Distributed systems
Assignment 3

NAME: SOURABH PATEL

ADMISSION NO: U19CS082

Implement echo client-server message passing application. Message sent from client should be displayed on server and then program should terminate.

1. Write a server (TCP) C Program that opens a listening socket and waits to serve client.

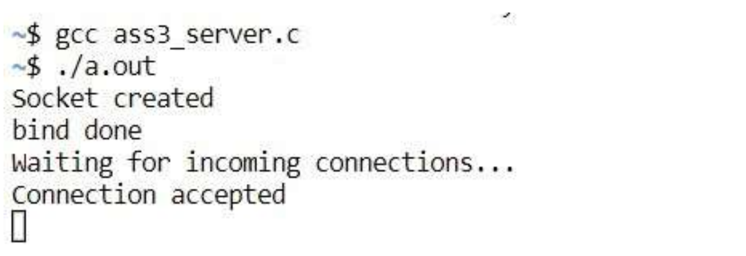
2. Write a client (TCP) C Program that connects with the server program knowing IP address and port number.

3. Get the input string from console on client and send it to server, server displays the same string.

SERVER:

```c
/*
    C socket server example
*/

#include<stdio.h>
#include<string.h>  //strlen
#include<sys/socket.h>
#include<arpa/inet.h>   //inet_addr
#include<unistd.h>  //write

int main(int argc , char *argv[])
{
    int socket_desc , client_sock , c , read_size;
    struct sockaddr_in server , client;
    char client_message[2000];

    //Create socket
    socket_desc = socket(AF_INET , SOCK_STREAM , 0);
    if (socket_desc == -1)
    {
        printf("Could not create socket");
    }
    puts("Socket created");

    //Prepare the sockaddr_in structure
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = htons( 8888 );

    //Bind
    if( bind(socket_desc,(struct sockaddr *)&server , sizeof(server)) < 0)
    {
        //print the error message
        perror("bind failed. Error");
        return 1;
    }
    puts("bind done");

    //Listen
    listen(socket_desc , 3);

    //Accept and incoming connection
    puts("Waiting for incoming connections...");
    c = sizeof(struct sockaddr_in);

    //accept connection from an incoming client
    client_sock = accept(socket_desc, (struct sockaddr *)&client,
(socklen_t*)&c);
    if (client_sock < 0)
    {
        perror("accept failed");
        return 1;
    }
    puts("Connection accepted");

    //Receive a message from client
    while( (read_size = recv(client_sock , client_message , 2000 , 0)) > 0
)
    {
        //Send the message back to client
        write(client_sock , client_message , strlen(client_message));
    }

    if(read_size == 0)
    {
        puts("Client disconnected");
        fflush(stdout);
    }
    else if(read_size == -1)
    {
        perror("recv failed");
    }

    return 0;
}
```

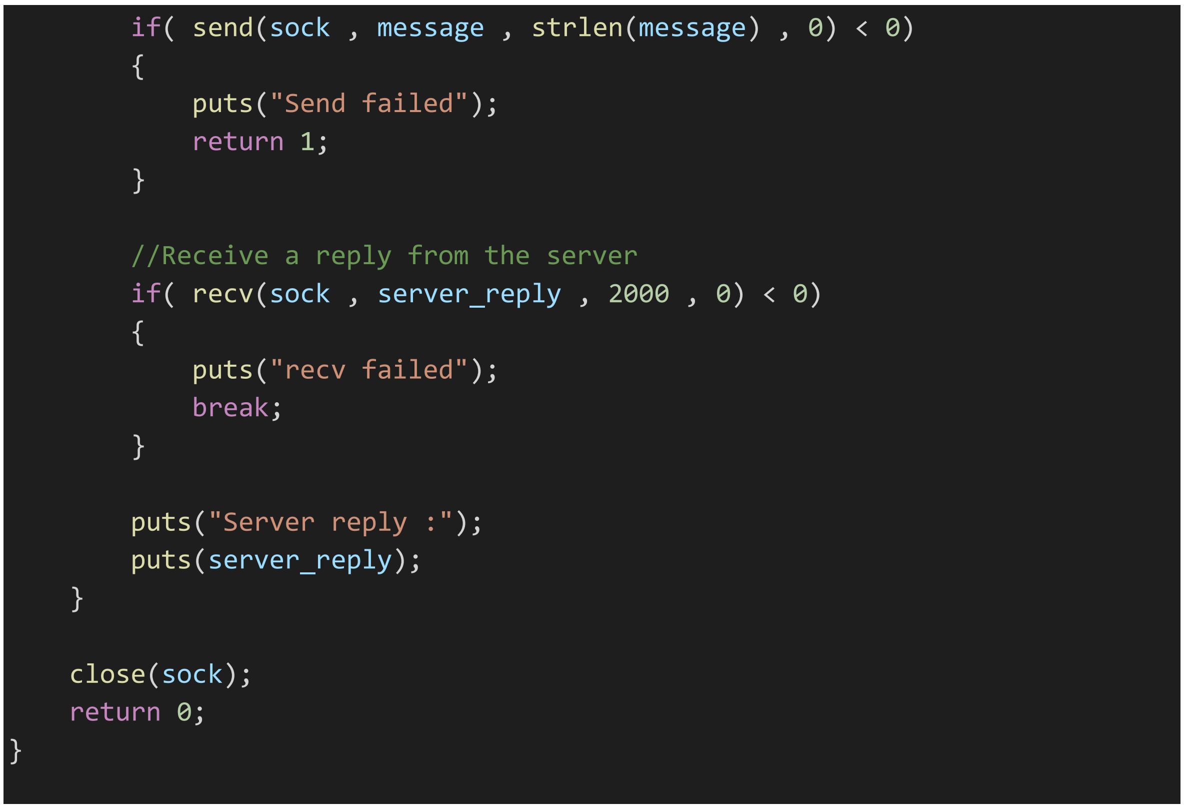
OUTPUT:



```
~$ gcc ass3_server.c
~$ ./a.out
Socket created
bind done
Waiting for incoming connections...
Connection accepted
▯
```

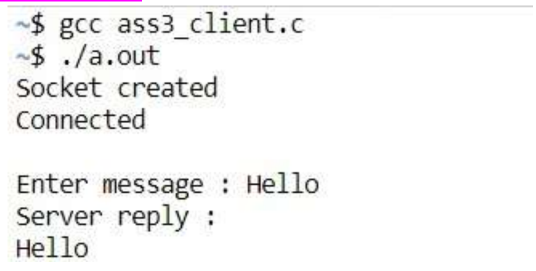
CLIENT:

```c
#include <stdio.h>  //printf
#include <string.h> //strlen
#include <sys/socket.h> //socket
#include <arpa/inet.h>  //inet_addr
#include <unistd.h>

int main(int argc , char *argv[])
{
    int sock;
    struct sockaddr_in server;
    char message[1000] , server_reply[2000];

    //Create socket
    sock = socket(AF_INET , SOCK_STREAM , 0);
    if (sock == -1)
    {
        printf("Could not create socket");
    }
    puts("Socket created");

    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    server.sin_family = AF_INET;
    server.sin_port = htons( 8888 );

    //Connect to remote server
    if (connect(sock , (struct sockaddr *)&server , sizeof(server)) < 0)
    {
        perror("connect failed. Error");
        return 1;
    }

    puts("Connected\n");

    //keep communicating with server
    while(1)
    {
        printf("Enter message : ");
        scanf("%s" , message);

        //Send some data
```



```c
        if( send(sock , message , strlen(message) , 0) < 0)
        {
            puts("Send failed");
            return 1;
        }

        //Receive a reply from the server
        if( recv(sock , server_reply , 2000 , 0) < 0)
        {
            puts("recv failed");
            break;
        }

        puts("Server reply :");
        puts(server_reply);
    }

    close(sock);
    return 0;
}
```

OUTPUT:



```
~$ gcc ass3_client.c
~$ ./a.out
Socket created
Connected

Enter message : Hello
Server reply :
Hello
```