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

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define MAX 80
#define PORT 8080
#define SA struct sockaddr

void func(int connfd)
{
char buff[MAX];
int n;
bzero(buff, MAX);
read(connfd, buff, sizeof(buff));
printf("\nFrom client: %s", buff);
}
int main()
{
int sockfd, connfd, len;
struct sockaddr_in servaddr, cli;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd == -1)
{
printf("\nSocket creation failed...\n");
exit(0);
}
else
printf("\nSocket successfully created..\n");
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
servaddr.sin_port = htons(PORT);
if ((bind(sockfd, (SA *)&servaddr, sizeof(servaddr))) != 0)
{
printf("\nSocket bind failed...\n");
exit(0);
}
else
printf("\nSocket successfully binded..\n");
if ((listen(sockfd, 5)) != 0)
{
printf("\nListen failed...\n");
exit(0);
}
else
printf("\nServer listening..\n");
len = sizeof(cli);
connfd = accept(sockfd, (SA *)&cli, (socklen_t *)&len);
if (connfd < 0)
{
printf("\nServer didn't accept the client...\n");
exit(0);
}
else
printf("\nServer accepted the client...\n");
func(connfd);
close(sockfd);
}
```

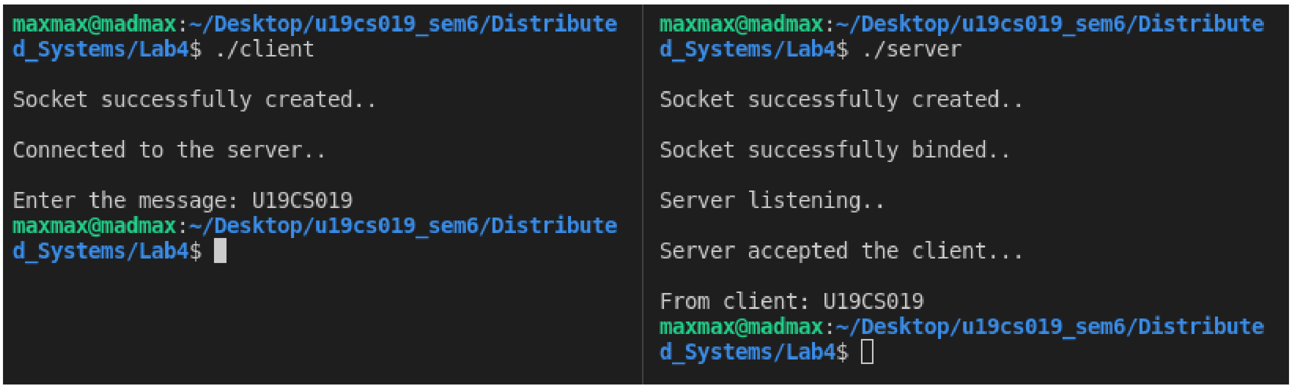
CLIENT.C

```
#include <stdio.h>
#include <netdb.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#include <arpa/inet.h>

#define MAX 80
#define PORT 8080
#define SA struct sockaddr

void func(int sockfd)
{
char buff[MAX];
int n;
bzero(buff, sizeof(buff));
printf("\nEnter the message: ");
n = 0;
while ((buff[n++] = getchar()) != '\n')
;
write(sockfd, buff, sizeof(buff));
}
int main()
{
int sockfd, connfd;
struct sockaddr_in servaddr, cli;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd == -1)
{
printf("\nSocket creation failed...\n");
exit(0);
}
else
printf("\nSocket successfully created..\n");
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
servaddr.sin_port = htons(PORT);
if (connect(sockfd, (SA *)&servaddr, sizeof(servaddr)) != 0)
{
printf("\nServer connection failed...\n");
exit(0);
}
else
printf("\nConnected to the server..\n");
func(sockfd);
close(sockfd);
}
```



```
maxmax@madmax:~/Desktop/u19cs019_sem6/Distribute
d_Systems/Lab4$ ./client

Socket successfully created..

Connected to the server..

Enter the message: U19CS019
maxmax@madmax:~/Desktop/u19cs019_sem6/Distribute
d_Systems/Lab4$ 
```

```
maxmax@madmax:~/Desktop/u19cs019_sem6/Distribute
d_Systems/Lab4$ ./server

Socket successfully created..

Socket successfully binded..

Server listening..

Server accepted the client...

From client: U19CS019
maxmax@madmax:~/Desktop/u19cs019_sem6/Distribute
d_Systems/Lab4$ 
```