LAB ASSIGNMENT-3

(28.07.2021)

NAME: RUDRASISH MISHRA
ROLL NO: 1906649
SECTION: IT-8
DATE: 28-07-21

QUESTION> Write a sender and receiver program in C using connection less socket. Both of them will exchange message with each other contineously. If any one receives the "exit" message from the other end then both of them will close the connection (Assume both client and server are running within the same host).

SOLUTION:

SERVER CODE:

```
        #include <stdio.h>
        # include <netinet/in.h>
        # include <netdb.h>
        # include <stdlib.h>
       # include <string.h>
       # include <sys/socket.h>
       # include <sys/types.h>
       # define MAX 80
       # define PORT 8080
       # define SA struct sock addr.
void func (int sockfd)
  { char buff [MAX];
    int n;
    for(;;)
  {   bzero (buff, MAX);
      read (sockfd, buff, sizeof (buff));
      Printf ("From Client : %s\t To Client : ", buff);
      b zero (buff, MAX);
      n= 0;
      while ((buff [n++] = getchar ()) != '\n');
      write (sock fd, buff, size of (buff));
   if (strcmp ("exit", buff, 4) == 0) {
  }}      Paint f ("Server Exit ...\n");
 }       break;
```

NAME: RUDRASISH MISHRA
ROLL NO: 1906649
SECTION: IT-8

```
int main ()
{
    int sockfd, connfd, len;
    struct sockaddr_in servaddr, cli;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd == -1)
     { printf ("socket creation failed ...\n");
       exit (0);
     }
else
  printf ("socket successfully created ..\n");
  bzero (& servaddr, sizeof (servaddr));
  servaddr.sin_family = AF_INET;
  servadd.sin_addr.s_addr = htonl(INADDR_ANY);
  servaddr.sin_port = htons(PORT);
 if ((bind (sockfd, (SA*) &servaddr, sizeof (servaddr))) !=0) {
     exit(0);
  }
   else
     printf ("socket successfully binded ...\n");
  if ((listen (sockfd, 5)) != 0) {
     printf ("listen failed ...\n");
     exit (0);
  }
  else
     printf ("server listening ...\n");
  len = sizeof(cli);
  connfd = accept (sockfd, (SA*)&cli, &len);
 if (connfd < 0)
 { printf ("server accept failed ...\n");
   exit(0);
 }
else
   printf ("server accept the client...\n");
   func (connfd);
   close (sockfd);
}
```

NAME : RUDRASISH MISHRA
ROLL NO: 1906619
SECTION : IT-8

CLIENT-CODE :

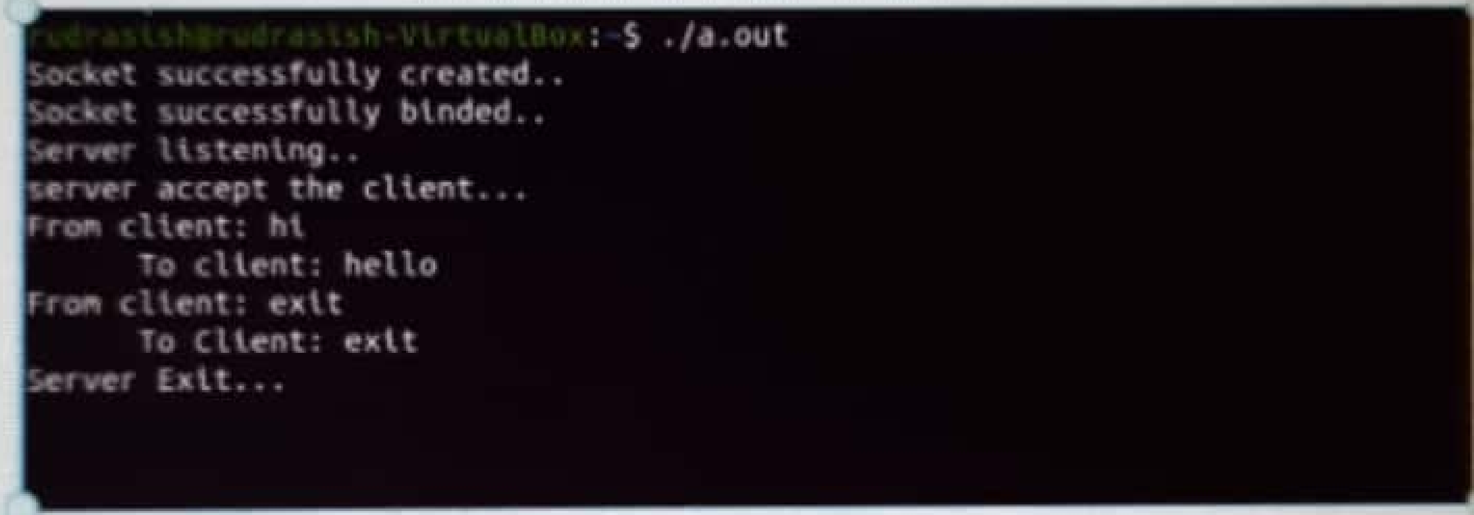
```
        #include <netdb.h>
        #include <stdio.h>
        # include <stlib.h>
        # include <string.h>
        # include <sys/socket.h>
        # define  MAX  80
        # define  PORT  8080
        # define  SA struct sockaddr
void func (int sockfd)
  {
           char buff[MAX];
           int n;
           for (;;)
        { bzero (buff, sizeof (buff));
          print (“Enter the string: ”);
          n = 0;
          while ((buff [n++] = getchar()) ! = '\n');
          write (sockfd, buff, sizeof (buff));
          bzero (buff, sizeof (buff));
          read (sockfd , buff , sizeof (buff));
        if ((strncmp (buff, "exit", 4)) = = 0)
          { print (“ Client Exit ...\n”);
          } break;
    }
}

        int main ()
  {
         int  sockfd ,connfd;
        Struct sockaddr_in servaddr, cli;
        sock fd = socket (AF_INET, SOCK_STREM,0);
      if (sockfd = = -1) { Print (“Socket creation failed ...\n”);
      }         exit (0);
      else

          Print (“ Socket succesfully 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)
      { print (“connection with the server failed ...\n”);
      } exit (0);
  else
       print (“connected to the server ...\n”);
  fun (sockfd);
  close (sockfd);
}
```

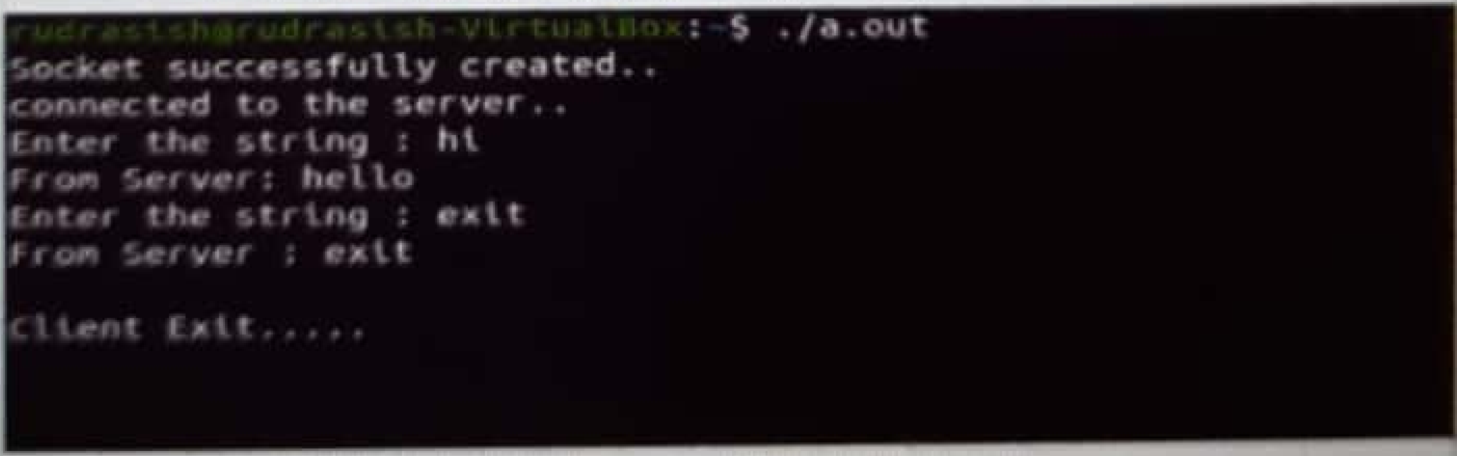
NAME : RUDRASISH [illegible]
ROLL NO: 1706619
SECTION : IT-8

SERVER OUTPUT :



```
rudrasish@rudrasish-VirtualBox:~$ ./a.out
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: hi
	 To client: hello
From client: exit
	 To Client: exit
Server Exit...
```

CLIENT OUTPUT :



```
rudrasish@rudrasish-VirtualBox:~$ ./a.out
Socket successfully created..
connected to the server..
Enter the string : hi
From Server: hello
Enter the string : exit
From Server : exit

Client Exit.....
```