

Instituto Tecnológico de Costa Rica

Administración de Tecnologías de Información

Lenguajes de Programación

Profesor: Andréi Fuentes Leiva



TAREA PROGRAMADA 1

Documentación Técnica

Víctor Vargas Ramírez

José Pablo Matamoros Moya

TABLA DE CONTENIDOS

Descripción Corta del Programa.....	3
Métodos estudiados.....	3
Diseño del Programa	4
Métodos y funciones del Programa	4
Librerías externas utilizadas.....	4
Diseño gráfico del funcionamiento del programa.....	5
Método para enviar archivos.....	5
Método para recibir archivos.....	5
Acerca de los Sockets.....	6
Análisis de resultados.....	6
Objetivos Alcanzados.....	6
Objetivos no Alcanzados ***IMPORTANTE PARA EJECUTAR.....	6
Conclusiones.....	6

Descripción Corta del Programa

El programa consta de funciones variadas de transmisión de datos, el mismo se encarga de realizar envíos de archivos de un computador a otro mediante TCP/IP (Protocolo de transferencia de red), el mismo utiliza la dirección IP del servidor remoto y un puerto válido para enviar el archivo, este programa debe de ser capaz de enviar los archivos a la misma vez que puede recibirlos, esto se hará mediante el uso de la función **fork()**, la cual se encargará de copiar el mismo programa en dos distintos procesos, cada uno con un número de identificación diferente, lo cual nos permite crear procesos padre e hijos, los procesos son idénticos salvo por su número de identificación de proceso (PID), es una bifurcación en el hilo de ejecución en un entorno multihilo (*multithreading* en inglés), el cual en palabras resumidas es un entorno que permite ejecutar múltiples hilos de ejecución al mismo tiempo.

El programa se ejecutará y será capaz de ejecutar procesos diferentes los cuales se encarguen de emitir y recibir datos en la red local donde se encuentren, los archivos a enviar podrán estar bajo cualquier extensión y podrán contener cualquier tipo de dato que se desee transmitir.

Métodos estudiados

A continuación se hará referencia a algunos de los métodos que se estudiaron para la realización de una solución sólida para dicho problema, estos se explicarán de manera breve para así generar un conocimiento sobre las bases en las cuales está escrito el código.

1. **Método Fork:** Este método 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 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 encontrará activo hasta que reciba una señal de recepción de archivo, así el mismo programa podrá realizar las dos tareas al mismo tiempo bajo el mismo contexto y código.
2. **Protocolos de transmisión TCP/IP:** Es un modelo de protocolos de red para la transmisión de datos de extremo a extremo, que especifica como los datos deben de ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario. El protocolo cuenta con 4 capas de abstracción las cuales se definen en la RFC1122 (Request for Comments), estas normativas están reguladas por la Internet Engineering Task Force (IETF)(en español *Fuerza de Tareas de Ingeniería de Internet*), una organización que tiene como objetivo contribuir a la ingeniería de internet, es conocida por ser la entidad encargada de definir los estándares de internet, conocidos como RFC.
3. **Sockets de Internet:** Es un concepto para referirse a la comunicación de dos programas ubicados en diferentes computadoras y su intercambio de flujo de datos, estos constituyen un mecanismo de entrega de paquetes de datos provenientes de la tarjeta de red a los procesos apropiados. Deben cumplirse ciertos requisitos para que los programas puedan comunicarse, como por ejemplo: que uno de los programas sea capaz de localizar al otro, conocer las direcciones IP de ambos programas además del puerto (número de identificación de cada proceso en el dispositivo de red), entre otros.
4. **Protocolo FTP:** (File Transfer Protocol, en español *protocolo de transferencia de archivos*), es un protocolo para la transferencia de archivos entre sistemas conectados a una red TCP, basándose en la arquitectura cliente-servidor, con el cual desde un equipo cliente podemos conectarnos a un servidor para descargar o cargar archivos. Esto nos permite enviar archivos desde una computadora a otra sin importar el sistema operativo con el que se cuente en cada una de ellas.

Diseño del Programa

A continuación se especificará el diseño elegido para el programa, además se indicará la manera en que este realiza las diferentes funciones que se le han programado.

Métodos y funciones del Programa

El siguiente es un listado de los métodos que se usan en el programa así como los datos que trabaja, su funcionalidad y valor de retorno:

Nombre de archivo .c	Nombre del Método	Parámetros a recibir	Función del método	Valor de Retorno
client.c	error	puntero a una secuencia de caracteres.	Esta función consiste en enviar al estándar output un mensaje de error y salir del programa	Void (no retorno)
client.c	main	CantidadArgumentos .nombrePrograma direccionIP numeroPuerto	Función main principal	Valor numérico 0 (cero)
server.c	error	puntero a una secuencia de caracteres.	Esta función consiste en enviar al estándar output un mensaje de error y salir del programa	Void (no retorno)
server.c	main	Puerto para la recepción de archivos	Función main principal	Valor numérico 0 (cero)

Librerías externas utilizadas

Aquí se indican las diferentes librerías externas que debieron ser incluidas en nuestro programa para el correcto funcionamiento de la aplicación:

Nombre de las cabeceras	Funcionalidad
stdio.h	Contiene las definiciones de los macros, constantes, declaraciones de funciones y demás usados por varias operación de entrada y salida estándar.
stdlib.h	Librería estándar de C (es utilizada por ejemplo en la función: gethostbyname)
string.h	Contiene definiciones utilizadas para la manipulación de cadenas de caracteres
sys/socket.h	Contiene definiciones de estructuras que son utilizadas por la función socket
sys/types.h	Contiene definiciones de estructuras que son utilizadas por la función socket
netinet/in.h	Contiene definiciones de constantes y estructuras que son necesarias para el uso de direcciones de dominio en internet

Diseño gráfico del funcionamiento del programa

A continuación se detalla de forma gráfica cual es el funcionamiento del programa y cuales son los requerimientos que este solicita para el correcto funcionamiento.

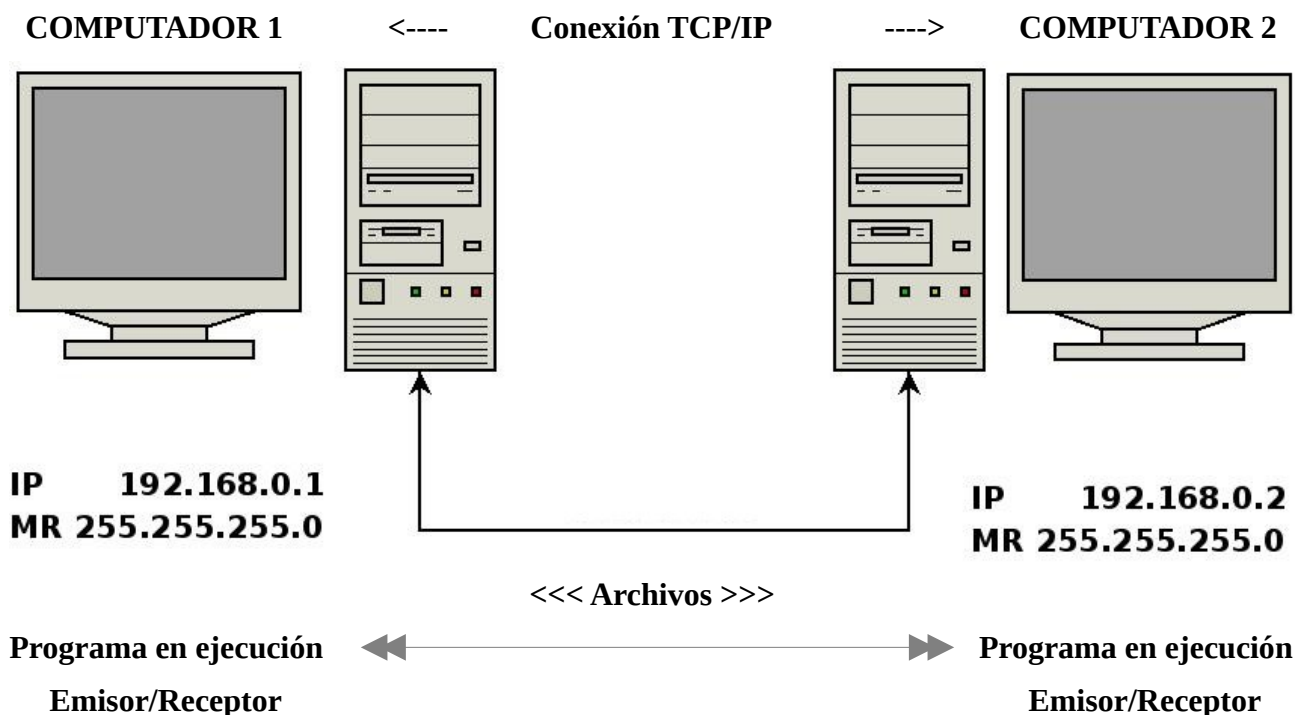


Gráfico 1

Como podemos ver en la gráfica los datos podrán ser enviados de un computador a otro mediante la dirección IP de las computadoras además de un puerto de comunicación para los programas, esto se puede realizar en una red interna conociendo de antemano los datos necesarios para enviar los archivos que se deseen.

***Nota: La MR que aparece en la figura no es utilizada en el programa, es la máscara de subred bajo la cual se encuentra la dirección IP asignada a cada computador. Además hay que tomar en cuenta que hay que manejar el puerto de comunicaciones por el cual se hará la transmisión en la red.

Método para enviar archivos

Se utiliza un espacio en la memoria al cual llamaremos buffer ya que es del mismo tipo, el cual nos proporciona un lugar para guardar las diferentes cadenas de caracteres del archivo a enviar, luego de almacenado se recorre por completo la cadena y se envía hasta que se encuentre el fin del archivo (EOF, End of File, en español *Final del Archivo*). De esto se encarga el programa cliente dentro de la aplicación.

Método para recibir archivos

El programa servidor dentro de la aplicación se encargará de mantenerse en lo que llamamos un periodo de escucha, esperando a que el socket asignado para la recepción de datos se active y de esta manera almacenar el archivo en un lugar específico utilizando un buffer igual que el creado para hacerlo, es de suma importancia que se designen bien las direcciones IP y puertos de la computadora ya que un mínimo error a la hora de ingresar los datos puede causar fallos significativos en el envío o recepción de archivos.

Acerca de los Sockets

Los sockets son una forma de hacer que dos programas se transmitan datos, basada en el protocolo TCP/IP, es un canal de comunicación entre dos programas que corren sobre ordenadores distintos o incluso en el mismo ordenador. Desde el punto de vista de la programación un **socket** es un fichero que se abre de manera especial, una vez abierto se pueden escribir y leer datos de él con las habituales funciones de **read()** y **write()** del lenguaje de programación C.

Análisis de resultados

A continuación se muestran los distintos resultados obtenidos en la creación del programa.

Objetivos Alcanzados

Se logra crear el programa cliente-servidor el cual permite hacer el envío de archivos de una computadora a otra indicando los datos necesarios para lograrlo, se logran hacer pruebas con resultados positivos enviando archivos en una red LAN privada, enviando archivos de texto de largo definido y logrando desplazarlo de un lugar a otro.

Objetivos no Alcanzados *****IMPORTANTE PARA EJECUTAR**

No se logra implementar la función de almacenado automático en una carpeta específica con el nombre original del archivo, las direcciones de los directorios donde será guardado el archivo recibido es uno establecido por defecto en el código fuente del programa. Por lo que cada vez que se desee implementar el programa en otra computadora se deberá modificar la dirección del directorio donde se desea almacenar el archivo.

Conclusiones

Se logra concluir mediante este trabajo que existen métodos funcionales que pueden programarse bajo ciertos paradigmas que incluyan envío y recepción de datos mediante comunicaciones de red de computadoras, logramos comprender de una manera amplia como se realiza la transmisión y empaquetamiento de datos en los protocolos estándares TCP/IP establecidos a nivel mundial para la comunicación de los datos.

Se logra comprender el funcionamiento del multihilo y su funcionalidad en las distintas aplicaciones que se deseen crear, así como sus beneficios a la hora de crear aplicaciones que logren satisfacer necesidades específicas que se busquen dentro del programa.

Se utilizaron las diferentes estructuras y código visto en clases para la realización de esta tarea por lo que se comprende que es una manera práctica de aplicar los conocimientos adquiridos durante el curso, y una forma de crear un ambiente de investigación y aplicación de conocimientos específicos sobre los temas aprendidos en las lecciones del curso de lenguajes de programación.