**Server**

|  |
| --- |
| //Include standard library |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  |  |
|  | //Include the socket and types for networking |
|  | #include <sys/types.h> |
|  | #include <sys/socket.h> |
|  |  |
|  | //For storing the socket address include the structure |
|  | #include <netinet/in.h> |
|  |  |
|  | int main(){ |
|  | //Message to send |
|  | char server\_msg[256]="Hello! The Socket is created successfully and connection is established\n"; |
|  |  |
|  |  |
|  |  |
|  |  |
|  | //Create socket |
|  | int server\_socket; |
|  | server\_socket=socket(AF\_INET,SOCK\_STREAM,0); |
|  |  |
|  | //Address Structure |
|  | struct sockaddr\_in server\_addr; |
|  | server\_addr.sin\_family=AF\_INET; |
|  | server\_addr.sin\_port=htons(9000); |
|  | server\_addr.sin\_addr.s\_addr=INADDR\_ANY; |
|  |  |
|  |  |
|  | //Binding the socket to specified IP and port- syntax same as connect function |
|  | bind(server\_socket,(struct sockaddr\*)&server\_addr,sizeof(server\_addr)); |
|  |  |
|  | //listen for connections |
|  | listen(server\_socket, 5); |
|  | /\*syntax |
|  | listen(current socket ,no of connections to listen) |
|  |  |
|  | \*/ |
|  |  |
|  |  |
|  | //integer to hold the client socket |
|  |  |
|  | int client\_socket; |
|  | client\_socket=accept(server\_socket,NULL,NULL); |
|  | //accept the connection from the client |
|  | /\* |
|  | syntax- |
|  | accept(current socket,address of client connection) |
|  | parameter-structures to have the address of the client |
|  | if we need the infomration of the client we can send the structure so that it will fill the data of the client |
|  | \*/ |
|  |  |
|  | //Send the message to the client socket |
|  |  |
|  | send(client\_socket,server\_msg,sizeof(server\_msg),0); |
|  | //Synatax same as recv |
|  |  |
|  |  |
|  | //close the socket |
|  | close(server\_socket); |
|  |  |
|  | return 0; |
|  | } |

**Client**

|  |
| --- |
|  |
| //We need include from standard library |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  |  |
|  | //We need types and socket api which contains most important functionality |
|  | #include <sys/types.h> |
|  | #include <sys/socket.h> |
|  |  |
|  | //we need structures to store the address information |
|  | #include <netinet/in.h> |
|  |  |
|  | int main(){ |
|  |  |
|  | //create a socket declare an interger for the address |
|  | int network\_socket; |
|  | network\_socket=socket(AF\_INET,SOCK\_STREAM,0); |
|  | /\*Parameter 1=domain name(internet socket we will pass), |
|  | Parameter 2 -to decide the layer 4 protocol udp or tcp |
|  | for tcp-SOCK\_STREAM |
|  | parameter-just specifies the protocol here 0 as we use tcp -default protocol |
|  | There are times where we have to specify the protocol explicitly |
|  | \*/ |
|  |  |
|  | //To connect to other socket |
|  | //we need to call the connect function in the socket |
|  | //But before we call the connect we should be able to specify to the address we have to connect |
|  | //netinet contains a structure that defines feilds of port number and ip address |
|  | struct sockaddr\_in server\_address; |
|  | //specify the address family same as family of socket i.e AFINET |
|  | server\_address.sin\_family=AF\_INET; |
|  | //specify the port |
|  | server\_address.sin\_port=htons(9000); |
|  | /\*data format is slightly different so we need to convert integer to right network byte order |
|  | so we use htons(int port) |
|  | \*/ |
|  | //finally specify the ip address |
|  | //sin\_addr is struct |
|  | server\_address.sin\_addr.s\_addr=INADDR\_ANY;//=0000 |
|  |  |
|  |  |
|  | //Now we can connect |
|  | int conn=connect(network\_socket,(struct sockaddr \*)&server\_address,sizeof(server\_address)); |
|  | /\*Syntax- |
|  | connect(current socket to connect ,address(cast the server address structure to sligtly different structure(struct sockaddr\*)),size of the server |
|  | address ) |
|  | this fucntion returns an integer so that we can do error handling |
|  | 0-everything ok |
|  | -1-not a good connection |
|  | \*/ |
|  |  |
|  | if(conn==-1){printf("There was an error while connection");} |
|  | //Now send or recieve data |
|  | //to fetch data declare string |
|  | char server\_response[256]; |
|  | recv(network\_socket,&server\_response,sizeof(server\_response),0); |
|  | /\*syntax |
|  | recv(current socket,address of the reciver,size of the reciver,optional flags parameter) |
|  | \*/ |
|  |  |
|  | //print out the data we get back |
|  |  |
|  | printf("The server sent the data:\n %s",server\_response); |
|  |  |
|  |  |
|  | //Close the socket once exchange is done |
|  | close(network\_socket); |
|  |  |
|  |  |
|  | return 0; |
|  |  |
|  | } |
|  |  |