**README**

**1/ To compile the code:**

cc –o client1 client1.c

cc –o client2 client2.c

cc –o client3 client3.c

cc –o server server.c

**2/ To run the code:**

For server: ./server

For one of clients:

Client 1: ./client1 IPaddress

Client 2: ./client2 IPaddress

Client3: ./client 3 IPaddress

*Note: IPaddress is the address of the server which you want to connect*

**3/ Explain the code:**

|  |  |
| --- | --- |
| CLIENT | |
| Line 1 to line 9 | Include files |
| Line 11 to line 49 | Create socket and set up TCP connection to server |
| Line 52 to line 53 | Read data from input file |
| Line 55 | Read line by line, each while loop will read one line at a time |
| Line 62 | Send the whole line read from input file to server |
| Line 63 to line 73 | Stroke the line read from input file into tokens, then put tokens into an array, and then print that array on client terminal as “unsorted array” |
| Line 76 to line 79 | Receive data from server. While loop will check 3 conditions, if one of them is false, the while loop will stop:   * strchr(reply, “\n”) == NULL: \n is sent by server with each transmission. We check for the \n and continue reading data each time we get back data from server. After server sent many pieces of reply data, it sent a single \n character so that the client would know the full line had been received. * ret > 0: make sure the receive is successful * count < sizeof(reply): make sure all bytes is receive successfully |
| Line 84 to line 92 | Eliminate all un-necessary \n \r, then put strings into a 2d array and then print that 2d array on client terminal as “sorted array” |
| Line 97 | Close the file after finishing |
| Line 100 to 102 | Handle the case when the file cannot be opened |
| Line 105 to 107 | Asking whether user want to exit the client or want to sort array again. The purpose of these line is to prevent the client exit immediately after finishing its job so that we can run clients concurrently |
| Line 108 to line 110 | Close and exit client |

|  |  |
| --- | --- |
| SERVER | |
| Line 1 to line 10 | Include files |
| Line 12 to line 76 | Set up TCP connection between server clients |
| Line 79 to line 82 | Using fork() to create child processes so that clients can run concurrently |
| Line 82 | Close parent socket |
| Line 87 to line 96 | Receive data from client. While loop will check 3 conditions, if one of them is false, the while loop will stop:   * strchr(reply, “\n”) == NULL: \n is sent by server with each transmission. After server sent many pieces of reply data, it sent a single \n character so that the client would know the full line had been received. * ret > 0: make sure the receive is successful * count < sizeof(reply): make sure all bytes is receive successfully |
| Line 100 to line 108 | Stroke whole line sent from client into tokens, then put these tokens into a 2d array |
| Line 110 | Merge sort the 2d array |
| Line 111 to 114 | Send one element of 2d array at a time because it is not allowed to send the whole 2d array to client. |
| Line 115 | After sending all array, server sent a single \n character as a signal so that client knew the whole array had been sent |
| Line 119 | Shutdown child process |
| Line 121 to 132 | Use waitpid() to handle zombies |
| Line 137 to 193 | Details of merge sort algorithm |

**4/ Example to run the program**

