



Tarea Programada 1

Elizabeth Córdoba

Eduardo Álvarez

Andrés Abarca

Josué Fallas

Marianela Cordero

10 de septiembre del 2013

Tabla de Contenidos

Descripción del problema.....	02
Diseño del programa.....	03
Librerías usadas.....	04
Análisis de resultados.....	06
Conclusion personal.....	07

Descripción del problema

La primera tarea programada tiene como objetivo implementar un clon de WhatsApp, utilizando el lenguaje de programación C y el sistema operativo Ubuntu.

El proyecto se divide en 3 secciones, específicamente:

1. Agregar amigos: en esta parte el usuario puede agregar nuevos amigos, señalando su username, su puerto y por ultimo su IP.
La información anterior será almacenada en un archivo de texto, esto para asegurarse que cada vez que se ejecute el programa la información esté disponible.
2. Enviar mensaje: un usuario puede enviarle un mensaje a cualquiera de sus amigos con solo especificar su username, el programa entregará el mensaje al IP y al puerto que pertenecen a dicho amigo.
3. Recibir mensajes: el usuario puede recibir mensajes.

Para llevar a cabo cada una de las partes mencionadas anteriormente se debe investigar el uso de sockets en Linux, esto permite el intercambio de mensajes entre computadoras. También se requiere de la implementación del fork (), para bifurcar el programa en dos procesos (recibir y entregar mensajes).

Asimismo se deben utilizar colores diferentes para especificar cuáles son los mensajes recibidos y cuáles son los enviados.

Diseño del programa

El programa debe cumplir con tres funcionalidades: agregar amigos, enviar mensajes y recibir mensajes.

Se decidió realizar un menú para que el usuario decida qué actividad desea realizar, el menú ofrece: 1 para agregar amigos, 2 para enviar mensajes y 3 para salir del programa.

En agregar amigos, el usuario debe ingresar el username, el ip y el puerto del contacto, estos se almacenan y se escriben en un archivo .txt, el cual se llama "Amigos.txt".

La información de los contactos se maneja mediante una estructura de datos. Es decir, primero se lee el archivo creado en agregar amigos, luego estos datos se almacenan en memoria mediante una lista simple, la cual es en realidad una lista de punteros de tipo struct NodosLista. Se tomó la decisión de utilizar una lista simple para poder realizar una búsqueda por medio del username, una vez que se encuentra el puntero que corresponde a ese username se puede extraer la ip y el puerto de ese username. Estos datos son necesarios para poder enviar un mensaje.

Después seguimos con la parte de enviar un mensaje, esta funciona con los sockets, que se conecta a un servidor y envía el mensaje al contacto existente, si el amigo o contacto no existe entonces nos mostrará un error en la aplicación, además cabe destacar que el servidor recibe el número de puerto que está disponible para establecer la conexión, el cliente será el que recibirá la dirección IP (del servidor) y el mismo puerto que estableció el servidor.

Librerías Utilizadas

- **Stdio:** esta es la librería estándar de C, involucra constantes, definiciones de macro, definiciones de funciones y definiciones de tipos que utilizan las operaciones de entrada y salida.
- **Stdlib:** esta librería se divide en tres categorías específicamente
 - Funciones aritméticas: `abs ()`, `div ()`, `Long int labs()`.
 - Funciones números aleatorios: `rand ()`, `srand()`.
 - Funciones conversión de cadenas: permite pasar cadenas de caracteres a números enteros, enteros largos y flotantes. En el clon de WhatsApp se utilizó la función `atoi` para convertir el IP y el puerto a números enteros.
- **String:** se encarga del manejo de caracteres, utilizando funciones como:
 - `Strcpy`: copia la cadena de caracteres del segundo parámetro al primero.
 - `Strcmp`: compara las cadenas de caracteres, devolviendo un cero si son iguales. Esta función se utilizó para saber si el nombre del usuario al que se desea enviar un mensaje existe.
- **Unistd:** esta es una biblioteca que tiene incorporado la función `fork()`, la cual se encargará de la bifurcar el programa en dos procesos iguales (enviar y recibir mensajes).

➤ **Sys/Socket y Sys/types:** esta librería se utiliza para permitir la interacción entre computadoras, utilizando los sockets. Involucra funciones como:

- `Bind()`: se encarga de la asociación de los puertos y la especificación de sus propiedades.
- `Listen ()`: especifica que un usuario desea conectarse con la computadora propia.
- `Socket ()`: permite la comunicación entre las computadoras.
- `Accept ()`: acepta y solicita conexiones.

➤ **Netinet:** familia de direcciones de internet, permite almacenar los protocolos de internet.

Análisis de resultados

A continuación se especificaran los objetivos que se cumplieron en el proyecto programado 1:

- Se lo logró implementar la sección de agregar amigos. En este caso el usuario digita el username, la IP y el puerto de la persona que desea agregar a su lista de amigos. Cada uno de los datos ingresados serán guardados en un archivo txt, el cual será cargado en memoria por medio de una lista simple; la cual se manipulará para obtener la información necesaria.
- Por otra parte el programa permite conectar las computadoras, por medio de sockets (cliente y servidor), dando lugar a la transacción de entrega y el recibimiento de mensajes.
- El mensaje enviado aparece de color azul y el entregado verde.
- También tiene implementado el fork (), lo cual permite bifurcar el envío de mensaje, así como la entrega de mensajes.

Objetivos que no se cumplieron:

- No es robusto, dado que cuando se presenta algún error de conexión el programa deja de funcionar (se sale del programa). No se pudo cumplir con este objetivo por falta de tiempo.
- No muestra el nombre de la persona que envía el mensaje o el nombre de la persona a la que se le envía el mensaje. Al igual que el punto anterior no se pudo cumplir con este punto por falta de tiempo.

Conclusión personal

En la Tarea Programada I, pusimos a prueba todos nuestros conocimientos sobre programación que hemos adquirido en otros cursos, además enfrentarnos al reto de aprender a programar en un lenguaje del cual no teníamos experiencia. Al aprender a utilizar el lenguaje de programación C también empezamos a trabajar con el sistema operativo Linux, específicamente con Ubuntu.

Durante el desarrollo de esta tarea adquirimos nuevos conocimientos mediante la investigación de funciones, procesos, estructuras de datos y manejo de información en C. Entre las funciones que usamos están los sockets y el manejo de direcciones IP y puertos, los cuales eran necesarios para poder crear la conexión entre dos computadoras para el envío y la recepción de mensajes, mediante la creación de un cliente y un servidor. La función fork, la cual nos permite bifurcar procesos.

Asimismo se obtuvieron conocimiento en el manejo de listas simples en C y como manipular los archivos txt.

En el desarrollo de esta tarea, reforzamos nuestra habilidad de trabajo en equipo ya que al trabajar en un grupo de 5 personas se dificulta un poco poder organizar las reuniones y poder adaptar los distintos puntos de vista de manera que se combinen de la mejor manera para obtener el resultado esperado. Pero al poder concluir exitosamente esta tarea, logramos superar la prueba y adquirimos experiencia tanto en programación como en habilidades interpersonales, para saber trabajar en equipo.

Bibliografía

- R, S. & Pozo, S. (2003). *Librerías C* Recuperado de <http://www.disa.bi.ehu.es/spanish/asignaturas/ii/LibreriasANSI-C.pdf>.
- Libre (s. f.). Recuperado de <http://www.fismat.umich.mx/mn1/manual/node15.html>.
- Manual C (s. f.). Recuperado de <http://www.fismat.umich.mx/mn1/manual/node18.html>.
- Estructuras y funciones de programación de sockets (s. f.). Recuperado de <http://informatica.uv.es/it3guia/ARS/practicas/Funciones.pdf>.
- Tutorialspoint (s. f.). *Fprintf() - C Library Function Example* Recuperado de http://www.tutorialspoint.com/c_standard_library/c_function_fprintf.htm
- Leer un archivo de texto en C?? - Yahoo! Respuestas (s. f.). Recuperado de <http://espanol.answers.yahoo.com/question/index?qid=20070510163447AAMTNdS>
- Funciones básicas para listas simplemente ligadas en C | Código Fuente (s. f.). Recuperado de <http://salomonrt.wordpress.com/2011/08/09/funciones-basicas-para-listas-simplemente-ligadas-en-c/>
- Cómo detectar los puertos abiertos en nuestra computadora o router (s. f.). Recuperado de <http://blog.desdelinux.net/como-detectar-los-puertos-abiertos-en-nuestra-computadora-o-router/>
- TCP/IP Programming in C - YouTube (s. f.). Recuperado de <http://www.youtube.com/watch?v=V6CohFrRNT0>

- ¿Cuáles son mis puertos abiertos en Ubuntu? (s. f.). Recuperado de <http://es.kioskea.net/faq/7056-cuales-son-mis-puertos-abiertos-en-ubuntu>
- Ubuntu (s. f.). Recuperado de <https://lists.ubuntu.com/archives/ubuntu-es/2007-July/025317.html>
- Sockets en C de Unix/Linux (s. f.). Recuperado de http://www.chuidiang.com/clinix/sockets/sockets_simp.php
- Socketss (s. f.). Recuperado de <http://www.pablin.com.ar/computer/info/sockets/index.htm>
- Gethostbyname(), gethostbyaddr() (s. f.). Recuperado de <http://beej.us/guide/bgnet/output/html/multipage/gethostbyname-man.html>
- Socket Programming (TCP) using C: | Prof. Vinod Pillai (s. f.). Recuperado de <http://vinodthebest.wordpress.com/2012/05/26/socket-programming-tcp-using-c/#comments>
- IANA ¿ IANA-managed Reserved Domains (s. f.). Recuperado el 10 de Septiembre del 2013, de <http://www.iana.org/domains/reserved>