

Laboratorio de Sistemas Operativos
Semestre A-2018
Práctica Cuatro
Laboratorio

Prof. Rodolfo Sumoza
Prep. Alvaro Araujo
Prep. Luis Sanchez

1. Comunicación Remota de Procesos.

Con el fin de realizar tareas compartidas o de compartir información es de interés que dos o mas procesos se comuniquen de manera remota. A continuación se estudian dos formas de comunicación basadas en una arquitectura cliente-servidor, estas formas de comunicación son los Sockets y los RPC.

2. Sockets.

Un socket se trata de un mecanismo que permite establecer una conexión entre dos procesos, que por lo general se encuentran en ejecución en maquinas diferentes conectadas a través de la red.

Existen básicamente dos tipos sockets:

- Sockets orientados a conexión: con este tipo de sockets ambos procesos deben conectarse entre si y hasta que no se encuentre establecida correctamente la conexión, no empezará la transmisión de datos por ninguna de las partes. Este tipo de sockets utiliza el protocolo TCP y se utilizan cuando la información transmitida es de suma importancia y no debe perder ningún dato, la utilización de este tipo de sockets implica que los procesos se queden bloqueados esperando o transmitiendo datos.
- Sockets no orientados a conexión: en este tipo de sockets no es necesario que exista una conexión entre los procesos para realizar una transmisión

de datos, en cualquier momento uno de los procesos puede transmitir datos sin importar si el otro proceso está escuchando. Este tipo de sockets utiliza el protocolo UDP lo que implica que no existe sincronización entre el origen y el destino.

Los pasos a seguir para la conexión con sockets en el programa que realizará la tarea de servidor, son los siguientes:

1. Apertura de un socket, mediante la función `socket()`.
2. Asociar el programa con el socket recién abierto mediante la función `bind()`.
3. Atender la conexión a través de la función `listen()`.
4. Aceptar conexiones de clientes a través de la función `accept()`.
5. Envío y recepción de datos por medio de las funciones `send()` y `recv()` respectivamente.
6. Cerrar el socket por medio de la función `close()`.

Los pasos a seguir para la conexión con sockets en el programa que realizará la tarea de cliente, son los siguientes:

1. Apertura de un socket, mediante la función `socket()`.
2. Solicitar una conexión a través de la función `connect()`.
3. Envío y recepción de datos por medio de las funciones `send()` y `recv()` respectivamente.
4. Cerrar el socket por medio de la función `close()`.

3. RPC (Remote Procedure Call).

Una llamada remota a un proceso es un mecanismo por medio del cual un proceso realiza una petición a otro proceso que se encuentra ejecutándose en una maquina conectada a la misma red, solicitando que ejecute alguna funcionalidad, el proceso que cumple la función de servidor responderá informando los resultados de la ejecución. Esta forma de comunicación presenta grandes ventajas debido a que se puede designar la ejecución de una rutina que necesite grandes prestaciones a una maquina que posea mejor rendimiento. Al igual que los sockets presentan una arquitectura cliente-servidor.

Los pasos a seguir para trabajar con RPC son los siguientes:

- En el servidor
 - Generar los archivos correspondientes a las rutinas que se ejecutarán en el servidor.

- Codificar las funciones que serán llamadas de manera remota.
 - Notificar al SO el nombre y versión de las funciones que se ofrecen.
 - Esperar por alguna llamada a las funciones ofrecidas.
- En el Cliente
 - Generar los archivos correspondientes a las rutinas que se ejecutarán en el cliente.
 - Llamar a las funciones ofrecidas por el servidor.
 - Interpretar los resultados de la ejecución de las funciones en el servidor.
 - Cerrar la conexión con el servidor.

4. Ejercicios

- Modificar el ejemplo de sockets proporcionado junto con esta guía para que el servidor esté en la capacidad de atender múltiples clientes y se pueda escoger la respuesta que es enviada a cada cliente.
- Construir dos programas (uno cliente, otro servidor) que a través una conexión por medio de RPC, se permita al programa cliente ordenar un conjunto de números utilizando diferentes métodos de ordenamiento, alojados en el servidor.