Assignment 3
Socket Programming
[Max marks: 10]

Instructions:

- Termination condition for both the questions: Server will keep on accepting the request until client sends "Bye" message.
- You are free to use internet.
- Please do not copy from any source or other's assignment.

Task 1: Program to reverse the string.

Build connection-oriented (TCP) client server model. Client sends the string to the server and server reverses the string sent by the client and sends it back to the client. [5 marks]

Action at Server :

1. Create a socket using *socket( )* system call..
2. Bind server's address and port using *bind( )* system call.
3. Convert the socket into a listening socket using *listen( )* system call.
4. Wait for client connection to complete using *accept( )* system call.
5. Receive the Client request using *recv()* system call which consist of the name of the command that is to be executed along with data parameters (if any)
6. The command is interpreted and executed.
7. On successful execution the result is passed back to the client by the server

Actions at Client

1. Create a socket.
2. Fill in the internet socket address structure (with server information).
3. Connect to server using connect system call.
4. The client passes the command and data parameters (if any) to the server.
5. Read the result sent by the server, write it to standard output.
6. Close the socket connection.

Sample Input: The Client sends the string "CNLAB"

Sample Output The string will get back as reverse "BALNC"

Task 2 : Implementation of UDP Domain Name Server (DNS) Client/Server. [5 marks]

DNS identifies the unique name of the host with its IP address through server client communication.

Actions at Server:

1. Create an array of hosts and its ip address in another array i.e. the lookup table
2. Create a datagram socket and bind it to a port

3. Create a datagram packet to receive client request. Use recv() function to receive message from client
4. Read the domain name from client to be resolved
5. Lookup the host array for the domain name
6. If found then retrieve corresponding address
7. Create a datagram packet and send ip address to client
8. Repeat steps 3-7 to resolve further requests from clients
9. Close the server socket
10. Stop

Actions at Client side:

1. Create a datagram socket
2. Get domain name from user. Use send() function.
3. Create a datagram packet and send domain name to the server
4. Create a datagram packet to receive server message from step 7. Read server's response
5. If ip address then display it else display "Domain does not exist"
6. Close the client socket
7. Stop

It is not necessary to put correct IP addresses. You can assign dummy IP addresses in 4 octat format, i.e. w.x.y.z. For example 152.255.255.255. The size of the lookup table must contain atleast 5 domain names. **Display the lookup also in the output.**

Sample Input:

Client sends domain name example google.com

Sample output:

Server replies back with the IP address corresponding to the domain name in lookup table example. 152.255.255.255.