Universidad
Nacional
de Córdoba

Cátedra de Sistemas Operativos II

Trabajo Práctico Nro I

Ivan M. Vignolles
11 de Abril de 2019

Índice

1 Introducción

Los sockets son una abstracción de comunicación entre procesos (IPC) que, en un sistema tipo UNIX, se implementan en un descriptor de archivo, sobre el cual se envía o recibe información, al igual que como se lee o escribe un archivo **CITATION NEEDED**. Son una herramienta muy importante, uno de los pilares de la comunicación entre procesos, y sumamente utilizada en la mayoría de las aplicaciones de red.

Propósito

El propósito de este trabajo es aplicar los conocimientos adquiridos sobre sockets y llamadas de sistema en general en una aplicación del tipo cliente-servidor que emula la comunicación entre una estación terrena, representada por una PC, y un satélite, que en este caso particular es representado por una Raspberry Pi.

Ámbito del Sistema

Tanto cliente como servidor correrán sobre Linux, el servidor proporcionará un prompt a partir del cual se podrá interactuar con el cliente y hacer diferentes peticiones.

Definiciones, Acrónimos y Abreviaturas

TCP

El protocolo de control de transmisión (TCP por sus siglas en ingles) es uno de los protocolos de la capa de transmisión más utilizados, permite crear conexiones entre dispositivos y de esta manera comunicar un flujo de datos [1].

Entre las principales características del protocolo TCP se pueden mencionar las siguientes: permite poner nuevamente los datagramas en orden cuando vienen del protocolo IP, permite que el monitoreo del flujo de los datos y así evita la saturación de la red, permite que los datos se formen en segmentos de longitud variada para entregarlos al protocolo IP, permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente. Por último, TCP permite comenzar y finalizar la comunicación amablemente [2].

UDP

El protocolo de datagramas de usuario (en inglés: User Datagram Protocol o UDP) es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción [3].

RTT

El tiempo de ida y vuelta (RTT, round-trip time o round-trip delay time) se aplica en el mundo de las telecomunicaciones y redes informáticas al tiempo que tarda un paquete de datos enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino [4].

Prompt

Se llama prompt al carácter o conjunto de caracteres que se muestran en una línea de comandos para indicar que está a la espera de órdenes. Este puede variar dependiendo del intérprete de comandos y suele ser configurable [5].

Descripción General del Documento

El documento se organiza de la siguiente forma: la sección 2 brinda una descripción del proyecto, incluyendo requisitos, funciones, y características. La sección 3 profundiza los requisitos del sistema. En la sección 4 se provee el diseño y en la sección 5 se muestran los resultados obtenidos.

2 Descripción General

Perspectiva del Producto

Tanto la aplicación del cliente como la del servidor serán desarrolladas en lenguaje C. Con el objetivo de simplificar la implementación, teniendo en cuenta que los procesos deben ser mono-thread, el servidor soportará únicamente una conexión a la vez.

Ambos procesos se ejecutaran sobre Linux, parte de la comunicación debe darse sobre un protocolo orientado a la conexión (TCP), y otra porción sobre protocolo no orientado a la conexión (UDP). Se debe hacer un uso eficiente de los buffers de comunicación de forma tal que asegure una comunicación veloz sin tener un gran impacto en la memoria RAM.

El servidor deberá solicitar una autenticación cada vez que ocurra una conexión para dar acceso al prompt desde el cual se pueden hacer consultas al cliente.

Funciones del Producto

Conexión

El sistema conecta dos dispositivos a través de internet y permite que interactúen.

Autenticación

Cada vez que una conexión ocurra, el servidor pedirá que el usuario se autentifique utilizando una contraseña antes de permitirle hacer cualquier petición al cliente.

Consulta de telemetría

El servidor puede hacer una petición al cliente para que éste le envíe información de sus recursos.

Consulta de escaneo

Se puede hacer una petición al cliente para recibir un escaneo de la Tierra.

Actualización de firmware

El servidor puede enviar una nueva versión del firmware del cliente y dar la orden de ejecutarlo, a continuación, el cliente reiniciará la conexión corriendo la versión actualizada.

Características de los Usuarios

Los usuarios deben tener un conocimiento básico de como manejar una consola, como así también deben conocer los comandos que el sistema provee.

Restricciones

Las principales restricciones del sistema se detallan a continuación:

- El servidor solo soporta una conexión a la vez.
- La contraseña de autenticación no puede ser modificada.
- El nombre del archivo que se pase cuando se de la orden de actualizar el firmware debe ser un archivo de firmware válido.
- El nombre de la imagen recibida al solicitar un escaneo no puede ser modificado, en caso de que un archivo con el mismo nombre ya exista, sera sobrescrito.
- El código del cliente y el servidor deben ser escritos en el lenguaje C, con el estilo de código descrito por el kernel de linux y debe ser corroborado con la utilidad *cppcheck*

Suposiciones y Dependencias

Requisitos Futuros

3 Requisitos Específicos

Interfaces Externas

Funciones

Requisitos de Rendimiento

Restricciones de Diseño

Atributos del Sistema

4 Diseño de solución

5 Implementación y Resultados

6 Conclusiones

7 Apéndices

References

[1] *Protocolo de control de transmisión*, https://es.wikipedia.org/wiki/Protocolo_de_control_de_transmisi\%C3\%B3n, Última vez accedido el 29-02-2020.

[2] Carlos Villagómez, *Protocolo tcp*, https://es.ccm.net/contents/281-protocolo-tcp, Última vez accedido el 29-02-2020.

[3] *Protocolo de datagramas de usuario*, https://es.wikipedia.org/wiki/Protocolo_de_datagramas_de_usuario, Última vez accedido el 29-02-2020.

[4] *Tiempo de ida y vuelta*, https://es.wikipedia.org/wiki/Tiempo_de_ida_y_vuelta, Última vez accedido el 29-02-2020.

[5] *Prompt*, https://es.wikipedia.org/wiki/Prompt, Última vez accedido el 29-02-2020.