SOCKETS

API

int socket (int domain, int type, int protocol)

Crea un endpoint para comunicación y regresa un descriptor de archivo que hace referencia al endpoint.

El argumento "domain" especifica un dominio de comunicación

int bind (int sockfd, const struct sockaddr * myaddr, int addrlen);

Asociar una dirección al socket.

myaddr: Asigna la dirección especificada en la estructura del tipo sockaddr

sockfd: Descriptor del socket involucrado

addrlen: tamaño de la estructura en *myaddr, es decir sizeof(myaddr)

La llamada al sistema bind( ) devuelve un 0, si se produjo un error devuelve -1.

int listen (int socketFd, int backlog)

La función listen( ) se invoca únicamente desde el servidor, y habilita al socket para poder recibir conexiones. Únicamente se aplica a sockets de tipos SOCK_STREAM.

sockfd: Identificador de socket obtenido en la función socket( ), que será utilizado para recibir conexiones.

backlog: Número máximo de conexiones en la cola de entrada de conexiones. Las conexiones activas quedan en estado de espera en esta cola hasta que sean aceptadas mediante la función accept( ).

int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);

La función accept() es utilizada en el servidor una vez que se ha invocado a la función listen(). Esta función espera hasta que algún cliente establezca una conexión con el servidor. Es una llamado bloqueante, esto es, la función no finalizará hasta que se haya producido una conexión o sea interrumpido por una señal.

sockfd: Identificador de socket habilitado para recibir conexiones.

addr: Puntero a una estructura sockaddr en donde se almacenará la información (dirección IP y número de puerto) del proceso que ha realizado la conexión.

addrlen: Debe contener un puntero a un valor entero que represente el tamaño de la estructura addr.

int connect(int sockfd, const struct sockaddr *serv_addr, socklen_t addrlen)

Esta función es invocada desde el cliente para solicitar el establecimiento de una conexión TCP.

La función connect() inicia la conexión con el servidor remoto, por parte del cliente.

sockfd: Identificador de socket devuelto por la función socket()

serv_addr: Estructura sockaddr que contiene la dirección IP y el número de puerto del destino.

addrlen: Debe ser inicializado al tamaño de la estructura serv_addr, pasada como parámetro.

Una vez que la conexión ha sido establecida, se inicia el intercambio de datos, utilizando para ello las funciones send() y recv().

ct send(int sockfd, const void *buf, size_t len, int flags)

sockfd: Identificador de socket para enviar datos.

buf: Puntero a los datos a ser enviados

len: Número de bytes a enviar

flags: Por defecto, 0.

La función send() devuelve el número de bytes enviados, que puede ser menor que la cantidad indicada en el parámetro len.

recv(int sockfd, void * buf, size_t len, int flags)

La función recv() se utiliza para recibir datos, y posee un formato similar a la función send()

sockfd: Identificador de socket para la recepción de los datos.

buf: Puntero a un buffer donde se almacenaran los datos recibidos.

len: Número máximo de byts a recibir.

flags: Modificar el comportamiento de la función recepción. Para un comportamiento por defecto, fijar el parametro a 0.

Servicios y la función de la capa de transporte.

* Rastreo de comunicación individual entre aplicaciones en los hosts de origen y destino.
* Segmentación de datos y manejo de cada parte.
* Reensamble de segmentos en streams de datos de aplicación
* Identificación de diferentes aplicaciones.

¿Qué es el modelo Cliente Servidor?

Es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Características que tienen los sockets TCP:

* Son orientados a la conexión.
* Se garantiza la transmisión de todos los octetos sin errores ni omisiones
* Se garantiza que todo octeto llegará a su destino en el mismo orden en el que se ha transmitido.