DOCUEMENTACIÓN EXTERNA

Jurgen Arley Elizondo, Ignacio Aguilera Mora, Luis Eladio Castro Montero

INSTITUTO TECNOLÓGICO DE COSTA RICA Lenguajes de programación

02/09/2014

Contenido

Descripción de problema

Se debe realizar un programa que sea de mensajería instantánea como lo es el chat de Facebook, whatsapp,etc. En este programa se debe realizar la conexión entre dos computadoras por medio del programa C, y este se debe conectar con un formato cliente servidor cliente.

El programa tiene las siguientes funciones

- **Registrar el usuario:** los usuarios se registran en el servidor central. Este obtiene la dirección de la IP del usuario de forma automática, el usuario no especifica nada como por ejemplo en Facebook Messenger. Con respecto al puerto, siempre se usará un valor predeterminado que estará especificado en un archivo de configuración, pero ese valor podrá ser modificado por los usuarios, sin necesidad de recopilar el código.

- **Enviar mensajes:** El programa permite a los usuarios enviar mensajes a alguno de los contactos, para lo cual se especificará el nombre de usuario, y el mensaje, y el programa envía el mensaje ha dicho usuario, a través del servidor. El mensaje podrá ser de texto. Cuando los mensajes son enviados, estos están encriptados y luego cuando se reciben se des encriptan.
  Cuando o puede ser un archivo y el usuario especifica el archivo por medio de la ruta completa del mismo. Los mensajes, tanto de texto como archivos, no serán enviados de un usuario directamente a otro usuario, los archivos también serán enviados por cliente servidor cliente.

- **Recibir mensajes:** El servidor no utiliza el fork ya que se usó una librería llamada lpthread que usa hilos para la recepción de información.

Diseño de programa

La elaboración de este programa es basado en el lenguaje de programación C, para que este compile, deben haber 5 programas:

1. **Server.c**, este tiene toda la información del servidor y utiliza la librería –lpthread para hacer la bifurcación de envío y recibido de mensajes.
2. **Client.c** este tiene toda la información de los clientes y también usa esa librería para poder enviar la información al servidor y recibirla de este.
3. **config_client.txt**, este tiene toda la información de red del servidor como lo es la IP y además el puerto por donde se quiere conectar y aquí se guarda la información de los usuarios.
4. **Config_server.txt,** este tiene información como lo es el puerto por donde debe el servidor recibir y enviar los mensaje.
5. **User.txt,** este tiene la información de los clientes que se han conectado al chat.

Estructura de Cliente y Servidor del programa

Colores: Los colores utilizados en este programa son, rojo, verde, amarillo, azul, morado, celeste y se definen con un #define “nombredelcolor” “x1.[….”

Cliente: El programa realiza variables inicializando para almacenar los nombres de usuarios y además el mensaje que será enviado, y crea variables como por ejemplo char mensaje [250] y esta almacena el mensaje que será enviado y encriptado al servidor. Luego abre el config_client.txt y obtiene la dirección del ip y el puerto. Luego realiza la conexión que se debe hacer con el servidor el cual ya debió haber sido inicializado, este crea el socket y recibe el dominio, protocolo y el tipo del socket.

Se establece la estructura de la dirección e indica al socket donde debe conectarse, después convierte el número de puerto para la orden de red-

Se solicita al usuario el nombre de usuario y la contraseña respectiva por medio del comando fgets(“se pone información para el usuario”), cuando este cumplió con las normas envía la contraseña a un socket por medio de la función send.

Realiza la lectura del archivo que tiene los usuarios y utilizó la función fopen(), y ahí muestra los usuarios en línea y además los que ya han sido conectados.

Luego le dice al usuario que digite el mensaje que desea enviar pero en la variable donde se va a guardar la información se realiza un bzero() para formatear la variable.

Luego se encripta el mensaje el cual es encriptado por medio del cifrado de Cesar, este se encripta y luego a la variable se le concatena el nombre del usuario del cual fue enviado, el mensaje, y el usuario al cual va dirigido.

La función recibir() los que hace es recibir el socket que le envía el servidor y luego lo obtiene le variables y luego se des encripta.

Servidor: Como primera instancia el servidor empieza leyendo el txt de los usuarios para poder saber quiénes pueden estar conectados y quiénes no.

Luego lee el archivo config_server.txt pero para poder obtener el puerto de la configuración y este así dejar abierta esta puerta para que cuando el client.c compile pueda conectarse con el servidor a partir de del puerto que este abrió.

La función server() es la que recibe el socket que le envía el cliente para poder comunicarse con el cliente 2 , esta función vuelve a verificar la existencia de los usuarios. Se utilizaron go to: en etiqueta para poder manejar las órdenes que se le desean enviar al computador.

Este servidor tiene la función de recibir información y concatena variables para poder enviarlo a sus usuarios con el texto unido por ejemplo Juan:dgljsng8|Pedro, se debe saber que el mensaje siempre ha estado encriptado y continuará hasta que el cliente lo reciba.

Librerías utilizadas:

Las librerías son archivos de la biblioteca estándar en C que contienen un conjunto de funciones, tipos relacionados y macros, y estos son proporcionados para facilitar la implementación de los programas. Todas las librerías son declaradas como archivos cabeceras. Para la creación de este programa de mensajería se utilizaron las siguientes:

- **#include <stdio.h>=** Contiene las definiciones de macros, constantes, declaraciones de funciones y definición de tipos, usados por operaciones estándar de entrada y salida.
- **#include <stdlib.h>=** Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras.
- **#include <unistd.h>=** Concede acceso a POSIx del api del Sistema operativo
- **#include <string.h>=** Es un archivo de la Biblioteca estándar del lenguaje de programación C que contiene la definición de macros, constantes, funciones y tipos de utilidad para trabajar con cadenas de caracteres y algunas operaciones de manipulación de memoria
- **#include <errno.h>=** Definen las macros que presentan un informe de error a través de códigos de error.
- **#include <sys/types.h**>= Definen las macros que brindan informes de los sockets
- **#include <sys/socket.h=** Librería que permite la comunicación entre dos programas o procesos. Contiene las funciones de sockects, y sus correspondientes constantes.
- **#include <netinet/in.h>=**Permite usar la variable errno.
- **#include <arpa/inet.h>=** Pone a disposición del tipo in_port_t y el tipo in_addr_t tal como se define en la descripción de <netinet / in.h> .
- **#include <pthread.h>= Es** la librería que permite usar hilos para la creación del socket y que birfurque.
- **#define MAX 250=** color**.**
- **#define Rojo "\x1b[31m**"= color.
- **#define Verde "\x1b[32m**"= color.
- **#define Amarillo "\x1b[33m**"= color.
- **#define Azul "\x1b[34m**"= color.
- **#define Morado "\x1b[35m**"= color.
- **#define Celeste "\x1b[36m**"= color.
- **#define ResetColor "\x1b[0m**"= color**.**

Análisis de los resultados

Para la implementación de este chat, se logró realizar la conexión cliente servidor cliente el cual envía mensajes pero este también envía archivos, la diferencia es que esto se realiza por aparte debido a diferentes problemas presentados durante el desarrollo del programa.

El sistema de mensajería sirve muy bien sin embargo presenta varias imperfecciones que si no se está consiente de estas el programa presentara errores, por ejemplo el mayor inconveniente que se nos presento fue el archivos de los usuarios, ya que este es utilizado por el cliente y el servidor lo que obliga a estos 2 archivos estar juntos en una carpeta para la correcta ejecución del programa, además de esto los usuarios solo se almacenan mientras el servidor este corriendo ya que cada vez que se abre se reinicia la lista de contactos. Otro problema es que debido a esto el primer cliente que se conecte no le aparecerá contactos y deberá esperar a que alguien más se conecte para así poder iniciar una conversación.

Otro de los principales problemas fue que no se pudo unir la parte de archivos al chat por lo que es un programa totalmente aparte.

Manual de usuario

Para poder ejecutar este programa es necesario que el sistema operativo del equipo en el que se encuentre sea Linux, ya que en otro no servirá. Una vez que se esté trabajando en Linux se procede a abrir la terminal, para esto puede utilizar el comando *CTRL+ALT+T,* luego con el comando *cd* se accede a la carpeta donde se encuentren los programas, por ejemplo *cd Desktop, cd Tarea_Programada.* Una vez que se accede a la carpeta desde la consola puede utilizar el comando *ls* para verificar los archivos que se encuentren en esta.

En caso de que los archivos no se encuentren compilados se debe usar el siguiente comando para el servidor y el cliente respectivamente: *gcc server.c –o server –lpthread, gcc client.c –o client –lpthread.* Una vez hecho esto ya se podrá ejecutar los programas pero se recomienda antes revisar los archivos *config_server.txt y config_client.txt* para asegurarse que la dirección de la IP sea la correcta y que los puertos sean los mismos.

Lo primero es ejecutar el servidor para esto se introduce en la consola *./server,* luego si se va a correr en el mismo equipo se deben abrir dos consolas más y dirigirse a la carpeta donde se encuentre el programa siguiendo los pasos mencionados anteriormente, una vez que se esté ahí se inserta en la consola ./client, esto en ambas. Luego de esto el programa le solicitara un nombre de usuario y una contraseña, en caso de que nunca haya usado el chat el mismo programa lo agregara a la lista de contactos, si es un usuario registrado verificara su contraseña. Posteriormente le preguntara con quien desea chatear, luego de elegir debe esperar a que el otro cliente lo seleccione a usted antes de enviar un mensaje. Cuando ambos han seleccionado con quien hablar se puede proceder a hablar, los mensajes que un usuario escribe se verán de color rojo, y los mensajes recibidos de color azul, para finalizar la conversación se debe enviar el mensaje “adiós”.

Por problemas de desarrollo es envio de archivos se maneja aparte del chat, para esto se deben realizar los primero pasos para acceder a la carpeta donde se encuentren los programas *archivos_clientes y archivos_servidor,* el método de compilación es más sencillo ya que se omite la sentencia –lpthread, ejemplo: *gcc archivos_clientes.c –o archivos_clientes, gcc archivos_servidor.c –o archivos_servidor.* Antes de enviar un archivo es importante que tenga el archivo a enviar en la misma carpeta del cliente. Cuando se ejecuta el programa este le solicita el nombre del archivo a enviar y este llegara a donde se encuentre el servidor.

Conclusión

La elaboración de esta tarea programada ayudó a que el equipo de trabajo conociera más acerca de las herramientas de programación que posee Linux, así como el entendimiento de código en lenguaje C. Los temas principales para búsqueda de información fueron el uso de sockets así como el manejo de archivos en C.

Uso de sockets: Se buscó información del uso de sockets, entendiendo con esta, el funcionamiento de las IP´s y puertos para la conexión entre dos usuarios Además, se aprendió del uso de los de los hilos para bifurcar la señal del servidor.

El manejo de archivos en C: El grupo de trabajo aprendió información de cómo se gestionan los archivos en c, lo que es leer y escribir.

Encriptación y des encriptación: Se aprendió a encriptar y des encriptar archivos y la información antes y después de enviarla al servidor para que esta información no sea robada.