CS 39006: Assignment 2

Using TCP Sockets

Date: 17-Jan-2022

Deadline: 24-Jan-2022, 2 pm

Problem 2(a):

In this problem, you will write a client-server system using UDP sockets to convert a given DNS name (for ex., *www.google.com*) into its IP address(es). The client program will prompt the user for a DNS name, read it as a string and send it to the server using the UDP socket The server will convert the name to its equivalent IP addresses and returns them back to the client. If no IP address can be found, a special address (0.0.0.0) is returned. The client will then print **all** the addresses returned (or a suitable message if 0.0.0.0 is returned), and exit. You can assume the server will handle the different requests iteratively one after the other.

It is possible that the server may be busy, and takes too long to respond. If the client does not receive a response within 2 seconds, it closes the connection and exits with a suitable message.

The *gethostbyname( )* function takes as parameter a DNS name and returns IP address(es) corresponding to it in a special data structure. You can read the description of the function from the man page and the material given.

You have to submit two C programs – dnsserver.c (containing the server program) and dnsclient.c (containing the client program). Create a single compressed folder (zip or tar.gz) with the name <roll number>_Assgn2a.zip or <roll number>_Assgn2a.tar.gz, and put both your client and server programs in it. Submit this compressed folder in Moodle in the proper link by the deadline.

Problem 2(b)

Suppose that in the DNS system above, some clients will connect over TCP and some will connect over UDP sockets. Thus, the server will need to open both a TCP socket and a UDP socket, and accept DNS name resolution request from any one of them whenever they come. Use the *select*() call to modify your UDP server of Problem 2(a) above to do this. Also, in this case, the server is a concurrent server, so it will handle each request by spawning a new process using the fork() call.

Also write a TCP client whose behaviour is the same as the UDP client of Problem 1(a), just that it will use a TCP socket instead of a UDP socket.

You have to submit three C programs – new_dnsserver.c (containing the new server program), dnsclient.c (the same client program you wrote for Problem 2(a), submit again), and newdnsclient.c (containing the new DNS client using TCP sockets). Create a single compressed

folder (zip or tar.gz) with the name <roll number>_Assgn2a.zip or <roll number>_Assgn2a.tar.gz, and put both your client and server programs in it. Submit this compressed folder in Moodle in the proper link by the deadline.