Indian Institute of Information Technology, Allahabad.

Course: Computer Networks (CNE532)

Batch: B.Tech (IT) Section B

Lab Assignment #2

Deadline: 29-04-2020

1. Write a client-server program that provides text and voice chat facility using datagram socket. Your application allows a user on one machine to talk to a user on another machine. Your application should provide non blocking chat facility to the users. This means, the user can send its message at any time without waiting for a reply from the other side. (Hint: Use select() system call).

2. Write a TCP client-server system to allow client programs to get the system date and time from the server. When a client connects to the server, the server gets the local time on the machine and sends it to the client. The client displays the date and time on the screen, and terminates. The server should be an iterative server.
   Submit two C files: time_server.c and time_client.c

3. Write a simple UDP iterative server and client to convert a given DNS name (for example, www.google.com) into its IP address(es). The client will read the DNS name as a string from the user and send it to the server. The server will convert it to one or more IP addresses and return it back to the client. The client will then print ALL the addresses returned, and exit. For basic UDP socket communication, see the sample program given. To get the IP address corresponding to a DNS name, use the function gethostbyname(). Look up the description of the function from the man page and the tutorial on the webpage.
   Submit two files: dns_server.c and dns_client.c

4. Now suppose that the same server will act both as the time server in Problem 1 and the DNS server in Problem 2. Thus, some clients will request over the UDP socket for name-to-IP conversion, and some will connect over a TCP socket for the time. Thus, the server now needs to open both a TCP socket and a UDP socket, and accept request from any one (using the accept() + read()/send() call for TCP, and recvfrom() call for UDP), whichever comes first. Use the select() call to make the server wait for any one of the two connections, and handle whichever comes first. All handlings are iterative.
   Submit one C file: combined_server.c