Devanshu Surana
1032210755, PC-12

CN

Lab Assignment -8

Aim: To write a C program for wired network using TCP socket to demonstrate i) Chat application ii) Mathematical operation.

Objectives: 1 To understand concept of socket programming.

Theory

i) Client/server communication

⇒ Server is a process that waits passively for requests from clients, processes the work specified and returns the result to the client that originated the request

→ Client is a process that initiates a service request.

- TCP sockets are used for communication betn a server and a client process. The server's code runs first, which opens a port and istens for incoming connection requests from clients. Once a client connects to the same (server) port, the client of server may send a message.

ii) Introduction to TCP (Transmission Control Protocol)

⇒ TCP is a standard that defines how to establish and maintain a network ~~conservation~~ conversation through which application programs can exchange data. TCP works with the internet protocol (IP), which defines how computer send data to each other. Clubbed together, TCP and IP are the basic rules that define the internet.

iii) TCP segment Header:

Every TCP segment consists of a 20 byte fixed format header. Header options may follow the fixed header. With a header so, that it can tag upto 65535 data bytes.

Page No.
Date

iv) TCP Connection Establishment and release:

→ To establish a connection, the three way (or 3-step) handshake occurs.

1) SYN: The active open is performed by the client sending a SYN to the server. The client sets the segments sequence number to a random value A.

2) SYN-ACK: In response the server replies with a SYN-ACK, the acknowledgement number is set to one more than the recieved Sequence number that the server chooses for the packet is another random number, B.

3) ACK → Finally, the client sends an ACK back to the server. The sequence number is set to one more than the recieved sequence number ie B+1.

- The steps 1,2 establish the connection parameter for one direction and it is acknowledged

- The steps 2,3 establish the connection parameter for the other direction and it is acknowledged. With these a full-duplex communication is established.

- Releasing a TCP connection is symmetric. Either port can send a TCP segment with the FIN bit set, meaning it has no more information to send when the FIN is acknowledged that direction is shut down. Still, data can continue to flow continually in the other direction

v) Introduction to sockets:

→ A socket programming interface provides the routines required for interprocess communication between applications, either on the local system or spread in a distribute, TCP/IP based network environment. Once a peer-to-peer connection is established, a socket descriptor is used to uniquely identify the connection.

vi) TCP socket functions:-

- The TCP socket is able to listen on the TCP port for incoming connections. The TCP socket is able to initiate a connection to a remote server.
- It is able to listen on the TCP port for incoming connection and to initiate a connection to a remote server

vii) TCP socket flow Description on server and client.

→ The following shows the flow of a TCP connection.

i) The server creates the listener socket that is waiting for remote clients to connect.

ii) Client issues the connect () socket function to start the TCP handshake (SYN|ACK, SYN, ACK). The server issues the accept () socket function to accept the connection request.

iii) The client and server issue that read () and write () socket functions to exchange data over the socket.

iv) Either the server or client decides to close the socket This causes the TCP closure sequence (FIN's and ACK's) to occur.

v) The server either closes the listener Socket or repeats beginning with step 2 to accept another connection from a remote client.

FAQ's

Ans 1) The registered ports are in the range 1024-49151.
List atleast 5 port.

i) 110 - POP3 used by email clients to retrieve email from a server.

ii) 119 - NNTP

iii) 123 - NTP

iv) 143 - IMAP

v) 161 - SNMP

vi) 443 - HTTP

ans 2) If you are exiting, all unixes will close open file descriptors on exit. If you are not exiting, you can just close it with a regular close() call.

ans 3) TCP wraps each data packet with a header containing 10 mandatory fields totaling 20 bytes. Each header holds information about the connection and the current data being sent. The 10 TCP header fields are as follows: source port - The sending devices port.
Destination port - The recieving devices port,

← 32 bytes →

Source Port Number (16 bits)								Destination Port (16 bits)
32 bit Sequence number								
32 bit Acknowledgement number								
Header length (4bits)	Reserved (6 bits)	URG	ACK	PSH	RSH	SYN	FIN	Window size (16 bits)
TCP checksum (16 bits)								Urgent Pointer (16 bits)
options (optional)								
Data (optional)								

good work

A+

21/11/2022