

1. WSASStartup:

WSASStartup is a function in the Windows Sockets (Winsock) API that is used to initialize the use of the Winsock DLL by a process. It takes as input a pointer to a WSADATA structure that receives details about the version of Winsock being used, and it returns an integer value that indicates whether the function was successful or not.

Parameters:

lpWSAData (output): A pointer to a WSADATA structure that receives information about the Winsock implementation.

wVersionRequest (input): The high-order byte will specify the minor version number and the low-order byte specifies the major version number.

Return Value:

int: If the function succeeds, it returns zero; otherwise, it returns an error code that indicates the reason for failure.

2. WSASocket:

WSASocket is a function in the Winsock API that creates a socket that can be used for communication over a network. It allows a process to specify the type of socket, the protocol, and other options.

Parameters:

af (input): The address family used by the socket (ex. AF_INET for IPv4).

type (input): The type of socket to be created (ex. SOCK_STREAM for TCP, SOCK_DGRAM for UDP).

protocol (input): The protocol to be used (ex. IPPROTO_TCP for TCP).

lpProtocolInfo (input): A pointer to a WSAPROTOCOL_INFO structure that specifies additional protocol information.

g (input): A group identifier for the socket, reserved for future use.

dwFlags (input): Flags that control the creation of the socket.

Return Value:

SOCKET: If the function succeeds, it returns a new socket descriptor that can be used for further communication; otherwise, it returns INVALID_SOCKET, which indicates an error.

3. WSAConnect:

WSAConnect is a function in the Winsock API that establishes a connection to a remote server from a client socket. It allows a process to specify the server's address, port number, and other options.

Parameters:

s (input): The socket descriptor identifying the client socket.

name (input): A pointer to a sockaddr structure that specifies the address of the remote server.

namelen (input): The size of sockaddr structure.

lpCallerData (input): A pointer to a WSABUF structure that contains any outgoing data to be sent during the connection establishment.

lpCalleeData (output): A pointer to a WSABUF structure that receives any incoming data received during the connection establishment.

lpSQOS (input): A pointer to a QOS structure that specifies the quality of service parameters for the connection.

lpGQOS (input): A pointer to a QOS structure that receives the quality of service parameters negotiated for the connection.

Return Value:

int: If the function succeeds, it returns zero; otherwise, it returns SOCKET_ERROR and a specific error code that indicates the reason for failure.

4. bind:

The function bind is in the Winsock API that associates a local address with a socket. It is typically used by a server to bind a socket to a specific IP address and port number, so that it can listen for incoming connections from clients.

Parameters:

s (input): The socket descriptor identifying the socket.

addr: A pointer to a sock_addr structure of the local address to assign to the bound socket.

namelen (input): The size of the sockaddr structure, bytes.

Return Value:

int: If the function succeeds, it returns zero; otherwise, it returns SOCKET_ERROR and a specific error code that indicates the reason for failure.

5. WSAAccept:

WSAAccept is a function in the Winsock API that accepts a new incoming connection on a listening socket. It is typically used by a server to accept incoming client connections after the bind and listen functions have been called.

Parameters:

s (input): The socket descriptor identifying the listening socket.

addr (output): A pointer to a sockaddr structure that receives the address of the connected client.

addrlen (input/output): A pointer to an integer that specifies the size of the sockaddr structure and receives the actual size of the address returned.

Module 5 Assignment

Gunnar Yonker

lpfnCondition (input): The address of an optional condition function that will make an accept or reject decision based on the caller information passed in as a parameter.

dwCallbackData (input): A user-defined value that is passed to the callback function.

Return Value:

SOCKET: If the function succeeds, it returns a new socket descriptor for the accepted connection; otherwise, it returns INVALID_SOCKET, which indicates an error.