Universidad
Nacional
de Córdoba

Cátedra de Sistemas Operativos II

Trabajo Práctico Nº1

Rivero, Franco Fabián

22 de abril del 2018

Índice:

Introducción:

Este trabajo consta de aprender a utilizar los sockets para la comunicación entre dos sistemas en este caso, seran por un lado una aplicación servidor la cual se hara con una Raspberry PI y la aplicación cliente que se hara sobre una PC. Para aplicar los conocimientos obtenidos sobre socket se va a realizar un bash que haga de interfaz de nuestra pc con la raspberry de modo, que cuando se envie un comando desde nuestra pc se ejecutara en el servidor

Descripción del problema

Se implemento la comunicación entre un servidor que con el que queremos comunicarnos y manejar su sistema operativo. Desde el servidor se deben aceptar comandos linux, por lo que se deberá realizar un login desde cualquier cliente hacia el servidor.

El cliente puede enviar cualquier comando aceptado por linux y ademas poder descargar cualquier archivo desde el servidor hacia el cliente.

Ámbito del Sistema

Para llevar a cabo la implementación del presente trabajo, se decidió utilizar la placa de desarrollo “Raspberry Pi 2” cuyo sistema operativo instalado es “Raspbian”. La elección de dicho sistema operativo no surgió de un estudio de qué sistema sería el óptimo para la implementación, simplemente se decidió por él ya que se encontraba previamente instalado en la placa.

Definiciones, Acrónimos y Abreviaturas

Raspberry Pi: es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de informática en las escuelas.
La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian)

Raspbian: es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Strech (Debian 9.4) para la placa computadora (SBC) Raspberry Pi. Técnicamente el sistema operativo es un port no oficial de Debian armhf para el procesador (CPU) de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento en según qué casos. El port fue necesario al no haber versión Debian armhf para la CPU ARMv6 que contiene el Raspberry PI.

Diseño de solución

El diseño del mismo fue hacer un cliente/servidor con proceso basado en el servidor. El servidor aws será un socket servidor y cada usuario un cliente, ambos orientados a conexión. En este tipo diseño, el servidor tiene la mayor parte de la funcionalidad como es la lógica de aplicación y lo respectivo con la base de datos. En el cliente está la lógica de presentación. Aunque no sería basada en Servidor complemente, ya que en el cliente se levanta un Servidor UDP y en el servidor un cliente UDP.

Implementación y Resultados

Como ya se ha dicho anteriormente, se implemento dos programas que son el servidor aws y el cliente. En el cliente se implementó un servidor TCP/IP. En donde el mismo, esta preparado para tomar los comandos ejecutados por el cliente en el servidor aws.

A su vez, otro proceso se encarga del servidor TCP/IP, donde se le conecta al servidor,espera primero un usuario y una contraseña y luego se espera que le envíe algún comando válido y luego realiza la tarea correspondiente. Sólo en el caso que se reciba el comando “download”, se crea un cliente UDP para realizar el pasaje de información, luego ese cliente termina y sigue esperando que se ingrese otro comando.

Con respecto al cliente se implementó un cliente TCP/IP y un servidor UDP. Ambos procesos son concurrentes. En la parte del cliente se debe estar conectado al servidor mediante un comando especificado anteriormente y luego ejecutar distintos comando, que son mostrados en pantalla. El servidor UDP se queda esperando para cuando el cliente UDP del servidor AWS se le conecte y le pase la información. Los resultados de ambos fueron exitosos con lo que respecta a la consigna pedida.

Conclusiones

El trabajo fue muy útil para terminar de entender la comunicación entre sockets y sus distintos protocolos, ya que son de gran importancia en las comunicaciones. Nos sirvio para poder conectarnos remotamente a cualquier dispositivo a traves de instalar simplemente el servidor implementado con el puerto 6020 libre y de esa forma poder conectarse en cualquier momento.