CS-356 Computer Networks Lab

Lab Session 5

Assignment

File Transfer using Stream Socket

Objective:

The objective of this assignment is to get familiar with stream sockets (also called TCP sockets) using POSIX C programming. A stream socket establishes a connection between the client and server, which remains there until one of them closes it (explicitly or implicitly on exit). The connection can be used to transfer an ordered sequence of bytes between two computers (processes) reliably.

Problem Statement:

Your task will be to write two programs - one program corresponding to the server process and another program corresponding to the client process. The client process requests for the content of a file (by providing the file name) and the server process sends the contents of that file to the client. The file is a text file of arbitrary size.

The transfer of the contents of the file works using a communication protocol as follows:

1. The client establishes a connection to the server using the connect() call.
2. The client reads the filename from the user (console input).
3. The client sends the file name to the server.
4. The server looks for the file in the local directory, if the file is not there it closes the connection without sending anything (the server does not send any error message to the client, it just closes the connection). The client needs to detect this and print an error message "ERR 01: File Not Found" and exit.
5. If the file is present, the server reads the contents of the file and sends it to the client. Since the file can be arbitrarily long, the server cannot send the entire file in a single send() call, and sends the file in small chunks using multiple send() calls until the entire file is transferred. The chunk size used by the server is not known to the client. The server closes the connection after sending the entire file.

6. The client reads the file contents sent by the server and copies them into a new file in the current directory. The client also maintains a running count of the number of bytes and the number of words in the file. For the purpose of this assignment, you can take ',' (comma), ';' (semicolon), ':' (colon), '.' (fullstop), or one or more whitespace (space/tab), or any combinations of them, as delimiters between two words. For example, "this is", "this is", "this,is", "this, is", "this, ; is" are all to be taken as 2 words).
7. After the entire file is transferred, the client prints a message "The file transfer is successful. Size of the file = X bytes, no. of words = Y", where X is the no. of bytes in the file and Y is the number of words in the file.
8. The server process keeps on waiting for the next client connection. The server process needs to be closed by sending an interrupt (using CTRL + C) from the console.

Requirements:

You need to ensure the following in your code:

1. No buffer of size > 100 can be used in either client or server.
2. The server cannot make more than one pass over the file, nor can it find the size of the file in any other way.
3. You cannot use fopen/fscanf/fprintf functions. You must use open/read/write functions to read/write from/to the file.

Submission Instruction:

You should write two C programs – `file_server.c` (contains the server program) and `file_client.c` (contains the client program).