Systems Software/Programming – Lab Manual

Lab 11 – Network Programming

Write C program for each of the problem below. After executing each of the C program, you will need to capture the output in text file format.
C program file will be named as StudentID_Lab11_x.c
Text file with output captured will be names as StudentID_Lab11_x.txt

This assignment will teach you how to create your own device driver for character device.

Problem1: Implementation of File Transfer from Server to Client using Sockets (Assume that at a time only 1 client request will be entertained by server)
Write StudentID_Lab11_1_ftp_server.c and StudentID_Lab11_1_ftp_client.c programs to send a file from server to client which is requested by client.

1. Server will be running on specified port (e.g. 15001). Using getaddrinfo(), socket(), bind() system calls, server will create file descriptor listenfd bound to server socket and then start listening on that socket using listen().
2. Client will connect to server using server's IP and port (e.g. 15001) using getaddrinfo() and connect() system calls with clientfd file descriptor.
3. Once server receives the connection request from client, connection is established by accepting connection at server end using accept() system call with returning of connfd. Now client and server are ready to communicate using clientfd on client side and connfd on server side.
4. Client asked user for a filename and will be send to the server then it will wait for the server's response. If file name entered by user is "quit" then client program closes the clientfd file descriptor and exits.
5. Once server receives the filename, it opens the file and sends the content of the file to client as it reads it. You can use a same size buffer on server and client side so that the same number of bytes are transferred in a single transmission.
6. When client starts receiving the file it will create a new file and write the content in that file. When server has finished sending the file, there should be special message from server to client indicating end of file.
7. Upon receiving end of file message client will close the file. Ensure that once file is sent file size and content of the file on server and client must be the same.

8. If server does not find a specified file an appropriate response must be sent to the client which is displayed to the user.
9. If client closes the clientfd, server will know that client side file descriptor is closed it should also close connfd and end that client thread.

Problem2: We enhance the problem 1 to allow multiple client requests to request for a file (please note different client can request for different files at the same time)

- In Problem 1 Step 2, when a client request is accepted by server using accept() system call, it will create a new POSIX thread with the function that will handle communication with client. You will need to pass connfd and client socket address to thread function using structure pointer.
- Hence, you will need to modify only server to create communication with different client in different threads upon client connection request. (Note: make a new copy of server program from StudentID_Lab11_1_ftp_server.c to StudentID_Lab11_2_ftp_server.c to make the necessary changes for this problem)
- Client program should work as is. (Note: continue to use the same client code from StudentID_Lab11_1_ftp_client.c)