Gita Alekhya Paul **19/08/2021**

Computer Networks
RA1911030010014
Experiment - 6
Half Duplex Chat Using TCP/IP

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

Server Code:

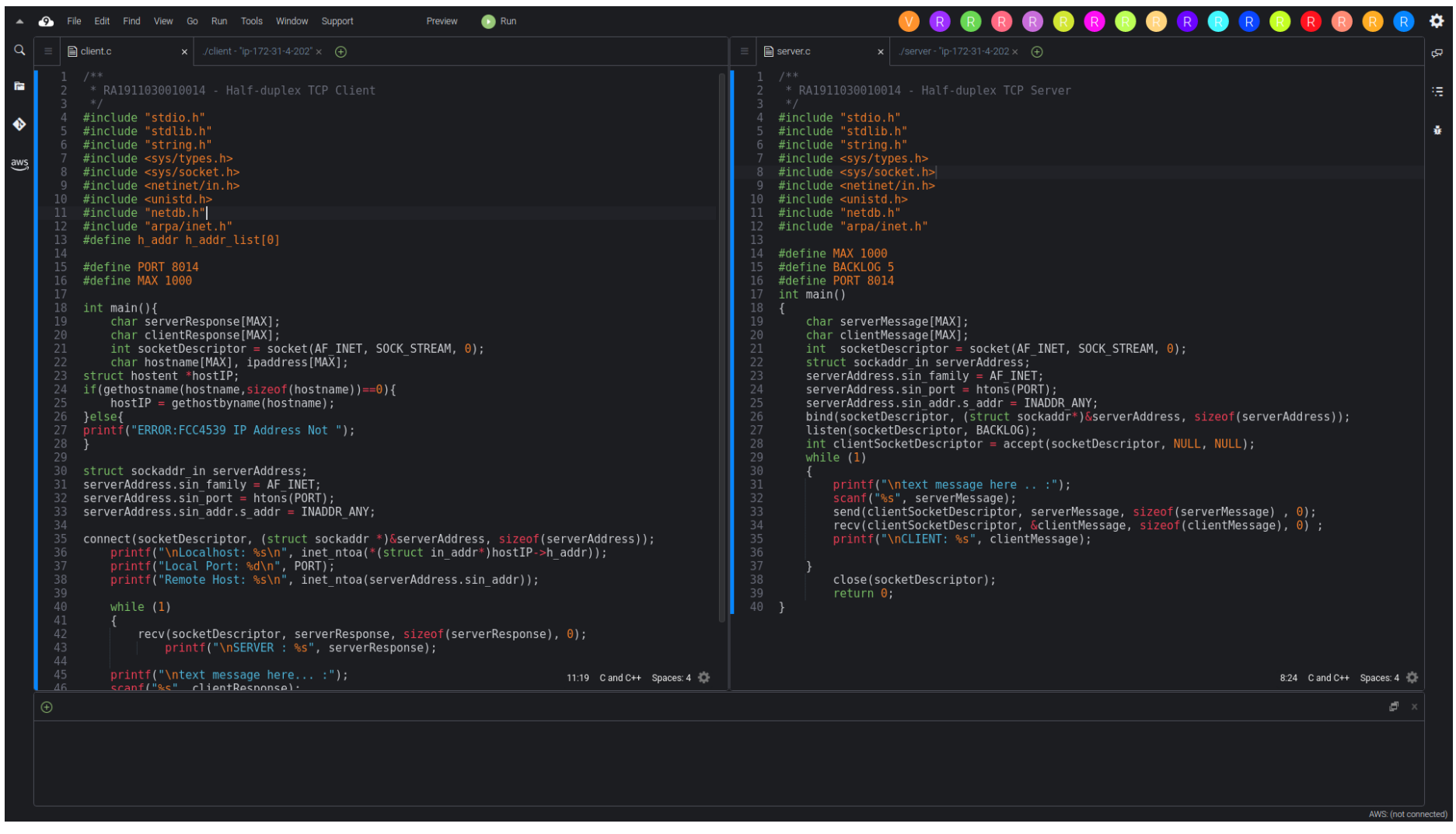
```c
/**
* RA1911030010014 - Half-duplex TCP Server
*/
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "netdb.h"
#include "arpa/inet.h"

#define MAX 1000
#define BACKLOG 5
#define PORT 8014
int main()
{
   char serverMessage[MAX];
   char clientMessage[MAX];
   int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
   struct sockaddr_in serverAddress;
   serverAddress.sin_family = AF_INET;
   serverAddress.sin_port = htons(PORT);
   serverAddress.sin_addr.s_addr = INADDR_ANY;
   bind(socketDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
   listen(socketDescriptor, BACKLOG);
   int clientSocketDescriptor = accept(socketDescriptor, NULL, NULL);
   while (1)
   {
       printf("\ntext message here .. :");
       scanf("%s", serverMessage);
       send(clientSocketDescriptor, serverMessage, sizeof(serverMessage), 0);
       recv(clientSocketDescriptor, &clientMessage, sizeof(clientMessage), 0);
       printf("\nCLIENT: %s", clientMessage);
   }
   close(socketDescriptor);
   return 0;
}
```

Client Code:

```c
/**
* RA1911030010014 - Half-duplex TCP 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 h_addr h_addr_list[0]

#define PORT 8014
#define MAX 1000

int main()
{
   char serverResponse[MAX];
   char clientResponse[MAX];
   int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
   char hostname[MAX], ipaddress[MAX];
   struct hostent *hostIP;
   if (gethostname(hostname, sizeof(hostname)) == 0)
   {
       hostIP = gethostbyname(hostname);
   }
   else
   {
       printf("ERROR:FCC4539 IP Address Not ");
   }

   struct sockaddr_in serverAddress;
   serverAddress.sin_family = AF_INET;
   serverAddress.sin_port = htons(PORT);
   serverAddress.sin_addr.s_addr = INADDR_ANY;

   connect(socketDescriptor, (struct sockaddr *)&serverAddress,
sizeof(serverAddress));
   printf("\nLocalhost: %s\n", inet_ntoa(*(struct in_addr *)hostIP->h_addr));
   printf("Local Port: %d\n", PORT);
   printf("Remote Host: %s\n", inet_ntoa(serverAddress.sin_addr));

   while (1)
   {
       recv(socketDescriptor, serverResponse, sizeof(serverResponse), 0);
       printf("\nSERVER : %s", serverResponse);
```

```
        printf("\ntext message here... :");
        scanf("%s", clientResponse);
        send(socketDescriptor, clientResponse, sizeof(clientResponse), 0);
    }
    close(socketDescriptor);
    return 0;
}
```

Output:



```c
struct sockaddr_in serverAddress;
serverAddress.sin_family = AF_INET;
serverAddress.sin_port = htons(PORT);
serverAddress.sin_addr.s_addr = INADDR_ANY;

connect(socketDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    printf("\nLocalhost: %s\n", inet_ntoa(*(struct in_addr*)hostIP->h_addr));
    printf("Local Port: %d\n", PORT);
    printf("Remote Host: %s\n", inet_ntoa(serverAddress.sin_addr));

    while (1)
    {
        recv(socketDescriptor, serverResponse, sizeof(serverResponse), 0);
            printf("\nSERVER : %s", serverResponse);

    printf("\ntext message here... :");
    scanf("%s", clientResponse);
    send(socketDescriptor, clientResponse, sizeof(clientResponse), 0);
    }
    close(socketDescriptor);
    return 0;
}
```

```c
/**
 * RA1911030010014 - Half-duplex TCP Server
 */
#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 MAX 1000
#define BACKLOG 5
#define PORT 8014
int main()
{
    char serverMessage[MAX];
    char clientMessage[MAX];
    int  socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_port = htons(PORT);
    serverAddress.sin_addr.s_addr = INADDR_ANY;
    bind(socketDescriptor, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
    listen(socketDescriptor, BACKLOG);
    int clientSocketDescriptor = accept(socketDescriptor, NULL, NULL);
    while (1)
    {
        printf("\ntext message here .. :");
        scanf("%s", serverMessage);
        send(clientSocketDescriptor, serverMessage, sizeof(serverMessage) , 0);
        recv(clientSocketDescriptor, &clientMessage, sizeof(clientMessage), 0) ;
        printf("\nCLIENT: %s", clientMessage);

    }
        close(socketDescriptor);
        return 0;
}
```

```
Vhemamalini:~/environment/Aug_19/RA1911030010014 $ gcc -o client client.c
Vhemamalini:~/environment/Aug_19/RA1911030010014 $ ./client

Localhost: 172.31.4.202
Local Port: 8014
Remote Host: 0.0.0.0

SERVER : hi-from-server
text message here... :hi-from-client

SERVER : again-server
text message here... :again-client
```

```
Vhemamalini:~/environment/Aug_19/RA1911030010014 $ gcc -o server server.c
Vhemamalini:~/environment/Aug_19/RA1911030010014 $ ./server

text message here .. :hi-from-server

CLIENT: hi-from-client
text message here .. :again-server

CLIENT: again-client
text message here .. :
```

Result:

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