CS69003: Networks Lab
Spring 2019
Grading Guidelines for Assignment 2

Marks Distributions: (Total: 25)

If they have used fopen/fscanf/fprintf anywhere in client/server instead of open/read/write as asked in the assignment, deduct 10% from whatever they get and add a comment.

TCP Server Program: 12

a) Define socket FD and call the socket function - 1
b) Create server address structure and initiate them - 1
c) Call the bind function – 1
d) Call the accept function with proper parameters - 1
e) Wait for the file name from the client (with recv call) – 2
    i) Give 0 if they assume that the entire file name can always be read with one recv() call only. To be correct, they should check if the last character received is '\0' or not to verify if the entire filename is received or not, and do recv() calls in a loop until that. So marks for this part will be 0 or 2.
f) Open the file on receiving the file name (if the file exists) - 1
g) Close the connection if the file is not there and come back to wait on accept() - 1
h) Read from the file in fixed size chunks - 1
    i) Can also read character by character and send; though inefficient, we have allowed it. But cannot read word by word with %s as then spaces will not be sent. Note that this text file is not of the same format as Assgn 1.
i) Send the chunks to the client - 1
j) Closing the connection after reading the entire file and come back to wait on accept() - 1
k) Close the file – 1

TCP Client Program: 13

a) Create the address structures and initiate them for connect – 1
b) Call the connect function - 1
c) Send filename to the server – 1
    i) Can send in one send() call. Ok if they use multiple though I don't see why anyone will.
d) Correct checking of return value of recv() on file not found and printing message – 1
    i) Return value will be 0 as server will close the socket and not send anything
    ii) No marks if they have used any other method (like send any specific message for not found etc.)

e) Reading the file contents through recv() calls – 7 (total, see split below. For parts iii and iv below, give 0 if they have counted words at the end after the entire file is written, they should do this at each step after each recv())
    i) Basic recv() call to read one chunk - 1
    ii) Writing the characters read to file - 1
    iii) Keeping running count of bytes read from return value of recv() - 1
    iv) Counting words in a single chunk correctly and keeping running count – 2
        1) Deduct 1 if they have not considered any number of occurrences of the delimiter characters
    v) Counting words spread across chunk boundaries (parts of word received in one recv() call and the next part in 2nd. Theoretically it can be split in more than 2 also depending on the chunk size, but ok if they have considered split across 2 only) – 2
        1) Give 1 marks if they have tried but not correct, 0 of they have not considered this at all
f) Correct checking of return value of recv() on end of file and close the file – 1
    i) Return value will be 0 as the server will close the socket afetr sending the file
    ii) No marks if they have used any other method (like sending size first, sending any special character/word) for detecting end
g) Close the socket and exit – 1