Jennifer Dos Reis 08-10323
Juliana Leon 08-10608

Proyecto 1 Laboratorio de Redes

Archivos:

- funciones.h: En este archivo se encuentra las firmas de las funciones implementadas en funciones.c además de los import a las librerías utilizadas que ofrece el lenguaje y los DEFINE de las constantes utilizadas

- funciones.c: En este archivo se implementan las funciones que se define en el archivo funciones.h

-makefile: Se genera la compilación de todos los archivos necesarios para la ejecución del proyecto , genera además los ejecutables de **bomba** y **centro**

-cliente.c : Implementa las funciones del cliente (bomba) según lo indicado en el enunciado del proyecto , la conexión con los servidores, las solicitudes de gasolinas , etc

-servidor.c: Implementa las funciones del servidor (centro) según el enunciado del proyecto, la espera de los clientes, y el suministro de gasolina de ser posible, etc.

Simulación:

-Realiza una simulación del abastecimientos de las bombas PDVSA, en donde los servidores son los centros de distribución y los clientes las bombas.

-Las bombas realizan la petición a los centros de suministro cuando así lo requieran, y si estos poseen el inventario necesario para enviar una gandola a la bomba que lo solicito , lo envian en el tiempo indicado por los parametros de entrada. Si el centro no puede realizar el en envío se le pide al siguiente en la lista de servidores ordenados por menor tiempo.

- El servidor espera la llegada de un cliente, y si su inventario es suficiente envía la gasolina

- Tanto el cliente como el servidor crean archivos de log, de la manera indicada en el enunciado, en la carpeta origen de la ejecución

Ejecución:

- En consola se ejecuta el makefile de la siguiente forma

$make

#luego se ejecutan los servidores de la siguiente forma, el orden de los argumentos es indiferente

$./centro -n Micentro -i InventacioInicial -s Suministro -t Tiempo -cp CapacidadMAx -p 10608(puerto o 10323)

Se ejecutan los clientes de la siguiente forma, el orden de los argumentos es indiferente

$./bomba -n Mibomba -cp CapacidadMax -i InventarioInicial -c Consumo -fc MiListaDNS

Consideraciones Especiales:

- Al inicio del cliente este se encarga de pedir los tiempos a todos los servidores de en la lista DNS dada, si alguno no logra responder, el cliente lo ignora luego en las peticiones de las gandolas

- El servidor atiende a una cantidad máxima de clientes al mismo tiempo (Especificado por la constante MAX_SERVERS)

- El máximo de caracteres de nombres se puede configurar con la constante MAX_LONG

- Para el tiempo tanto en el servidor como en el cliente se crea un hilo encargado de llevar el tiempo

- Los errores en general como los de conexión por ejemplo se reportan por la salida estándar. Si el cliente puede recuperarse del error continúa con la ejecución de lo contrario aborta.

Resumen Estructural:

Cliente:

- Validacion de parametros de entrada
- Lectura del archivo DNS
- Ordenamiento de los servidores por menor tiempo

Comienzo de la simulación

Evaluar la necesidad de gasolina
si la necesito pido al primer servidor en mi lista
si este no me la puede enviar la solicitud al siguiente
si no la necesito espero que si lo haga

Servidor:

- Validacion de parametros de entrada
- Configuración del socket del servidor (socket(), bind(), listen())
- Iniciar el contador de tiempo con un hilo hijo
- Mientras el tiempo sea menor a 480 minutos
    - Aceptar la petición del cliente
    - Verificar que el máximo de concurrencia no haya sido alcanzado
        - Si el máximo no ha sido alcanzado se crea un hijo para atender la petición
        - En caso contrario responder al cliente que no se le puede prestar servicio

* El hilo contador de tiempo se encarga también de actualizar el inventario cada segundo.
* El hilo que atiende las peticiones realiza las siguientes funciones:
- Recibe la solicitud
    - Si el cliente solicita información sobre el tiempo de entrega, se la envía
    - En caso contrario el cliente ha pedido gasolina entonces:
        - Si existe disponibilidad en el inventario, se envía la gasolina
        - En caso contrario se responde al cliente que no existe disponibilidad

Aplicación:

-Es una aplicación distribuida donde la operación no se realiza localmente en su totalidad por esto cae en la definición de distribuido aunque utiliza la arquitectura cliente-servidor que es lo más básico de sistemas distribuidos.

- Se utilizan Sockets para la comunicación por lo tanto nos encontramos en la familia de protocolos TCP/IP, es orientado a conexiones, se garantiza la transmisión de los datos y su orden.

- Un socket asocia una dirección y un puerto local, un protocolo de transporte , una dirección y un puerto remoto que permiten la transmisión de datos entre máquinas.