



CÁTEDRA SISTEMAS OPERATIVOS II

Departamento de Computación FCEFyN - UNC

Gestion de AWS con Sockets

TROMBOTTO, Agustin Mat: 39071116
gutitrombotto@gmail.com

16 de abril de 2017

Version del Documento V1.0

1. Introducción

En la Provincia de Córdoba existen aproximadamente 400 estaciones hidrometeorológicas automáticas (AWS por sus siglas en inglés), de distintas marcas, modelos y configuraciones, que pertenecen a distintas redes, instituciones y organismos, desparrramadas por todo el territorio provincial. Cada AWS consta de una serie de sensores (barómetro, termómetro, sensor de radiación solar, etc.), conectado a un sistema de adquisición de datos, que toma medidas de los sensores (telemetría) un período determinado de tiempo.

Para poder obtener datos de este sistema, se implementa un servidor el cual posee la información y un cliente que, al conectarse, puede obtener los datos de las estaciones mencionadas

1.0.1. Propósito

Se desarrolla un sistema de comunicación mediante sockets implementados en lenguaje C. Clientes se conectan a un servidor el cual debe transferir información sobre las estaciones meteorológicas

1.0.2. Ambito del Sistema

Actualmente el software se puede ejecutar un servidor con localhost montado.

1.0.3. Definiciones, Acrónimos y Abreviaturas

1.0.4. Referencias

1.0.5. Descripción General del Documento

Se detalla, en este documento los objetivos, el diseño e implementación, los bugs del sistema y las conclusiones del mismo.

2. Descripción General

2.0.1. Perspectiva del Producto

El producto consiste en la ejecución de un programa cliente y un programa servidor. Una vez ejecutados, el cliente se conecta al servidor y le podrá solicitar información sobre las estaciones meteorológicas en cuestión al servidor mediante las funciones proveídas por el mismo. El cliente puede ser ejecutado en cualquier dispositivo con una interfaz de comunicaciones. Se le debe establecer previamente el IP del dispositivo a la cual se conectará

2.0.2. Funciones del Producto

El software debe realizar la conexión, control y transferencia de datos telemétricos entre el un cliente y el servidor que contiene los datos de las AWS (arquitectura cliente servidor).

2.0.3. Características de los Usuarios

El usuario debe pedir un usuario y una contraseña al administrador del software el cual creará los datos necesarios para poder establecer la conexión.

2.0.4. Suposiciones y Dependencias

2.0.5. Requisitos Futuros

La implementación del sistema posee funcionalidades pendientes a saber:

- Envío de prompt por parte del servidor al cliente
- Respuesta del servidor con la información correcta según la función a ejecutar
- Traducir y probar en código en un servidor remoto

3. Requisitos Específicos

- Todos los procesos deben ser mono-thread
- Uso de Cppcheck y la compilación con el uso de las flags de warning -Werror, -Wall y -pedantic
- Proveer los archivos fuente, así como cualquier otro archivo asociado a la compilación

3.0.1. Interfaces Externas

3.0.2. Funciones

El servidor debe contar con las siguientes funcionalidades:

- **listar:** muestra un listado de todas las estaciones que hay en la “base de datos”, y muestra de que sensores tiene datos.
- **descargar no_estación:** descarga un archivo con todos los datos de no_estación.
- **diario_precipitacion no_estación:** muestra el acumulado diario de la variable precipitación de no_estación (no_día: acumulado mm).
- **mensual_precipitacion no_estación:** muestra el acumulado mensual de la variable precipitación (no_día: acumulado mm).
- **promedio variable:** muestra el promedio de todas las muestras de la variable de cada estación (no_estacion: promedio).
- **desconectar:** termina la sesión del usuario.

Además el servidor debe proveer al cliente de un prompt de la forma: **usuario@servidor**

También un sistema de autenticación, donde el usuario deberá ingresar su contraseña y el servidor validar si pertenece a un usuario apto para ingresar.

3.0.3. Requisitos de Rendimiento

El sistema deberá tener la opción de ser ejecutado en un servidor remoto en el cual se establece una dirección IP, se levanta un socket en el puerto 6020, y el cliente se deberá poder conectar a dicha IP con dicho puerto desde la misma red.

3.0.4. Restricciones de Diseño

- El sistema no podrá contar con un sistema de gestión de base de datos.
- El sistema se correrá en un equipo que cuenta con las herramientas típicas de consola para el desarrollo de programas (Ej: gcc, make)

3.0.5. Atributos del Sistema

3.0.6. Otros Requisitos

Para la presentación final, el desarrollo se correrá en una INTEL Galileo V1, sobre la cual se comprobará el prototipo.

4. Diseño de solución

Para la realización de dos programas:

- servidor.c : contiene toda la lógica de interconexión con el cliente además de la forma de adquirir los datos telemétricos y la autenticación del usuario
- cliente.c: funciones de interacción con el servidor así también la forma de mostrar los resultados.

4.0.1. Servidor

Se subdivide las funcionalidades realizadas en el servidor en módulos incluidos por el mismo de la siguiente manera:

- server_functions : contiene la implementación de las funciones del producto
- socket_functions : funciones para enviar y recibir los datos
- estaciones: posee estructuras de datos para el manejo de la información de las estaciones meteorológicas
- parse_functions: funciones para obtener los datos del archivo

4.0.2. Cliente

- cliente_info
- socket_functions : funciones para enviar y recibir los datos

Para más información de la implementación del software ver la documentación en la carpeta "Docs"

5. Implementación y Resultados

Actualmente el software no se encuentra con los resultados acorde a los requerimientos del sistema. Las funciones del servidor estan implementadas en el servidor pero no se transfieren al cliente aun. Se puede ver la obtencion de datos en la siguiente figura:

6. Conclusiones

7. Apendice

Referencias

- [1] Douglas E. Comer, "Internetworking With TCP/IP,"*Vol I: Principles, Protocols, and Arquitecture*, pp. 473-480, Sixth Edition 2014.