Eco client server-client

**Client**

#include<stdio.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<string.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

int main()

{

int sock, bytes\_recieved,port;

printf("Enter port :");

scanf("%d",&port);

char send\_data[1024],recv\_data[1024];

struct sockaddr\_in server\_addr;

sock = socket(AF\_INET, SOCK\_STREAM, 0);

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(port);

server\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

bzero(&(server\_addr.sin\_zero),8);

connect(sock, (struct sockaddr \*)&server\_addr,

sizeof(struct sockaddr));

while(1)

{

printf("\nSEND (q or Q to quit) : ");

scanf("%s",send\_data);

if (strcmp(send\_data , "q") != 0 && strcmp(send\_data , "Q") != 0)

send(sock,send\_data,strlen(send\_data), 0);

else

{

send(sock,send\_data,strlen(send\_data), 0);

close(sock);

break;

}

bytes\_recieved=recv(sock,recv\_data,1024,0);

recv\_data[bytes\_recieved] = '\0';

if (strcmp(recv\_data , "q") == 0 || strcmp(recv\_data , "Q") == 0)

{

close(sock);

exit;

}

else

printf("\nRecieved data = %s " , recv\_data);

}

return 0;

}

**Server**

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

int main()

{

int sock, connected, bytes\_recieved , true = 1;

char send\_data [1024] , recv\_data[1024];

struct sockaddr\_in server\_addr,client\_addr;

int sin\_size;

sock = socket(AF\_INET, SOCK\_STREAM, 0);

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(8080);

server\_addr.sin\_addr.s\_addr = INADDR\_ANY;

bind(sock, (struct sockaddr \*)&server\_addr, sizeof(struct sockaddr));

listen(sock, 5);

printf("\nTCPServer Waiting for client ");

fflush(stdout);

while(1)

{

sin\_size = sizeof(struct sockaddr\_in);

connected = accept(sock, (struct sockaddr \*)&client\_addr,&sin\_size);

printf("\n I got a connection from (%s , %d)", inet\_ntoa(client\_addr.sin\_addr),ntohs(client\_addr.sin\_port));

while (1)

{

bytes\_recieved = recv(connected,recv\_data,1024,0);

recv\_data[bytes\_recieved] = '\0';

if (strcmp(recv\_data , "q") == 0 || strcmp(recv\_data , "Q") == 0)

{

close(connected);

break;

}

else

printf("\n RECIEVED DATA = %s " , recv\_data);

if (strcmp(recv\_data , "q") == 0 || strcmp(recv\_data , "Q") == 0)

{

send(connected, recv\_data,strlen(recv\_data), 0);

close(connected);

break;

}

else

send(connected, recv\_data,strlen(recv\_data), 0);

fflush(stdout);

}

}

close(sock);

return 0;

}

Output: