

# Programación Distribuida y Tiempo Real

## Facultad de Informática - Universidad Nacional de La Plata

### Practica 2

1) Para los ejemplos de RPC proporcionados (\*.tar, analizar en el orden dado a los nombres de los archivos):

- a.- Mostrar cómo serían los mismos procedimientos si fueran locales, es decir haciendo el proceso inverso del realizado en la clase de explicación de RPC.
- b.- Ejecutar los procesos y mostrar la salida obtenida (del “cliente” y del “servidor”) en cada uno de los casos.
- c.- Mostrar experimentos donde se produzcan errores de conectividad del lado del cliente y del lado del servidor. Si es necesario realice cambios mínimos para, por ejemplo, incluir `sleep()` o `exit()`, de forma tal que no se reciban comunicaciones o no haya receptor para las comunicaciones. Verifique con UDP y con TCP.

2) Describir/analizar las opciones

- a) - N
- b) -M y - A

verificando si se pueden utilizar estas opciones y comentar que puede ser necesario para tener procesamiento concurrente del “lado del cliente” y del “lado del servidor” con la versión utilizada de `rpcgen`. Una lista completa de opciones se describe en <http://download.oracle.com/docs/cd/E19683-01/816-1435/rpcgenpguide-1939/index.html>

3) Analizar la *transparencia* de RPC en cuanto al manejo de parámetros de los procedimientos remotos. Considerar lo que sucede en el caso de los valores de retorno. Puede aprovechar los ejemplos provistos.

4) Con la finalidad de contar con una versión muy restringida de un sistema de archivos remoto, en el cual se puedan llevar a cabo las operaciones enunciadas informalmente como

- leer: dado un nombre de archivo, una posición y una cantidad de bytes a leer, retorna 1) los bytes efectivamente leídos desde la posición pedida y la cantidad pedida en caso de ser posible, y 2) la cantidad de bytes que efectivamente se retornan leídos.
- escribir: dado un nombre de archivo, una cantidad de bytes determinada, y un buffer a partir del cual están los datos, se escriben los datos en el archivo dado. Si el archivo existe, los datos se agregan al final, si el archivo no existe, se crea y se le escriben los datos. En todos los casos se retorna la cantidad de bytes escritos.

- a.- Defina e implemente con RPC un servidor. Documente todas las decisiones tomadas.
- b.- Implemente un cliente RPC del servidor anterior que copie un archivo del sistema de archivos del servidor en el sistema de archivos local y genere una copia del mismo archivo en el sistema de archivos del servidor. En todos los casos se deben usar las operaciones de lectura y escritura del servidor definidas en el ítem anterior, sin cambios específicos del servidor para este ítem en particular. Al finalizar la ejecución del cliente deben quedar tres archivos en total: el original en el lado del servidor, una copia del original en el lado del cliente y una copia en el servidor del archivo original. El comando diff no debe identificar ninguna diferencia entre ningún par de estos tres archivos.

#### 5) Timeouts en RPC:

- a.- Desarrollar un experimento que muestre el timeout definido para las llamadas RPC y el promedio de tiempo de una llamada RPC.
- b.- Reducir el timeout de las llamadas RPC a un 10% menos del promedio encontrado anteriormente. Mostrar y explicar el resultado para 10 llamadas.
- c.- Desarrollar un cliente/servidor RPC de forma tal que siempre se supere el tiempo de timeout. Una forma sencilla puede utilizar el tiempo de timeout como parámetro del procedimiento remoto, donde se lo utiliza del lado del servidor en una llamada a sleep( ), por ejemplo.

**Material de consulta.** Puede resultar importante y/o útil revisar las definiciones de los RFCs de XDR y RPC:

- RFC 4506 - XDR: External Data Representation Standard en <http://tools.ietf.org/html/rfc4506>
- RFC 5531 - RPC: Remote Procedure Call Protocol Specification Version 2 en <http://tools.ietf.org/html/rfc5531>

#### **Entrega de la practica** (individual o en grupos de dos alumnos como máximo):

- Se debe entregar un único informe detallando lo realizado para cada ejercicio. Debe tener un formato bien definido identificando materia, trabajo práctico y autor/es. Se debe entregar en formato electrónico con tipo de archivo .pdf, en tamaño de hoja A4.
- Para cada programa modificado o generado para resolver los ejercicios, debe explicarse el cambio o la implementación realizada. Si bien el programa fuente puede estar comentado, el cambio o la implementación realizada debe explicarse en el texto del informe (no es aceptable “ver código fuente” en el informe).

- Se debe entregar en formato electrónico tanto el informe como todo el código fuente usado/desarrollado.