Tarea Programada I

Lenguajes de Programación

Instituto Tecnológico de Costa Rica
Administración de Tecnologías de Información
Prof: Andréi Fuentes L.

Estefany Quesada Montero **200940160**
Esteban Aguilar Valverde **200969856**
Esteban Mora Soto **201115237**

Descripción del problema

Se debe realizar una comunicación entre dos computadores por medio de la investigación de sockets. La comunicación se divide en dos procesos, uno va a recibir mensajes mientras que el otro los va a mandar, utilizando un socket para cada uno de los procesos.

Para que se puedan comunicar los dos computadores y entablar una conversación se necesitara que cada uno tenga el programa y de esa forma se van a ir intercalando los proceso de recibir (proceso servidor) y el de enviar (proceso cliente) mostrándose en pantalla los mensajes que se están enviando entre ellos. A la hora de escribir la palabra "Adiós" se deberán cerrar los sockets.

Para el programa se solicito realizar ciertas funciones previas al programa.

1- Primeramente se debía generar en un programa dos funciones una función que tomara un arreglo de caracteres y enviarlo a otra función la cual retornaría el mensaje y una confirmación de recepción
2- En segundo lugar era generar sockets de cliente y servidor y conectarlos entre si ya sea por una o mas computadoras estos sockets en el lado del cliente recibiría un arreglo de caracteres y los enviaría al servidor el cual enviaría un recibido y luego se cerraría la conexión.
3- Luego seria adaptar los sockets anteriores para que recibieran y enviaran más de un mensaje para generar una conversación de cliente servidor hasta que se enviara la palabra "Adios".
4- Final mente el programa el proyecto seria adaptar cliente y servidor en el mismo archivo y un proceso fork adjunto que generara el cambio de cliente a servidor y viceversa donde dos computadores tendría el archivo en C y el cliente de un ordenador se conectaría con el servidor del otro programa y viceversa para generar una conversación tipo chat entre dos ordenadores.

Librerías Usadas

stdio.h : Para utilizar funciones de entrada y salida.

stdlib.h : Para utilizar funciones para control de procesos (exit).

unistd.h: Funciones para control de procesos.

string.h: Funciones para trabajar con cadenas de caracteres como strlen.

sys/types.h: Permite utilizar diferentes tipos de datos.

sys/socket.h: Permite utilizar funciones como socklen_t.

netinet/in.h: Familia de protocolo de internet.

netdb.h: Para realizar operaciones en red.

Diseño del programa.

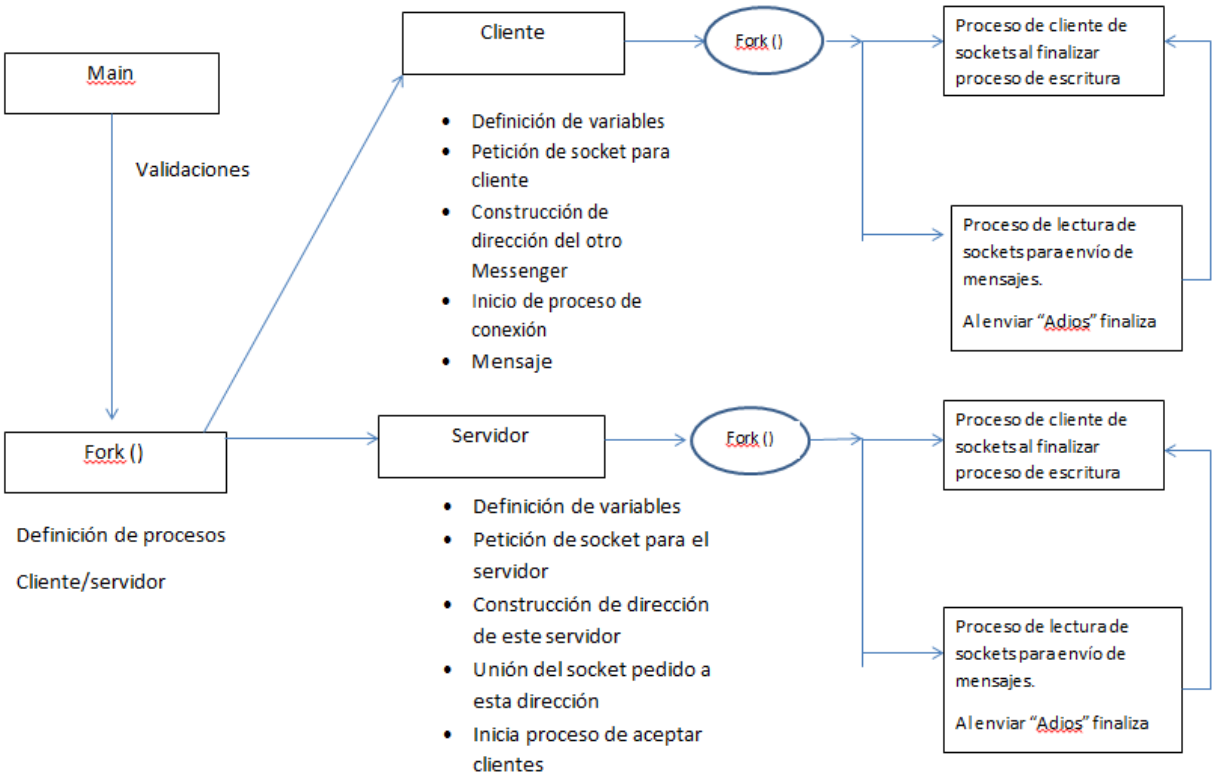
Para este programa decidimos los siguientes aspectos.

- Un solo fork() para todo el programa (se pensó en varios para separar el envió de la recepción de mensajes, el cliente del servidor y otro para separar el estado de continuidad del programa del sistema de cerrado de los sockets pero fue complicado e ineficiente).
- En la parte de los sockets se decidió generar verificaciones para saber en que lugares estaba fallando algunos puntos como si existían los puertos o si no estaban y ocupados además de advertencias en el envió y recepción en caso de perder la conexión o denegar la aceptación de un cliente.
- Al momento de envió y recepción de mensajes se decidió generar encabezados para cada uno y avisos de si se logro enviar o recibir el mensaje.
- También al momento del chat se decidió generar un bucle que revisara el estado de la conversación esto para cerrar los sockets en caso de enviar la palabra “Adios”.
- Para servidor se decide eliminar la verificación de si se introduce información ya que el cliente verifica esto desde un inicio.
- En servidor también se crea un bucle para revisar el estado del mismo para poder comparar lo que recibe y compararlo con la palabra de terminación “Adios” esto para saber si se deben cerrar los sockets.

Algoritmos usados

- Un algoritmo error que recibe un puntero a un char esta función envía un mensaje de error especificando el error producido.
- Un bucle de espera de parte del cliente con un tiempo de funcionamiento de no mas de un minuto para esperar a que exista un servidor habilitado, en caso de no existir servidor después de ese tiempo cierra la conexión del cliente.
- Funciones generales que utilizan los sockets en lenguajes c (basados en la investigación de sockets en este lenguaje) como conect, socket, accept, bind entre otras.
- Un bucle para el estado de la conversación después de conectados los sistemas, esto para saber cuando se debían cerrar los sockets.

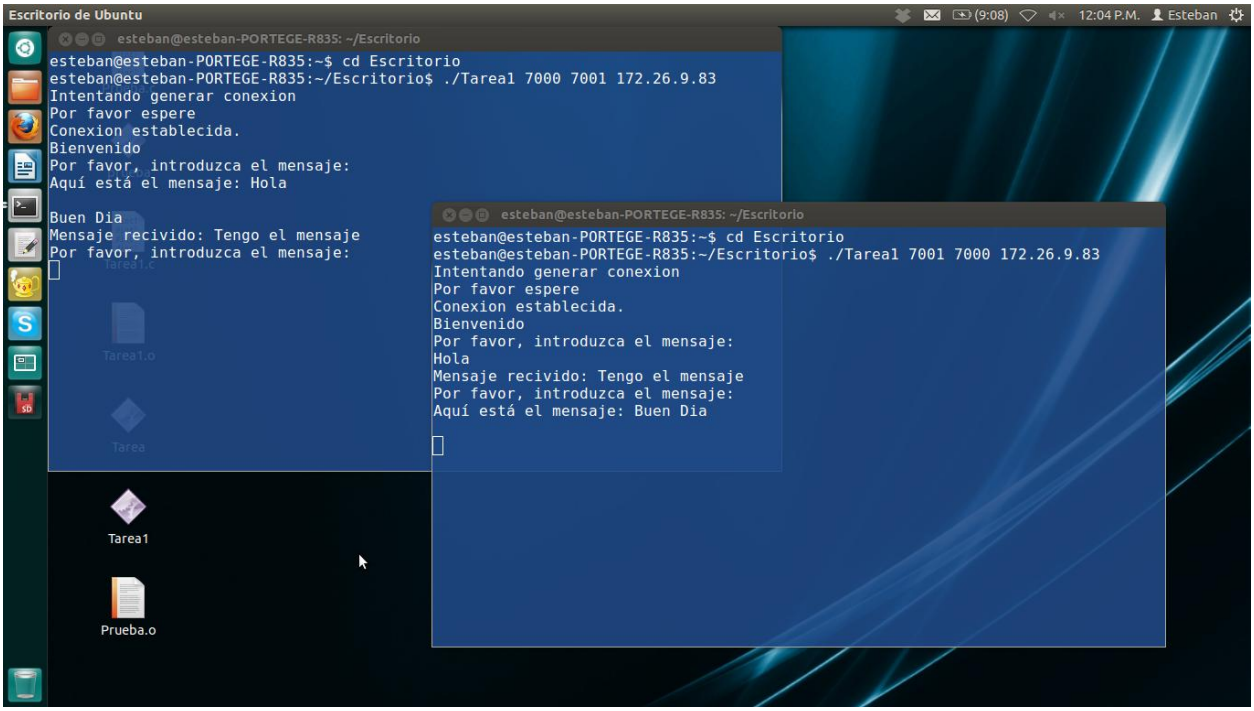
Diagrama del proyecto



Manual de usuario

- Primeramente de debe abrir una terminal (dos en caso de intentarlo dentro de la misma computadora).
- Ir a la dirección en la que se encuentra el programa EJ: cd Documentos/C/Progra
- Compilar el programa mediante gcc <Nombre del archivo .c> -o <Nombre del archivo de salida o exe>
- Ejecutar archivo con el nombre del archivo el puerto de nuestro servidor el puerto de cliente y la IP del cliente Ej. /<Archivo de salida> <Puerto del servidor> <Puerto del cliente> <IP del cliente> similar a esto [./Prueba 3478 3479 192.168.1.244]
- Esperar a que se conecten los computadores
- Iniciar la conversación hasta que algún usuario envié "Adios".

Imagen de ejemplo



Análisis de Resultados

Se crearon dos procesos, uno que envía una hilera de caracteres al segundo proceso y este le responde con otra hilera. Luego se crearon 2 programas, programa cliente el cual puede mandar mensajes al otro programa llamado servidor, y este desplegar un mensaje que ya recibió el mensaje.

Se modificaron los programas para crear un solo programa con dos sockets uno para enviar mensajes y otro para recibirlos y el programa utiliza cada uno de ellos mediante el uso de un *fork* para cambiar entre los dos procesos.

El programa permite entablar una conversación con otro computador que tenga el mismo programa. La ejecución finaliza en l momento en que se digite “Adios” y de este modo se cierran los sockets.

Conclusión Personal

Se hicieron varias investigaciones para entender el funcionamiento de sockets en C, para poder realizar una comunicación entre dos computadores. También se investigo sobre el uso de algunas librerías que se son necesarias para el buen funcionamiento de cada uno de los procesos dentro del programa, también para la utilización correcta de sockets y para poder realizar operaciones en red por ejemplo. Se adquirió conocimiento acerca de unir dos procesos diferentes mediante *forks* para poder ser implementados en un solo programa y así lograr que cada programa en el computador pueda recibir y enviar mensajes por medio de los sockets.