

Reporte de práctica: Práctica 3
Vidal Sánchez José Antonio
Código:213496757
Ing. En computación

Descripción de la práctica

ACTIVIDAD DE APRENDIZAJE X

Título de la actividad: Servidor TCP con nombre de cliente

Objetivo: Implementar un servidor TCP

Fecha de entrega: Semana Y

Problemática

Mientras que la mayoría de los servicios consisten en conectarse a sistema y negociar datos, también es importante conocer y poder implementar el otro lado. En TCP hay una clara diferencia entre quién inicia una conexión, donde el extremo que inicia la conexión se llama comúnmente “cliente” y el extremo que espera las conexiones se llama normalmente “servidor”.

A pesar de esta diferenciación, una vez conectados, cualquier extremo puede enviar y recibir datos en cualquier momento y ya no importa quién es el servidor o el cliente. Incluso cualquiera puede cerrar la conexión en cualquier momento.

Para colocar una conexión a la espera de los clientes, se debe asociar una dirección y puerto a la escucha. Este número de puerto no puede ser elegido de forma aleatorio, ya existe un catalogo de puertos conocidos y rangos para cada fin. Por ejemplo, los primeros 1024 son reservados para el sistema operativo del 1 al 1023; del 1024 al 49151 son los puertos registrados, usables por las aplicaciones; y el resto, hasta el 65535 puertos dinámicos para conexiones salientes.

Las aplicaciones deben tratar de usar un puerto libre entre 1024 y 49151 y evitar utilizar otro puerto ya usado por otra aplicación. Aunque usar un puerto “ya conocido”, es poco recomendable, porque ambas aplicaciones van a tener un conflicto cuando ambas sean ejecutadas.

Detalles técnicos

Para poner un socket a la espera de conexiones entrantes, el procedimiento es ligeramente diferente. Primero se crea el descriptor de archivo con socket, luego se asocia un puerto y una dirección con bind. Seguido, se pone el socket a la escucha con listen. Para aceptar los clientes entrantes se utiliza accept.

Cuando se acepta una nueva conexión entrante, se crea otro descriptor de archivo que representa la conexión con el nuevo cliente. El socket a la escucha permanece abierto pudiendo recibir más clientes. Por cada cliente se producirá un nuevo descriptor de archivo.

Durante la llamada a la función accept, se puede saber la dirección y el puerto pasando los argumentos correctos. La estructura sockaddr puede ser convertida a una dirección con el formato “NNN.NNN.NNN.NNN” usando la función gethostbyaddr.

Para esta entrega, programa un servidor que ponga un puerto a la escucha (indicado por tu asesor), espere 1 cliente, y envíe una cadena de texto al cliente que se conecte. Recuerda cerrar ambos descriptors de archivos. También deberás imprimir la dirección IP del cliente que se conectó (en formato “NNN.NNN.NNN.NNN”) y el puerto de origen.

Para probar tu servidor utiliza el cliente TCP hecho en los entregables previos.

Reporte-Notas y dificultades:

El principal problema que encontre en la realizaci3n de esta pr3ctica fue el estado TIME_WAIT en el que entra el socket TCP tras cerrarlo. Debido a ello, una vez que cerraba tanto el servidor como el cliente, a3n as3 ten3a que esperar un tiempo para poder volver a correr mi programa servidor, pues aparece el error “*address already in use*”. Investigando, encontre que pod3a evitar dicho tiempo usando la funci3n *setsockopt* y la opci3n *SO_REUSEADDR*; de esta manera, puedo asignar la direcci3n y puerto a mi socket aunque se encuentre en TIME_WAIT.

Corregir este detalle me pareci3 importante, pues en caso de que un programa m3s grande se “*crashee*”, esto permite ponerlo en l3nea lo m3s pronto posible.