

El objetivo del programa es la implementación de un Sistema de Mensajería escrita en el lenguaje de programación C, con el cual se conoce como implementar sockets y con ello lograr la comunicación y envío de mensajes y archivos por medio de diferentes computadoras.

Tarea Programada 1

Programa de Mensajería

Yaxiri Azofeifa García
Kenneth Céspedes Garbanzo
Carolina Robles Uribe

Tabla de contenido

Descripción del problema.....	2
Métodos utilizados.....	2
Socket.....	2
TCP/IP(Transmisión Control Protocolo).....	2
Método Fork	2
Diseño del programa	3
Decisión del diseño.....	3
Algoritmos usados (Funciones principales del programa)	3
Diagrama Lógico	5
Librería de C utilizadas	6
Stdio.h	6
Stdlib.h	6
Unistd.h.....	6
String.h	6
Sys/types.h	6
Sys/socket.h	6
Netinet.h	6
Netdb.h	7
Análisis de resultados.....	7
Objetivos alcanzados.....	7
Objetivos no alcanzados.....	7
Manual de Usuario	7
Conclusión	8

Descripción del programa

Al implementar un sistema de información a nivel computacional, es de suma importancia el manejo correcto de datos, con el cual se cubren las necesidades requeridas en dicha implementación. Además de esto, se debe tomar en cuenta el control eficaz de dicha aplicación. Esta tarea está desarrollada bajo el entorno y las reglas descritas por el Lenguaje Imperativo C.

Las especificaciones y funcionalidades a implementar en la tarea en la cual viene detallada nos llevan a desarrollar un programa para la comunicación entre dos o más nodos pertenecientes a una red específica, con la cual se logrará el envío de mensajes y archivos entre diferentes usuarios.

Para la comunicación de los mismos, se debió usar el concepto de sockets, que nos permite una mayor facilidad del intercambio de datos, pero deben tomarse en cuenta ciertos requisitos y características específicas para que los programas puedan comunicarse entre sí, con la modalidad de cliente-servidor con el que se utiliza TCP orientado a la conexión.

La comunicación entre los usuarios debe tener la facilidad de recibir mensajes o archivos y al mismo tiempo poder enviar los mismos, y poder tener conexión con uno o más nodos de la red. La función que permite lo descrito anteriormente es llamada *fork()*, la cual se encarga de copiar el mismo programa en dos distintos procesos, diferenciando a cada uno con un número de identificación, que nos permite crear procesos padres e hijos. Este proceso nos permite una bifurcación en el hilo de ejecución en un entorno multi-hilo.

Métodos estudiados:

Socket: Es la herramienta que ofrece a los sistemas operativos para la comunicación entre diferentes procesos de flujo de datos, estos constituyen un mecanismo de entrega de paquetes de datos provenientes de una tarjeta de red a los procesos apropiados. Para mantener un buen control de socket es importante el manejo de ciertos requisitos para que la comunicación pueda darse con éxito, por ejemplo: conocer la IP de ambos programas y el puerto del mismo.

TCP/IP (Transmisión Control Protocolo): Esta es una red con el protocolo de que cada máquina está identificada por medio de una dirección IP que debe ser única. Sin embargo en cada máquina se encuentran múltiples procesos ejecutándose simultáneamente. Los procesos deben de ser asociados al número de puerto para identificar los paquetes que reciba la máquina. La conexión TCP tiene tres etapas: Establecimiento de conexión, transferencia de datos y fin de conexión.

Método Fork(): Este nos permite crear una copia del programa, generando un proceso hijo y uno padre, los cuales serán ejecutados al mismo tiempo bajo un diferente PID. Esto lo permite lo que permite es crear un proceso encargado de realizar el envío de archivos hacia el servidor remoto y al mismo tiempo ejecutar el proceso de recepción el cual se encontrara activo hasta que una señal de recepción de archivo, así el mismo programa podrá realizar las dos tareas al mismo tiempo, bajo el mismo contexto y código.

Diseño del programa

A continuación, se presenta y especifica el diseño usado para el programa. Además, el manejo de las funciones implementadas en ella.

Decisión del diseño:

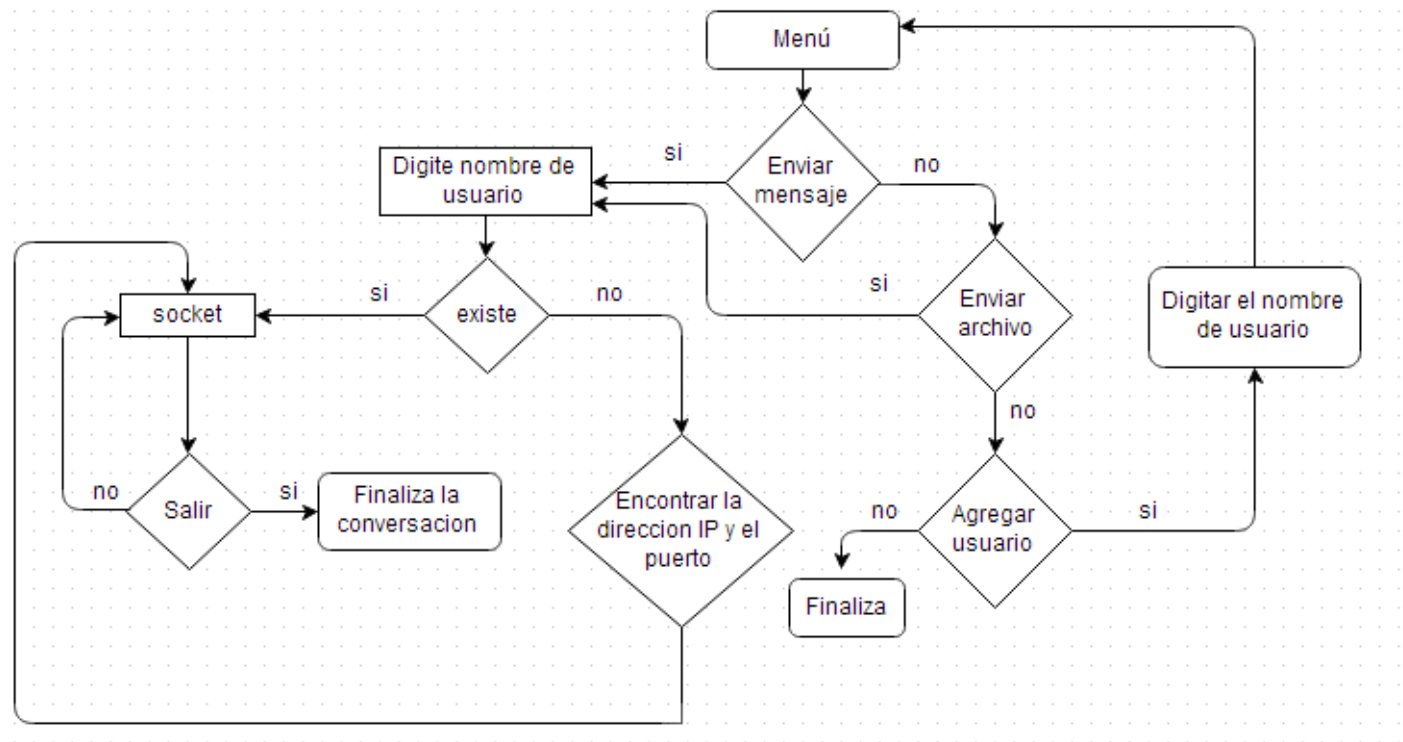
- El programa se desarrolló en el Lenguaje de Programación C, por ser requisito de la tarea.
- Se decidió utilizar el protocolo orientado a conexión Proceso de Control de Transmisión (TCP, por sus siglas en inglés) ya que esto representaba mayor seguridad y confiabilidad en la transmisión de datos.
- Se utiliza un menú, con esto se da mayor facilidad y agilidad ya que se le solicita al usuario insertar una selección mediante una serie de números para acceder a las diferentes funciones del programa.
- El diseño del programa está determinado para ser utilizado solamente por medio de la consola.

Algoritmos usados (Funciones principales del programa):

Nombre de la Función	Funcionamiento	Valor de retorno
Lee()	Consiste en recorrer el documento .txt donde se almacenaron los contactos junto con la información necesaria. Este se recorre mediante un puntero de tipo FILE, y junto con la implementación fgets(), se lee cada línea del documento y se almacena la información necesaria para enviar mensajes en arreglos estáticos.	No existe valor de retorno. El resultado son los datos recuperados del documento donde se encuentra la información de los contactos.
Escribir()	Es la encargada de agregar al documento los contactos que son guardados por el usuario. Mediante el fprintf(), se escribe en el archivo .txt especificando durante el escaneo de datos que se vaya almacenando en una línea por dato insertado.	No existe valor de retorno. Lo que concluye es el almacenamiento de los datos insertados por el usuario en el documento de texto.
Sockets()	Es el encargado de abrir la conexión entre los usuarios para que estos puedan enviar y recibir mensajes. En esta función se implementa el fork(). Cuando los usuarios se comunican ingresan a un ciclo infinito. La comunicación se cierra si un usuario escribe la palabra "salir".	No existe valor de retorno. La función comunica a los usuarios.
ValidaContactos()	Consiste en verificar si el usuario que se está intentando agregar ya existe en el archivo de contactos.	El retorno de esta función es 1, si el usuario no existe en el documento y 0 si el usuario si se encuentra en el documento.
Fork()	Hace referencia a la creación de una copia de sí mismo por parte de un programa, que entonces actúa como un "proceso	No existe valor de retorno.

	<p>hijo” del proceso originario, ahora llamado “padre”. Los procesos resultantes son idénticos, salvo que tienen distinto número de proceso (PID).</p> <p>Más generalmente, una bifurcación en un entorno multihilo significa que un hilo de ejecución se bifurca</p>	
Función de envió de archivos	Consiste en la función para el envió y recepción de archivos, estos pueden ser tipo .pdf, audios, videos, entre otros diferentes documentos.	No existe valor de retorno.
Encriptar	Esta función consiste en la encriptación de texto al ser enviado por el cliente, el cual escribe el texto original y este es encriptado al ser pasado al servidor, para seguridad del mismo. Y luego este es desencriptado para la recepción del otro usuario.	No existe valor de retorno.

Diagrama Lógico



Librerías de C utilizadas

Stdio.h: Esta es la biblioteca estándar del Lenguaje de programación C, el archivo de cabecera que contiene las definiciones de macros, constantes, declaraciones de funciones y definición de tipos usados por varias operaciones estándar de entrada y salida. La biblioteca estándar `stdio.h` nos permite manipular ficheros desde nuestros programas en C, para lo cual hemos de incluir la siguiente directiva al comienzo de nuestros ficheros de código: **#include <stdio.h>**

Stdlib.h: La biblioteca estándar `stdlib.h` es el archivo de cabecera de la biblioteca estándar de propósito general de los lenguajes de programación de C. Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras.

Las funciones de la biblioteca `stdlib.h` pueden ser agrupadas en las categorías de:

- Conversión de tipos.
- Generación de números pseudp-aleatorios.
- Gestión de memoria dinámica.
- Control de procesos.
- Ordenamiento y búsqueda.
- Aritmética.

Unistd.h: Librería para obtener acceso a variables.

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. Se utiliza para la creación de sockets y las correspondientes constantes.

Sys/types.h: Es una librería estándar del lenguajes de programación C. Esta librería define los tipos de datos de sistemas primitivos. Se define los tipos de datos utilizados en el código fuente del sistema. Debido a que algunos tipos de datos del sistema son accesibles a código de usuario, que pueden ser utilizados para mejorar la portabilidad a través de diferentes máquinas y sistemas operativos. También incluye las definiciones de tipos de enteros sin signo y direcciones.

Sys/socket.h: Esta estructura se utiliza para definir una dirección de conector que se utilizará en las funciones. La cabecera `<sys/socket.h>` definirá la estructura. Esta estructura será la siguiente:

- Lo suficientemente grande como para dar cabida a todas las estructuras de dirección de protocolo específico apoyados.
- Alineados en un límite adecuado para que los punteros a que pueden ser emitidos como punteros a estructuras de direcciones específicas de protocolo y se utilizan para acceder a los campos de las estructuras sin problemas de alineación.

netinet/in.h: La estructura funciona como una familia de direcciones de internet.

Netdb.h: La cabecera <netdb.h> puede definir el tipo **in_port_t** y **in_addr_t** tipo descrito en <netinet/in.h>. La cabecera <netdb.h> definirá la estructura **hostent** que incluye al menos los siguientes miembros:

- Nombre oficial del host.
- Tipo de dirección.
- La longitud de la dirección.
- Dirección estructurada de información.
- Direcciones errores de información.

Análisis de resultados

Objetivos alcanzados

- Se logró la conexión entre los usuarios por medio del protocolo TCP que se requería para el proyecto.
- Se alcanzó la correcta creación y enlace de los sockets.
- Se logró el almacenamiento de datos en los archivos de configuración.
- Se consiguió la lectura de datos almacenados para la implementación de los sockets.
- Se logró con éxito el envío y recepción de tanto mensajes como archivos de forma eficaz y mediante una red inalámbrica doméstica.
- La bifurcación, el proceso de recepción y envío simultáneamente, en otras palabras la implementación del Fork.
- Se logró eficazmente la encriptación en el envío de mensajes de texto.
- Se obtuvo con éxito el makefile.

Objetivos no alcanzados:

- Encriptación de archivos: no se logró el objetivo de encriptación por la falta de conocimiento y la poca información en línea para implementar el código dentro de la programación.

Manual de usuario

El debido archivo se encuentra adjunto con el nombre de Manual de Usuario.pdf.

Conclusión

La Tarea Programada 1, requirió de mucha investigación y análisis de comportamiento de gran cantidad de funciones implementadas. Además de paciencia, empeño, tiempo y dedicación para alcanzar con éxito los objetivos propuestos.

El uso de temas vistos en clase y lo investigado, aumentó en gran manera el conocimiento de cada uno de los integrantes del grupo de trabajo. Gracias a esto, se lograron encontrar temas más importantes y relevantes para el logro del proyecto, como lo fue el uso de sockets y la implementación del fork para lograr la comunicación efectiva entre los usuarios.