

21 Lecture - CS506

Important Subjective

Certainly, here are 10 short subjective questions related to Socket Programming along with their answers:

****Question 1: What is Socket Programming?***

****Answer:**** Socket programming is a technique in computer networking that enables communication between processes running on different devices over a network using sockets, which are endpoints for sending and receiving data.

****Question 2: Explain the client-server model in Socket Programming.****

****Answer:**** The client-server model involves two types of processes: the client, which requests services, and the server, which provides services. In socket programming, the client establishes a connection to the server's socket, allowing them to exchange data.

****Question 3: What is a socket?***

****Answer:**** A socket is an endpoint for sending or receiving data across a computer network. It is identified by an IP address and a port number and is used to establish a connection between two computers.

****Question 4: What is the difference between TCP and UDP sockets?***

****Answer:**** TCP (Transmission Control Protocol) sockets provide reliable, connection-oriented communication with error checking and data sequencing. UDP (User Datagram Protocol) sockets provide connectionless, unreliable communication without error checking and sequencing.

****Question 5: How is a socket identified in a network connection?***

****Answer:**** A socket is identified by a combination of an IP address and a port number. The IP address locates the device, while the port number identifies a specific application or process on that device.

****Question 6: What is the purpose of the `bind()` function in socket programming?***

****Answer:**** The `bind()` function is used to associate a socket with a specific IP address and port number on the local machine. This allows the socket to listen for incoming connections or to send data from that address.

****Question 7:** Explain the steps involved in establishing a TCP connection using socket programming.

****Answer:**** The steps are:

1. Server creates a socket using `socket()` and binds it using `bind()`.
2. Server listens for incoming connections using `listen()`.
3. Client creates a socket using `socket()`.
4. Client initiates a connection to the server using `connect()`.
5. Server accepts the incoming connection using `accept()`.

****Question 8:** What is the role of the `accept()` function in socket programming?

****Answer:**** The `accept()` function is used by a server to accept an incoming connection request from a client. It creates a new socket for communication with that client and returns the socket's descriptor.

****Question 9:** How does error handling work in socket programming?

****Answer:**** Error handling involves checking return values of socket functions. Negative values indicate errors, and you can use functions like `perror()` to print error messages. Common errors include connection failures, timeouts, and invalid addresses.

****Question 10:** Can a single program act as both a client and a server using socket programming?

****Answer:**** Yes, a program can act as both a client and a server by implementing both client-side and server-side socket functionalities. This is often seen in peer-to-peer applications where devices communicate directly with each other.