- ▶ **Float**
- ▶ **double**
- ▶ **Quadruple**(punto flotante de cuadruple presición)
- ▶ **Opaque** (de longitud fija o variable)
- ▶ **String**
- ▶ **Array** (de longitud fija o variable)
- ▶ **Struct**
- ▶ **Union**
- ▶ **Void**
- ▶ **constant**

Aspectos de transparencia de ONC-RPC

Se debe depurar las funciones como funciones locales y posteriormente realizar las alteraciones para convertirlas en funciones remotas:

- ▶ **Paso de Parámetros.**

- ▶ Solo se permite pasar parámetros por valor.

- ▶ **Enlace.** El cliente debe contactar al sistema remoto apropiado para la ejecución de un procedimiento específico

- ▶ El host remoto
- ▶ El proceso servidor deseado en dicho host.

Conversión de una llamada local en una RPC

- ▶ **Protocolo de Transporte.**

- ▶ Tanto el protocolo ONC como el DCE soportan a TCP y UDP como protocolos de transporte

- ▶ **Manejo de Excepciones.**

- ▶ La implementación provee mecanismos para la detección de fallos, ya que en un ambiente distribuido la posibilidad de fallos aumenta

- ▶ **Representación de los Datos.**

- ▶ Dado que el cliente y el Servidor se pueden ejecutar en sistemas de arquitecturas diferentes. La representación estandarizada de datos empleada XDR.

Conversión de una llamada local en una RPC

- ▶ **Desempeño.**

- ▶ Este disminuye entre 10 y 100 veces en comparación con un procedimiento local, sin embargo reduce el tiempo de desarrollo de aplicaciones distribuidas.

- ▶ **Seguridad.**

- ▶ En RPC existen los mismos problemas de seguridad relacionados con la ejecución remota de comandos.

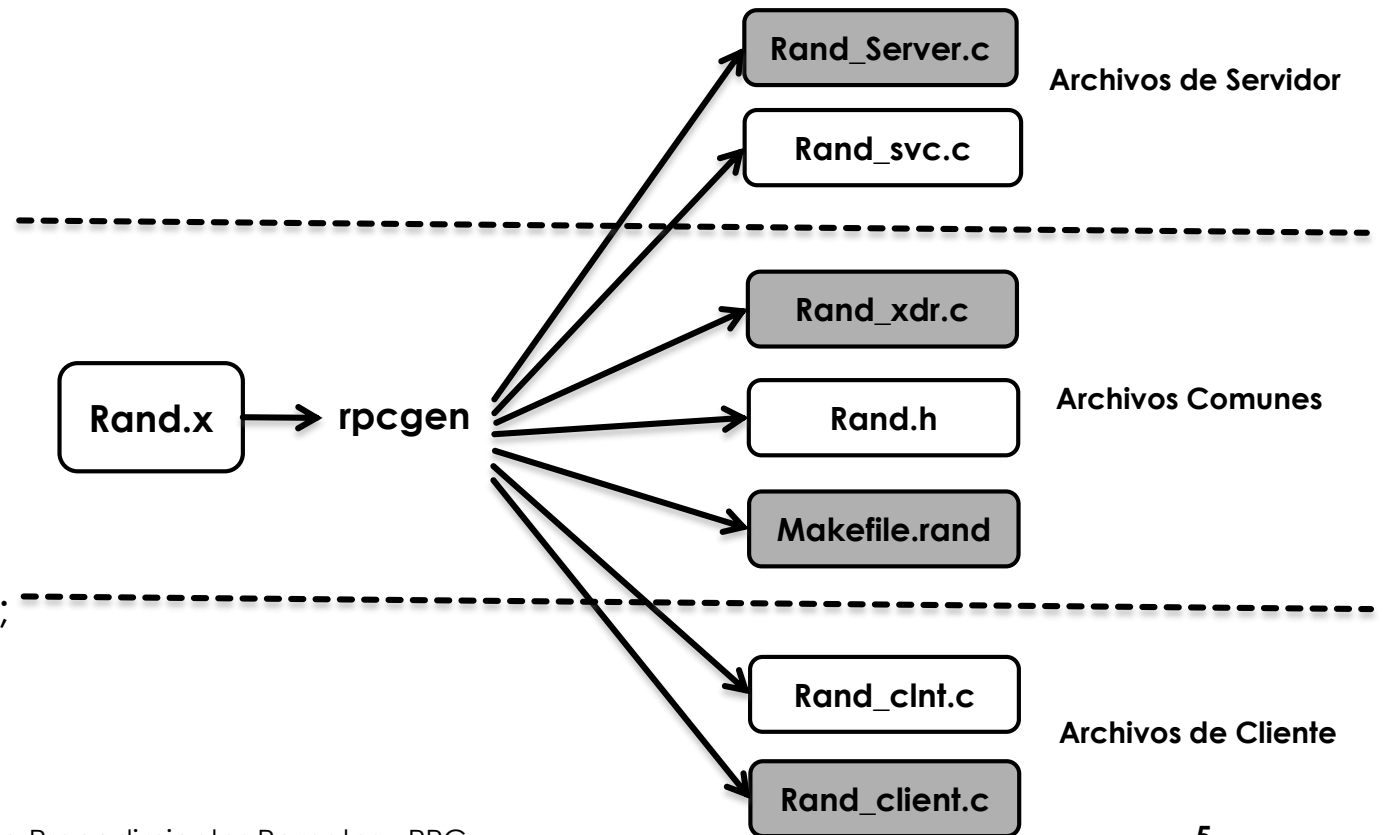
Ejemplo: Generación de números aleatorios mediante un servicio local

- ▶ Desarrollar un programa que genere números aleatorios haciendo uso de las funciones incluidas en la biblioteca **stdlib.h**:
 - ▶ `srand48(int semilla);`
 - ▶ `drand48();`

Ejemplo: Conversión de un Servicio Local a un Servicio Remoto Mediante RPCGEN

Paso 1: Definición de la interfaz

```
/* ---- rand.x ---- */  
program RAND_PROG {  
    version RAND_VERS {  
        void INICIALIZA_RANDOM(long) = 1;  
        double OBTIENE_NUEVO_RANDOM(void) = 2;  
    } = 1;  
} = 0x31111111;
```



Ejemplo: Conversión de un Servicio Local a un Servicio Remoto Mediante RPCGEN

- ▶ **Makefile.proto:** Es el archivo *makefile* para compilar todo el código del cliente y el servidor
- ▶ **Proto_clnt.c:** Contiene el stub del cliente (normalmente no se modifica)
- ▶ **Proto_svc.c:** Contiene el stub del servidor (normalmente no se modifica)
- ▶ **Proto.h:** El encabezado que contiene todos los tipos XDR (Aquí puede verse como rpcgen convirtió los tipos definidos en el archivo .x)

Ejemplo: Conversión de un Servicio Local a un Servicio Remoto Mediante RPCGEN

- ▶ **Proto_client.c:** Contiene un esqueleto del programa principal del cliente (insertar código que prepare los valores de los argumentos para el servicio remoto)
- ▶ **Proto_server.c:** Contiene los stubs de los servicios remotos (puede ser necesario modificar la forma en que estas funciones utilizan los parámetros)
- ▶ **Proto_xdr.c:** contiene filtros XDR que el stub del cliente y el servidor necesitan (normalmente no se modifica)

Ejemplo: Conversión de un Servicio Local a un Servicio Remoto Mediante RPCGEN

- ▶ **Paso 2:** Ejecutar RPCGEN para generar los archivos necesarios a partir del archivo de especificación “.x”.
 - ▶ **Rpcgen -C -a rand.x**
- ▶ **Paso 3:** Modificar el archivo **rand_client.c** de modo que contenga el código del cliente.
- ▶ **Paso 4:** Modificar el archivo **rand_svc.c** de modo que contenga las funciones que se invocarán remotamente.

Compilación de la Aplicación

- ▶ Compilación del Cliente:
 - ▶ `cc -c rand_client.c`
 - ▶ `cc -c rand_clnt.c`
 - ▶ `cc -o RandCliente rand_client.o rand_clnt.o`
- ▶ Compilación del Servidor:
 - ▶ `cc -c rand_server.c`
 - ▶ `cc -c rand_svc.c`
 - ▶ `cc -o RandServer rand_server.o rand_svc.o`
- ▶ Ejecución del Servidor y del Cliente:
 - ▶ `./RandServer`
 - ▶ `./RandCliente 127.0.0.1 2333 10`

Entornos RPC

- ▶ **SUN-RPC** (ONC-RPC Network Computing - RPC)
- ▶ **DCE/RPC** (Distributed Computing Environment RPC)
- ▶ **Java-RMI** (Remote Methods Invocation)
- ▶ **CORBA** (Common Object Requesting Broker Architecture)
- ▶ **SOAP** (Simple Object Access Protocol)
- ▶ **DCOM** (Distributed Component Object Model)
- ▶ **.NET Remoting**