ASSIGNMENT 1 – DESIGN MODIFICATION

- Developing a server program that accepts an array of integers from a client and returns the sorted array to the client based on TCP/IP client-server fashion.

MD. AZHARULLAH SHARIFF
B130727CS

PROGRAMS AND STEPS :

There are two C program
1) 1_B130727CS#client.c - The client program
2) 1_B130727CS#server.c - The server program

The client and server, both create a socket of kind IPv4, streamed and TCP. Then, a struct (of sockaddr_in) is created and is assigned all the required values (sin_family, sin_port, sin_addr) in both the programs. Then, in the server, we would bind the socket with a port on the local machine and keep listening for incoming connections from the client. Then we would connect the client to the remote host. The server would then accept the incoming connection and now a connection would have been established between the client and server. The client and server could communicate by passing data back and forth on the streamed sockets. For the required program, the numbers are taken as input in the client program and the numbers are sent one by one to the server. The choice number is also taken as input and sent to the server. The server receives the numbers and stores them in an array. The choice number is also received. Then, a sorting algorithm (Bubble sort) is run and the array is sorted. Then, the numbers are sent back, one-by-one, in the sorted order to the client and the client stores the numbers in the sorted order in an array based on the choice number. If choice is 1, array is sent in the same order. If choice is 2, array is sent in the reversed order. The sorted array is then displayed from the client onto the console. After the communication is over, the connection on the socket descriptor is closed.In the server and client codes, appropriate messages are displayed indicating each step.

CLIENT PROGRAM :
1)
sockfd = socket(PF_INET, SOCK_STREAM, 0);
This creates a socket and returns a socket descriptor.

2) myaddress.sin_family = AF_INET;
myaddress.sin_port = htons(3501);
myaddress.sin_addr.s_addr = INADDR_ANY;
memset(&(myaddress.sin_zero),'\0',8);
This assigns the appropriate values to the myaddress struct.

3) connectfd = connect(sockfd, (struct sockaddr *)&myaddress, sizeof(struct sockaddr));
This sends a connection request to the server with the created socket's descriptor

4) Then 'n', the number of integers is scanned as an input.

5) send(sockfd, &z, 4, 0);The n numbers are accepted one-by-one and

immediately sent to the server as soon as each number is accepted. Then, the choice number is also sent

6) recv(sockfd, &z, 4, 0);
Each number, received from the server, after sorting is accepted and stored in an array.

7) close(sockfd);
Finally, the array of integers are displayed and then the connection on the socket descriptor is closed.

SERVER PROGRAM :

1)
sockfd = socket(PF_INET, SOCK_STREAM, 0);
This creates a socket and returns a socket descriptor.

2) myaddress.sin_family = AF_INET;
myaddress.sin_port = htons(3501);
myaddress.sin_addr.s_addr = INADDR_ANY;
memset(&(myaddress.sin_zero),'\0',8);
This assigns the appropriate values to the myaddress struct.

3) bindfd = bind(sockfd, (struct sockaddr *)&myaddress, sizeof(myaddress));This is to bind the socket with a port on the local machine . keep listening for incoming connections from the client.

4) listenfd = listen(sockfd, backlog);
This is to keep listening for incoming connections from the client.

5) acceptfd = accept(sockfd, (struct sockaddr *)&clientaddress, &clientaddresssize);
The server accepts the incoming connection from the client and now a connection would have been established between the client and server.

6) recv(sockfd, &z, 4, 0);
Each number, received from the client, after input is accepted and stored in an array. The choice number is also received.

7) send(sockfd, &z, 4, 0);
The sorted numbers are sent one-by-one to the client based on the choice.