Swapnanil Dhol
RA1811031010010
CN EX 3- 01/September/2020

SIMPLE TCP/IP CLIENT SERVER COMMUNICATION

GIVEN REQUIREMENTS:

There are two hosts, Client and Server. The Client accepts the message from the user and sends it to the Server. The Server receives the message and prints it.

TECHNICAL OBJECTIVE:

To implement a simple TCP Client-Server application , where the Client on establishing a connection with the Server, sends a string to the Server. The Server reads the String and prints it.

METHODOLOGY:

Server:

- Include the necessary header files.
- Create a socket using socket function with family AF_INET, type as SOCK_STREAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin_family to AF_INET, sin_addr to INADDR_ANY, sin_port to a dynamically assigned port number.
- Bind the local host address to socket using the bind function.
- Listen on the socket for connection request from the client.
- Accept connection request from the client using accept function.
- Within an infinite loop, using the recv function receive message from the client and print it on the console.

Server Code

```
#include <stdio.h>

#include <netdb.h>

#include <unistd.h>

#include <arpa/inet.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 Exiting...\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... Exiting now.\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
    if ((listen(sockfd, 5)) != 0) {
        printf("Listening failed...\n");
        exit(0);
    } else
        printf("Server listening..\n");
len = sizeof(cli);
connfd = accept(sockfd, (SA*)&cli, &len);
if (connfd < 0) {
    printf("Server accepted failed...\n");
    exit(0); }
else
```

```
        printf("server accepted the client...\n");
    func(connfd);
    close(sockfd);
}
```

Client:

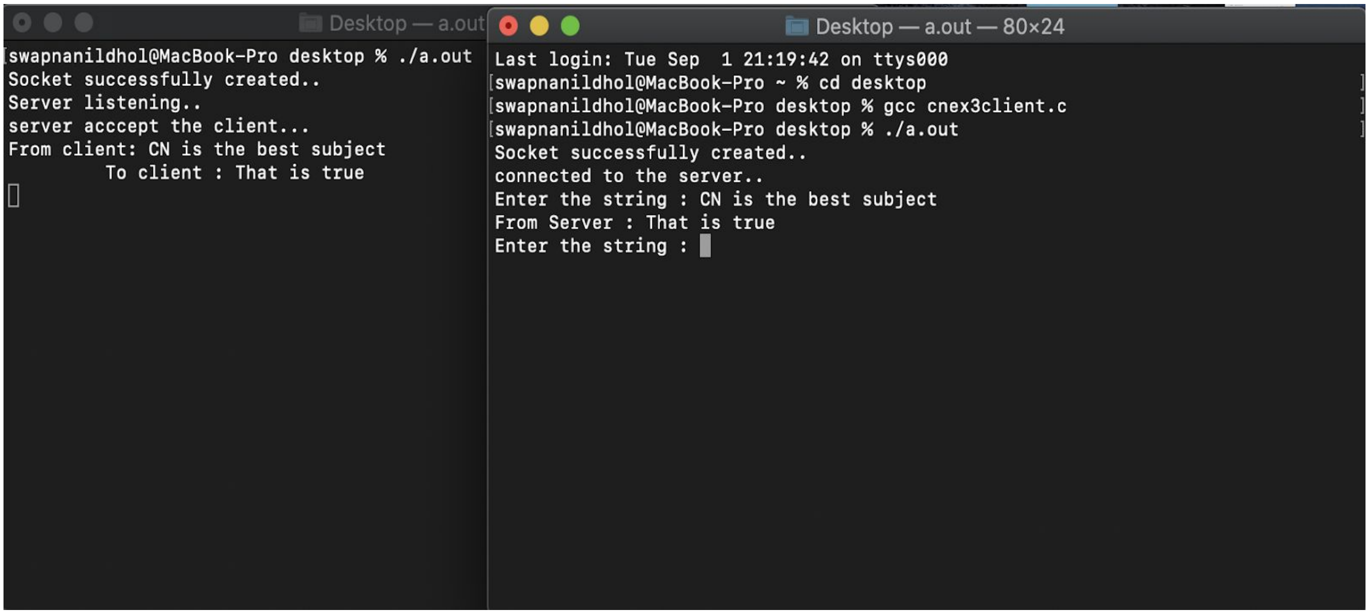
- Include the necessary header files.
- Create a socket using socket function with family AF_INET, type as SOCK_STREAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin_family to AF_INET.
- Get the server IP address and port number from the console.
- Using gethostbyname function assign it to a hostent structure, and assign it to sin_addr of the server address structure.
- Request a connection from the server using the connect function.
- Within an infinite loop, read message from the console and send the message to the server using the send function.

Client Code

```
#include<unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.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 crea2on 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("connec2on with the server failed...\n");

    exit(0);

  } else

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

close(sockfd);

}
```

Screenshots



```
[swapnanildhol@MacBook-Pro desktop % ./a.out
Socket successfully created..
Server listening..
server acccept the client...
From client: CN is the best subject
         To client : That is true
```

```
Last login: Tue Sep  1 21:19:42 on ttys000
[swapnanildhol@MacBook-Pro ~ % cd desktop
[swapnanildhol@MacBook-Pro desktop % gcc cnex3client.c
[swapnanildhol@MacBook-Pro desktop % ./a.out
Socket successfully created..
connected to the server..
Enter the string : CN is the best subject
From Server : That is true
Enter the string :
```