

Laboratorio N° 5

Interfaz de Socket

Protocolos de Comunicación TCP/IP

Grupo 4

Integrantes:

- Farber, Juan
- Flores Wittich, Pablo José
- Romano, Hugo Leandro
- Tadeo, Joaquín

1) ¿QUÉ TIPO DE SOCKET DECIDIÓ USAR PARA EL DESARROLLO DE LAS APLICACIONES? ¿POR QUÉ?

UD. EN SU CARÁCTER DE PROG. UNIVERSITARIO, LIC. EN INFORMÁTICA, ING. EN COMPUTACIÓN ACTUALMENTE ESTÁ TRABAJANDO EN UNA EMPRESA DEDICADA A DAR CAPACITACIONES A OTRAS EMPRESAS EN EL ÁREA DE INFORMÁTICA (COREL DRAW, AUTOCAD, PHOTOSHOP, ADOBE ILLUSTRATOR, Y SOFTWARE PARA EL MANEJO DE GRANDES IMÁGENES SATELITALES). RECIENTEMENTE, LO PUSIERON A CARGO DE LA COMPRA DE 25 EQUIPOS PARA EQUIPAR UN LABORATORIO, DESTINADO A LA CAPACITACIÓN DE LA EMPRESA EN LA QUE UD. TRABAJA. EL NUEVO SOFTWARE A ADQUIRIR DEBERÁ PODER TRABAJAR EFICIENTEMENTE CON UN HARDWARE PREEXISTENTE, CONSISTENTE EN 2 IMPRESORAS LASER HP 4050, 2 SCANERS AGFA CON CONEXIÓN SCSI, TAMAÑO A3.

Se decidió usar sockets de datagrama, utilizando el protocolo de transporte UDP. Las razones son las siguientes:

- **Simpleza:** Al ser UDP un protocolo no confiable y no orientado a conexión, la complejidad al desarrollar el código se vio reducida en gran medida.
- **Menor overhead en la comunicación:** Los paquetes UDP son más livianos que los paquetes TCP.
- **Velocidad:** Por lo expuesto anteriormente, el protocolo UDP permite una velocidad de transferencia superior a la del protocolo TCP.
- **No hay un gran intercambio de datos:** Decidimos no gastar recursos en el proceso de establecimiento de la conexión.
- **Aprendizaje:** Ya que no se profundiza mucho en UDP y la mayoría de los ejemplos están desarrollados para aplicaciones orientadas a conexión.
- **Broadcast:** Como se debe enviar el mismo mensaje a todos los hosts de la subred, deberíamos utilizar UDP como protocolo de transporte, ya que en TCP esto no sería posible.

2) ¿QUÉ TIPO DE SERVIDOR DECIDIÓ DESARROLLAR? ¿POR QUÉ?

Se decidió desarrollar un servidor iterativo. Las razones son las siguientes:

- **La complejidad de configurar un servidor iterativo es menor,** ya que no hay que manejar la concurrencia de procesos.
- **El servicio solicitado utiliza una pequeña fracción del tiempo para su procesamiento:** Solo debemos mandar un datagrama, que lleva un mensaje de texto encapsulado.
- Aunque no es una norma, **los servidores iterativos suelen ser asociados con los sockets de datagrama.**
- **El mismo y único proceso servidor entrega los mensajes al cliente.**

3) ¿QUÉ RUTINAS DE ENVÍO Y RECEPCIÓN DE MENSAJES USÓ? ¿POR QUÉ?

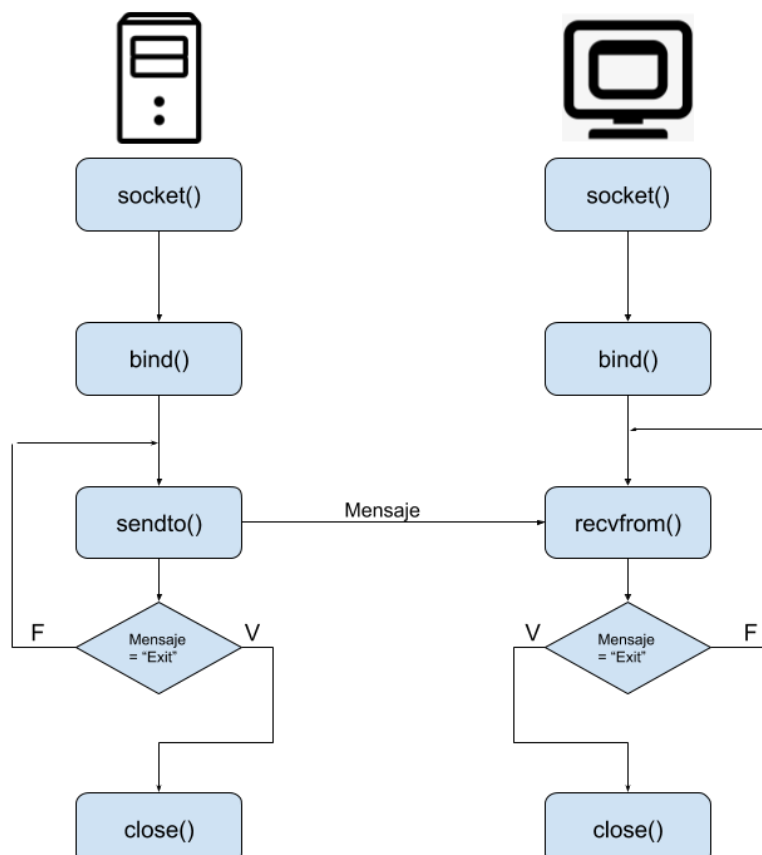
Las rutinas utilizadas para el envío y recepción de mensajes fueron las apropiadas a sockets de datagramas, ya que ellas proveen como parte del argumento a las direcciones y puertos necesarios para la comunicación entre el servidor y el cliente. Ellas son:

- **sendto():** Para enviar los mensajes desde el servidor.
- **recvfrom():** Para recibir los mensajes del lado del cliente.

4) DESCRIBA LAS DIFICULTADES QUE TUVO AL DESARROLLAR EL LABORATORIO.

- **Coyuntura actual:** Nos resultó difícil ponernos de acuerdo para trabajar en el ambiente virtualizado de manera remota, distribuirnos las tareas y mantener los archivos actualizados con los cambios propuestos por cada uno.
- **Material guía:** Encontramos poco material para el desarrollo de la aplicación utilizando UDP. Esto nos llevó a relacionar los conceptos adquiridos a lo largo de la materia, de una manera lenta e incremental.
- **Codificar primero, luego diseñar:** Llegó un momento en el que, al integrar las distintas porciones de código propuestas por cada uno, el coste de entenderlo era mayor que descartar lo realizado. Entonces se decidió empezar de nuevo centrándonos en el diseño inicial, más que en el código en sí.
- **Mala interpretación del enunciado:** Una vez “terminada” la aplicación, nos dimos cuenta que habíamos interpretado al revés los roles. En efecto, pasamos horas tratando de resolver problemas inexistentes.
- **Envío de un servidor a múltiples clientes:** Siguiendo el punto anterior, una vez que “finalizamos” la aplicación, nos dimos con que el servidor debía enviar el mensaje a todos los clientes de la subred, por lo que tuvimos que adaptar el funcionamiento del servidor para que envíe un mensaje broadcast y no unicast.
- **Control de errores trasladado a la aplicación:** Al ser UDP no confiable, el control de errores no es algo de lo que nos podemos abstraer.

DIAGRAMA DE FLUJO



BIBLIOGRAFÍA

- [1] Saade, S - Protocolos de Comunicación en Internet - Edunt - 2016.
- [2] Saade, S y Borelli, J. - Programación en Redes TCP/IP Utilizando Interfase de Sockets
- [3] Albaca Paravan, C, Saade, S y Lutz, F - Programación distribuida con interfaz de socket en IPv6.
- [4] IBM Knowledge Center - IBM. Consultado 15/11/2020.
https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.hala001/itcsocketconselsoc.htm