Gabriel Kasten Altamirano

Daniset González Lima

CINVESTAV

Práctica # 1

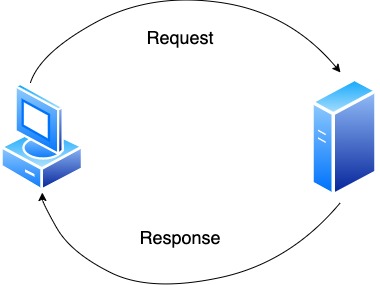
Conexión Cliente - Servidor

Sistemas Distribuidos

Dr. Felix Francisco Ramos Corchado

28/02/2020

La comunicación entre procesos a través de sockets se basa en la filosofía Cliente–Servidor. El cliente envía una petición y el servidor le envía al cliente una respuesta.

****

Es necesario que se utilicen puertos libres para establecer la comunicación entre las entidades, ya que existen algunos puertos reservados que no podrán utilizarse para otro propósito.

Algunos ejemplos de puertos reservados:

* ftp 20 y 21
* telnet 23
* SMTP 25
* Time 37
* www – http: 80

Rango de puertos:

* 1 – 255 Aplicaciones públicas
* 255 – 1023 Para determinadas aplicaciones que necesiten privilegios de super usuario.
* 1024 – 4999 Usados por procesos de usuario y del sistema
* > 5000 Usados solo por procesos de usuario

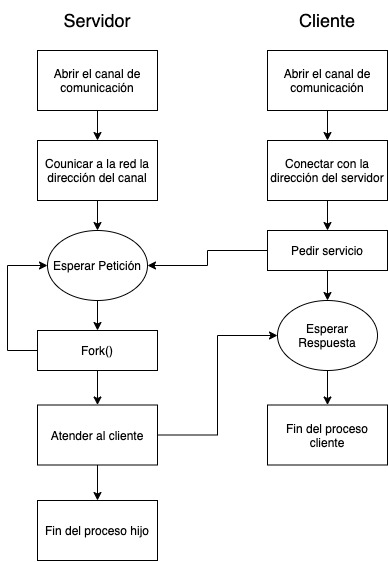
Para que un cliente pueda enviarle una petición al servidor que se encuentre dentro de la misma red, es necesaria la dirección IP del servidor y el número de puerto por donde se enviará la información, además, el cliente necesitará saber el puerto por el cual el servidor enviará la respuesta.

En esta práctica se utilizó el protocolo TCP (Transmission Control Protocol) y se utilizaron sockets para realizar la comunicación entre el cliente y el servidor. Esta herramienta de envío de información permite la comunicación bidireccional tanto en procesos ejecutados en la misma máquina como en procesos ejecutados en distintas máquinas.

Existen dos tipos de sockets:

* Socket de flujo: Al conectar se realiza una búsqueda de un camino libre entre origen y destino. Se mantiene el camino en toda la conexión.
* Socket de datagrama: No se fija un camino. Cada paquete podrá ir por cualquier sitio. No se garantiza la recepción secuencial.

En este caso, se utilizó un Socket de flujo, lo que indica que es un socket de TCP.

****

**Ejecución de la práctica:**

El proyecto consta de tres programas “client\_raspberry”, “server\_raspberry” y “server\_and\_client”, en una computadora se ejecuta el programa “server\_and\_client” que cuenta con la parte de cliente y de servidor, mientras que en la raspberry se deben tener los dos programas restantes, uno de cliente y otro de servidor.

**Pasos:**

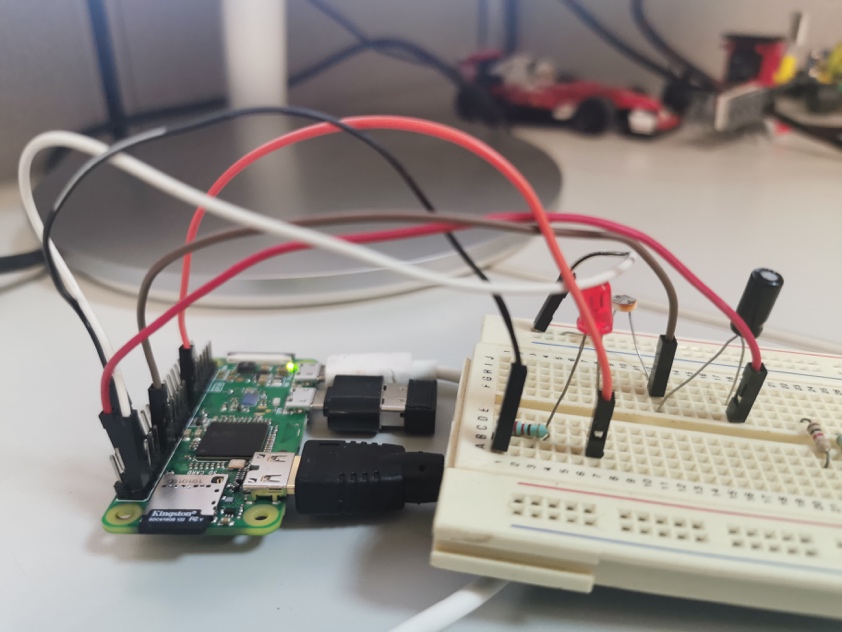
* Ejecutar el programa “server\_and\_client” en una computadora
* Ejecutar el programa de server\_raspberry” en la raspberry
* Ejecutar programa “client\_raspberry”

**Nota: Tanto la raspberry como la computadora con la que se comunicará deberán encontrarse dentro de la misma red. Es necesario que en los tres archivos se actualicen las direcciones IP, en los programas de cliente se deberá ingresar la IP del cliente, y en los del servidor la IP del servidor.**

**Descripción del funcionamiento:**

El objetivo de la práctica es la implementación de sockets para el envío y recepción de información entre un cliente y un servidor.

El cliente de la raspberry le enviará al servidor valores sensados por una fotoresistencia LDR, los cuales, recibirá el servidor y después actuará como cliente y según el valor de luz sensado le solicitará al servidor de la raspberry encender o apagar la luz (en este caso se representa mediante un led) según sea el caso. En caso de poca luz, se solicitará encender el led y en caso de mucha luz se apagará el led.

****

**Conexiónes:**

* Los valores de la fotoresistencia serán enviados al pin 11 de la raspberry.
* El circuito de la fotoresistencia se alimenta con 3.3V, los cuales se toman del pin 1.
* Mediante el pin 10 se enviará la instrucción de encendido o apagado del LED.
* Las demás conexiones son tierra física (GND) que puede ser tomada de distintos pines de la raspberry, en este caso se utilizaron los pines 6 y 25.

<https://docs.python.org/2/library/socketserver.html>

<http://sopa.dis.ulpgc.es/progsis/material-didactico-teorico/tema7_1transporpagina.pdf>

<https://www.speedcheck.org/es/wiki/tcp/>