READ ME

There are two separate make file for server and client.

**make -f make\_tcp\_preforked**

above command compiles server.

**make -f make\_tcp\_client\_file**

above command compiles client.

**make -f make\_tcp\_preforked clean**

above command removes all the server object files created.

**make -f make\_tcp\_client\_file clean**

above command removes all the client object files created.

**cd <directory>**

above command changes the directory where files are present.

This is a connection oriented(TCP) concurrent pre forked multiprocessing server where child processes are created well before the request comes from client. Child process calls accept() to receive new connection request. Once the request comes in from client it searches for particular file reads the data and send back the data to client. Client receives the data and stores it locally in the file.

Client Filename: **tcp\_client\_file.c**

Server Filename: **tcp\_preforked\_file\_server.c**

* Open two terminals and change the directory if applicable.
* Run the mentioned make files for server and client in their terminals respectively.
* In one terminal type the below mentioned command to start the server by giving the port number and number of child processes required as command line argument.

./ **tcp\_preforked\_file\_server 2 8003**

* After starting the server in one terminal run the client on other terminal. In order to do so use the below mentioned command along with port number and desired file name as command line arguments. Port number used here should be exactly same as used when invoking the server.

**./tcp\_client\_file 127.0.0.1 8003 students.txt**

* At last use the mentioned make clean to remove all object files created.

