Francisco Blázquez Matías 70919093L

David Pulido Macías 70905670M

Redes de computadores I

El API de Sockets:

Implementación de la práctica:

La práctica implementa un protocolo HTTP/1.1 de conexión entre cliente y servidor utilizando dos protocolos de transporte: TCP y UDP.

Para ello hemos seguido el esquema implementado en el ejemplo proporcionado, incorporando al mismo las modificaciones necesarias para proporcionar la funcionalidad especificada en el enunciado de la práctica.

Hemos modificado aspectos relacionados con el manejo de ficheros, así como diversas comprobaciones necesarias para adaptar el funcionamiento del programa a las especificaciones pedidas, modificaciones que se han ido comentando en los propios ficheros .c. Donde queda más claro el objetivo de estas modificaciones.

Ambas funciones, una para TCP y una para UDP siguen esquemas parecidos, implementados cada uno con las funciones de envío y recepción de datos correspondientes (recv y send para TCP y recvfrom y sendto para UDP).

En servidor.c la función para TCP implementa un esquema en el que se recibe una petición del cliente, se trata dicha petición y se envía una respuesta al cliente.c
La función para UDP sigue un esquema parecido al de TCP, hay que destacar que la primera recepción no se hace dentro de la propia función por lo que dentro de la función y para seguir recibiendo las peticiones que sigue enviando el cliente se ha implementado un esquema en el que primero trata los datos (ya que siempre existirá una primera recepción fuera de la función), seguidamente enviará una respuesta al cliente y después de eso y controlado con un timeout (por si es el último mensaje enviado por el cliente) el servidor vuelve a recibir otra petición del cliente.

En cliente.c la función para TCP trata la entrada de datos introducida por el cliente y el fichero de ordenes proporcionado y lo envía siguiendo el protocolo HTTP/1.1 al servidor, después de eso recibe la respuesta del servidor y trata esa respuesta sacando a un fichero ".txt" el código html proporcionado en la respuesta.
La función para UDP del cliente.c sigue el mismo esquema que la de TCP exceptuando que al ser recvfrom una función bloqueante, se controla mediante un timeout y una serie de intentos la recepción de la respuesta del servidor.

También se ha implementado un fichero en el que se guarda un historial de las respuestas del servidor al cliente llamado peticiones.log

La salida de cada una de las peticiones del usuario va dirigida a un fichero con el nombre "puerto_efimero".txt

Protocolo de transporte:

Consideramos que el protocolo de transporte más adecuado para la implementación de esta práctica sería el TCP, que está orientado a conexión. Puesto que mediante el protocolo TCP se consigue mantener una conexión lógica entre los dos extremos, de manera que se puede controlar que todos los segmentos lleguen de forma adecuada, método necesario cuando existen transferencias para las que no pueden faltar bytes. En nuestro caso, se produce una conversación entre cliente y servidor de manera que ambos envían y reciben de una manera determinada la información, es decir, si en algún momento se perdiese algún mensaje durante toda la comunicación podrían producirse ciertos problemas, por tanto el protocolo TCP sería el más adecuado, al permitir una conversación estable.