**A client wants to establish a reliable connection to the server, which is on the same LAN. Demonstrate the concept with the relevant handshake communication.**

**AIM:** To establish a reliable connection between the server and the client which is on the same LAN using handshake communication.

**PROCEDURE:**

1. First we have to make a connection between the server and the client using code in ubuntu or python.
2. Now after writing the code for the server and client save the code.
3. To run the code we use the required command like for server we use ./server and for client we use ./client
4. After the connection is made we have to show the established connection by passing on the messages from the server and the client side.
5. The message will show the required connection between the server and the client.

**PROGRAM:**

**Server:**

#include <unistd.h>

#include <stdio.h>

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

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);

bzero(buff, MAX);

n = 0;

while ((buff[n++] = getchar()) != '\n')

;

write(sockfd, buff, sizeof(buff));

if (strncmp("exit", buff, 4) == 0)

{

printf("Server Exit...\n");

break;

}

}

}

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;

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

servaddr.sin\_port = htons(PORT);

if ((bind(sockfd, (SA \*)&servaddr, sizeof(servaddr))) != 0)

{

printf("socket bind failed...\n");

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 acccept failed...\n");

exit(0);

}

else

printf("server acccept the client...\n");

func(connfd);

close(sockfd);

}

**CLIENT:**

#include <unistd.h>

#include <netdb.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <arpa/inet.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));

printf("Enter the string : ");

n = 0;

while ((buff[n++] = getchar()) != '\n')

;

write(sockfd, buff, sizeof(buff));

bzero(buff, sizeof(buff));

read(sockfd, buff, sizeof(buff));

printf("From Server : %s", buff);

if ((strncmp(buff, "exit", 4)) == 0)

{

printf("Client Exit...\n");

break;

}

}

}

int main()

{

int sockfd, connfd;

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;

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("connection with the server failed...\n");

exit(0);

}

else

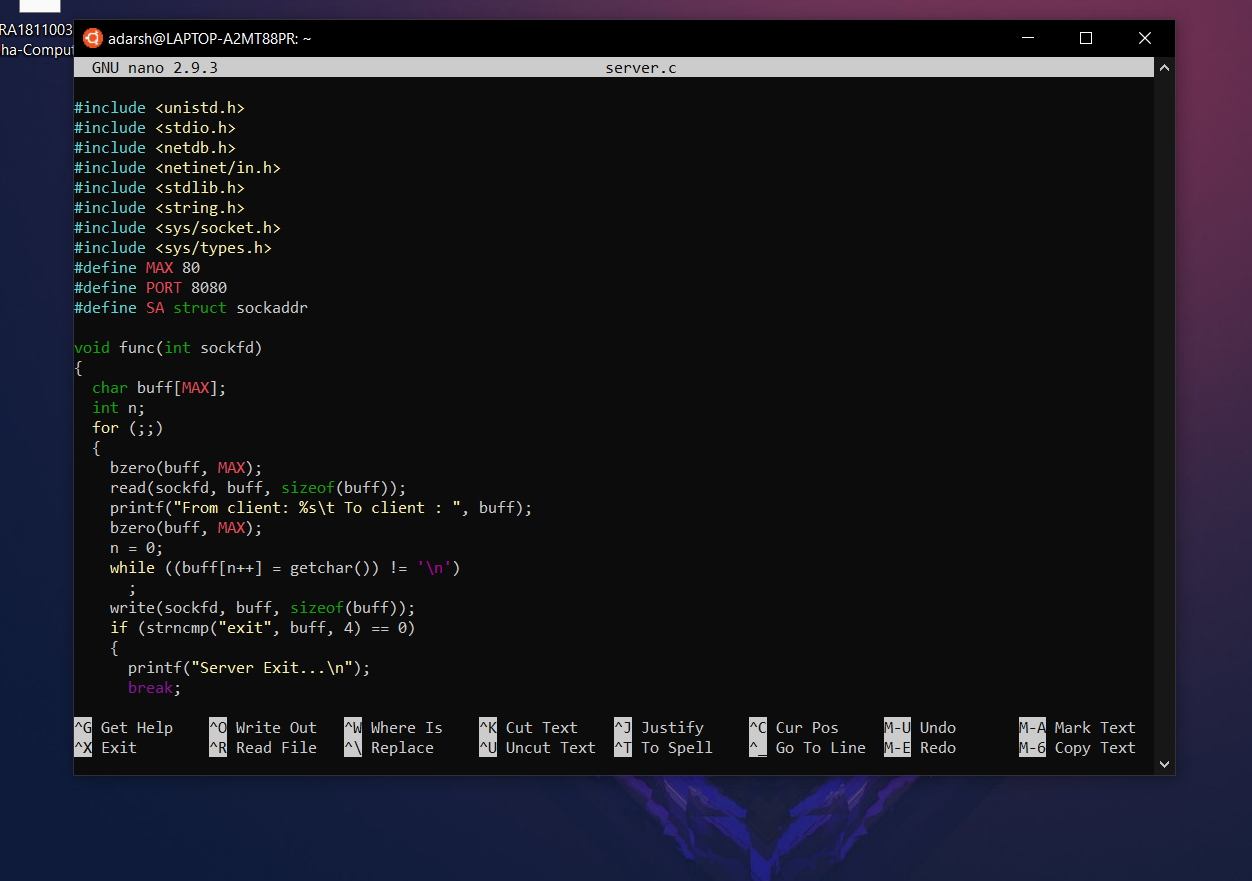
printf("connected to the server..\n");

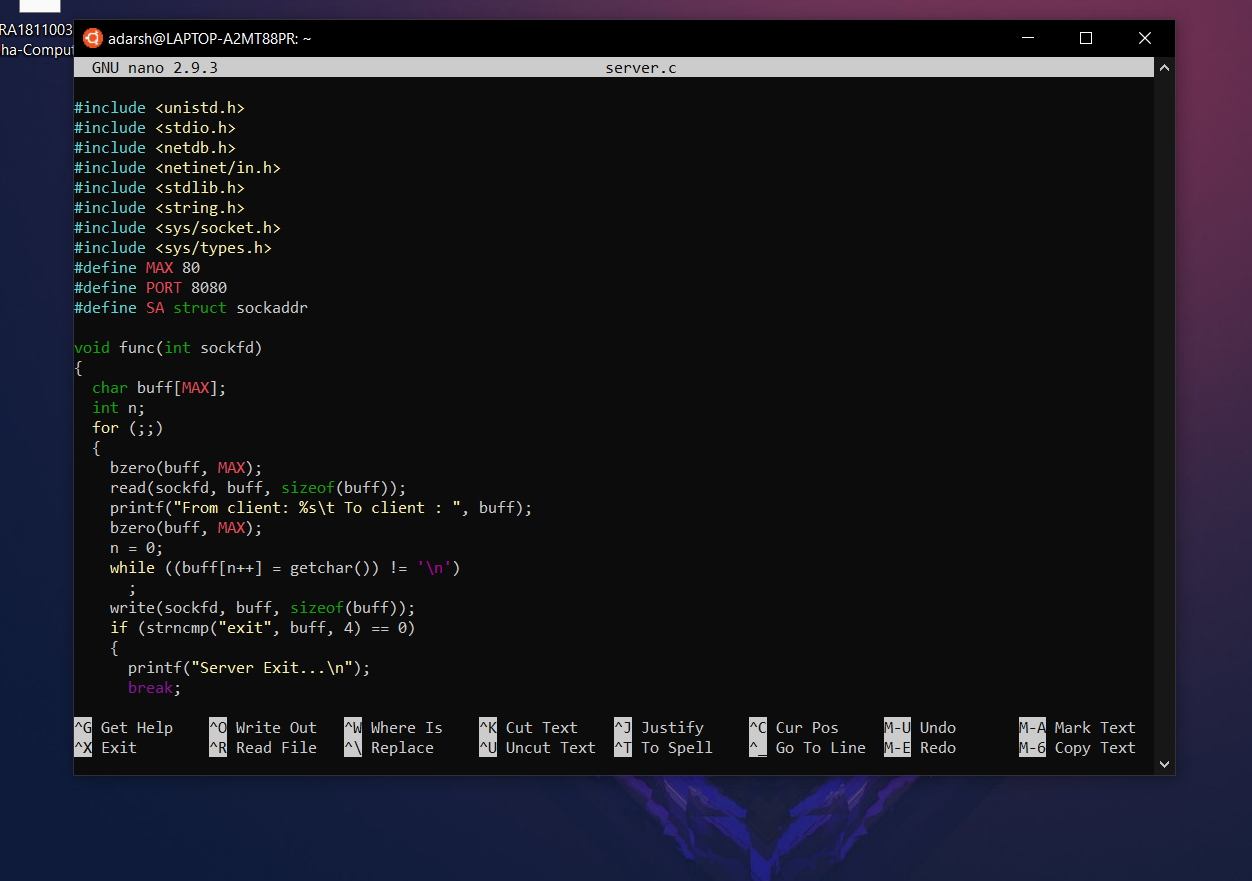
func(sockfd);

close(sockfd);

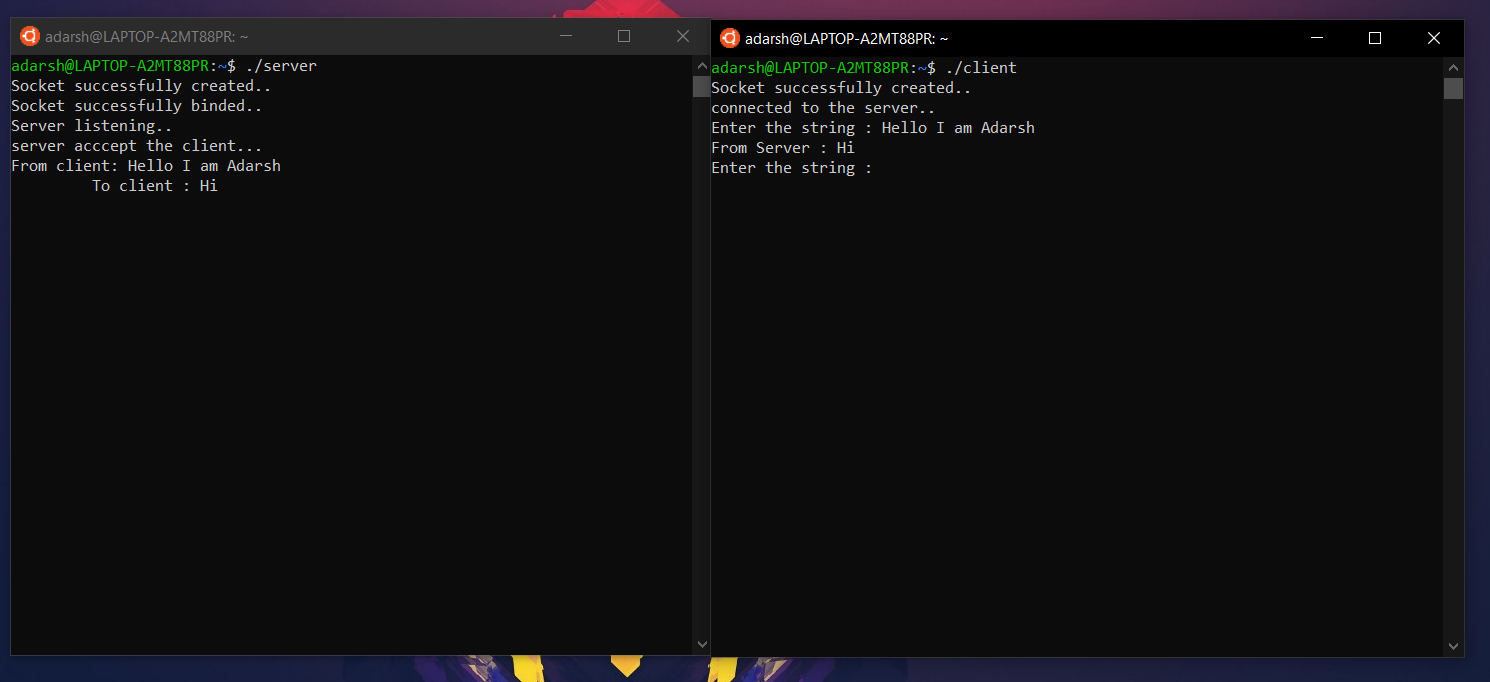
}

**SCREENSHOTS OF THE PROGRAM AND THE OUTPUT:**

****

****

**OUTPUT:**

****

**RESULT:**

The required TCP connection is made between the client and server which is on the same LAN using handshaking communication.