Gita Alekhya Paul **30/08/2021**

Computer Networks
RA1911030010014
Experiment - 7
Full Duplex Chat Using TCP/IP

Aim: **To create a Full Duplex Chat Using TCP/IP.**

Server Code:

```
/**
* RA1911030010014 - Full Duplex Chat Using TCP/IP Server
*/
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <string.h>

#define PORT 8014

int main(int argc, char *argv[])
{
   int clientSocketDescriptor, socketDescriptor;
   struct sockaddr_in serverAddress, clientAddress;
   socklen_t clientLength;
   char recvBuffer[1000], sendBuffer[1000];
   pid_t cpid;
   bzero(&serverAddress, sizeof(serverAddress));
   serverAddress.sin_family = AF_INET;
   serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
   serverAddress.sin_port = htons(PORT);
   socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
   bind(socketDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
   listen(socketDescriptor, 5);
   printf("Server is running on Port %d...\n", PORT);
   clientSocketDescriptor = accept(socketDescriptor, (struct sockaddr
*)&clientAddress, &clientLength);
   cpid = fork();

   if (cpid == 0)
   {
       while (1)
       {
           bzero(&recvBuffer, sizeof(recvBuffer));
           recv(clientSocketDescriptor, recvBuffer, sizeof(recvBuffer), 0);
           printf("\nCLIENT : %s\n", recvBuffer);
       }
   }
```

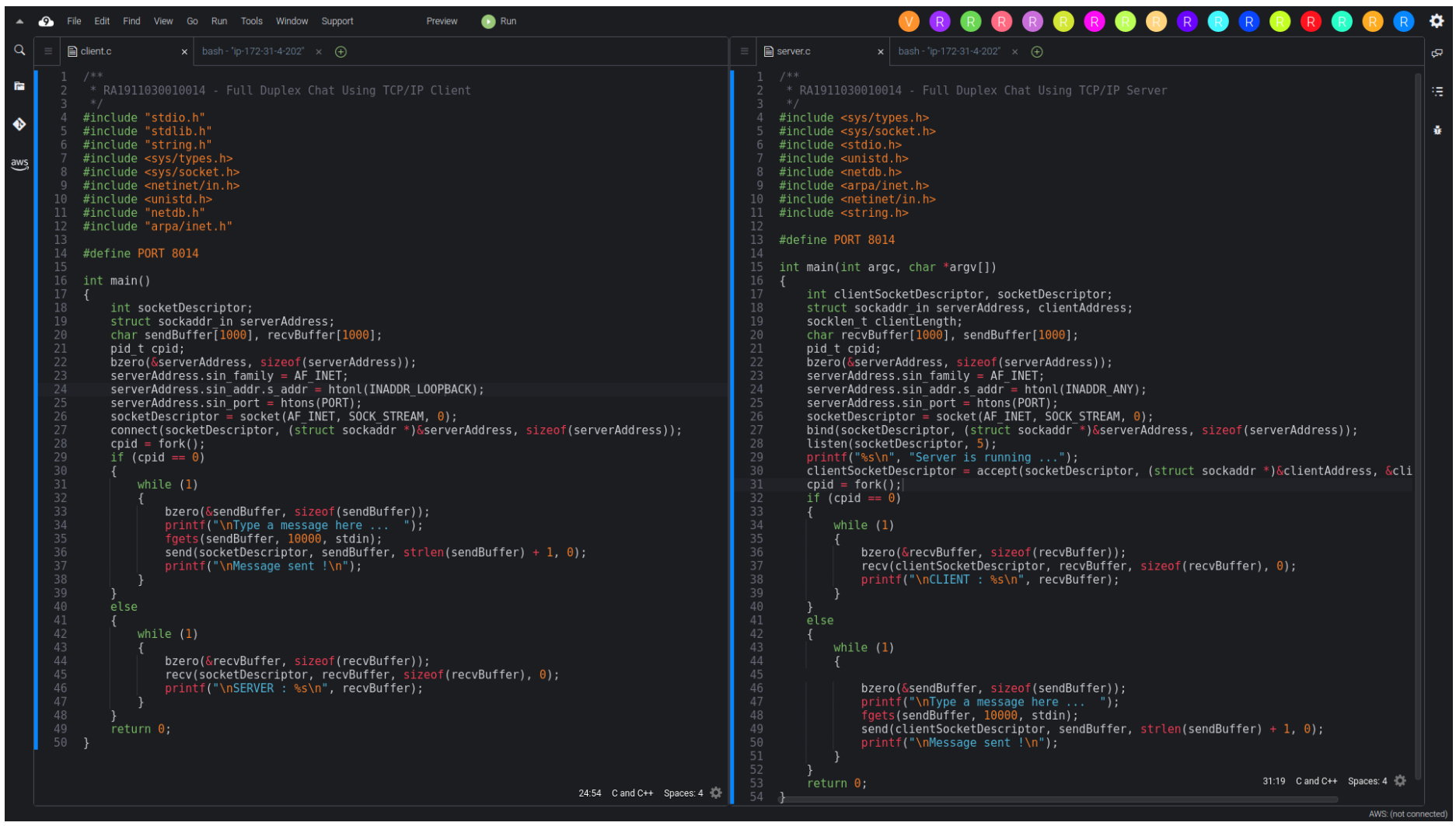
```
    else
    {
        while (1)
        {

            bzero(&sendBuffer, sizeof(sendBuffer));
            printf("\nType a message here ...  ");
            fgets(sendBuffer, 10000, stdin);
            send(clientSocketDescriptor, sendBuffer, strlen(sendBuffer) + 1, 0);
            printf("\nMessage sent !\n");
        }
    }
    return 0;
}
```

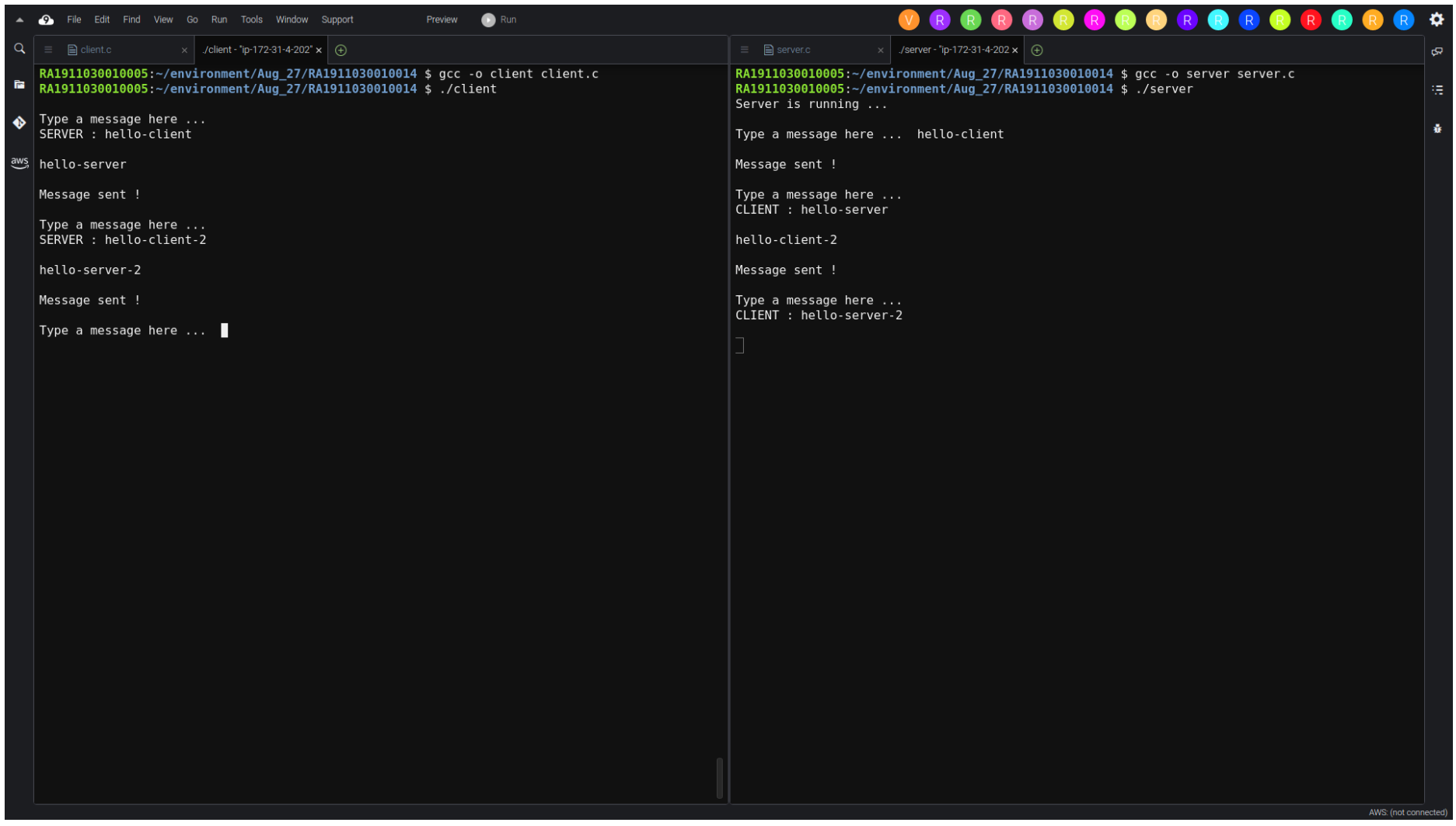
Client Code:

```
/**
* RA1911030010014 - Full Duplex Chat Using TCP/IP Client
*/
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "netdb.h"
#include "arpa/inet.h"

#define PORT 8014

int main()
{
    int socketDescriptor;
    struct sockaddr_in serverAddress;
    char sendBuffer[1000], recvBuffer[1000];
    pid_t cpid;
    bzero(&serverAddress, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = htonl(INADDR_LOOPBACK);
    serverAddress.sin_port = htons(PORT);
    socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
    connect(socketDescriptor, (struct sockaddr *)&serverAddress,
sizeof(serverAddress));
    cpid = fork();
    if (cpid == 0)
    {
        while (1)
        {
```

```
            bzero(&sendBuffer, sizeof(sendBuffer));
            printf("\nType a message here ...  ");
            fgets(sendBuffer, 10000, stdin);
            send(socketDescriptor, sendBuffer, strlen(sendBuffer) + 1, 0);
            printf("\nMessage sent !\n");
        }
    }
    else
    {
        while (1)
        {
            bzero(&recvBuffer, sizeof(recvBuffer));
            recv(socketDescriptor, recvBuffer, sizeof(recvBuffer), 0);
            printf("\nSERVER : %s\n", recvBuffer);
        }
    }
    return 0;
}
```

Output:





Result:

The required code for the Full Duplex Chat Using TCP/IP was written in the AWS Cloud9 environment and successfully compiled.