Exp No: 4

Date: 12/09/2020

CHAT USING TCP

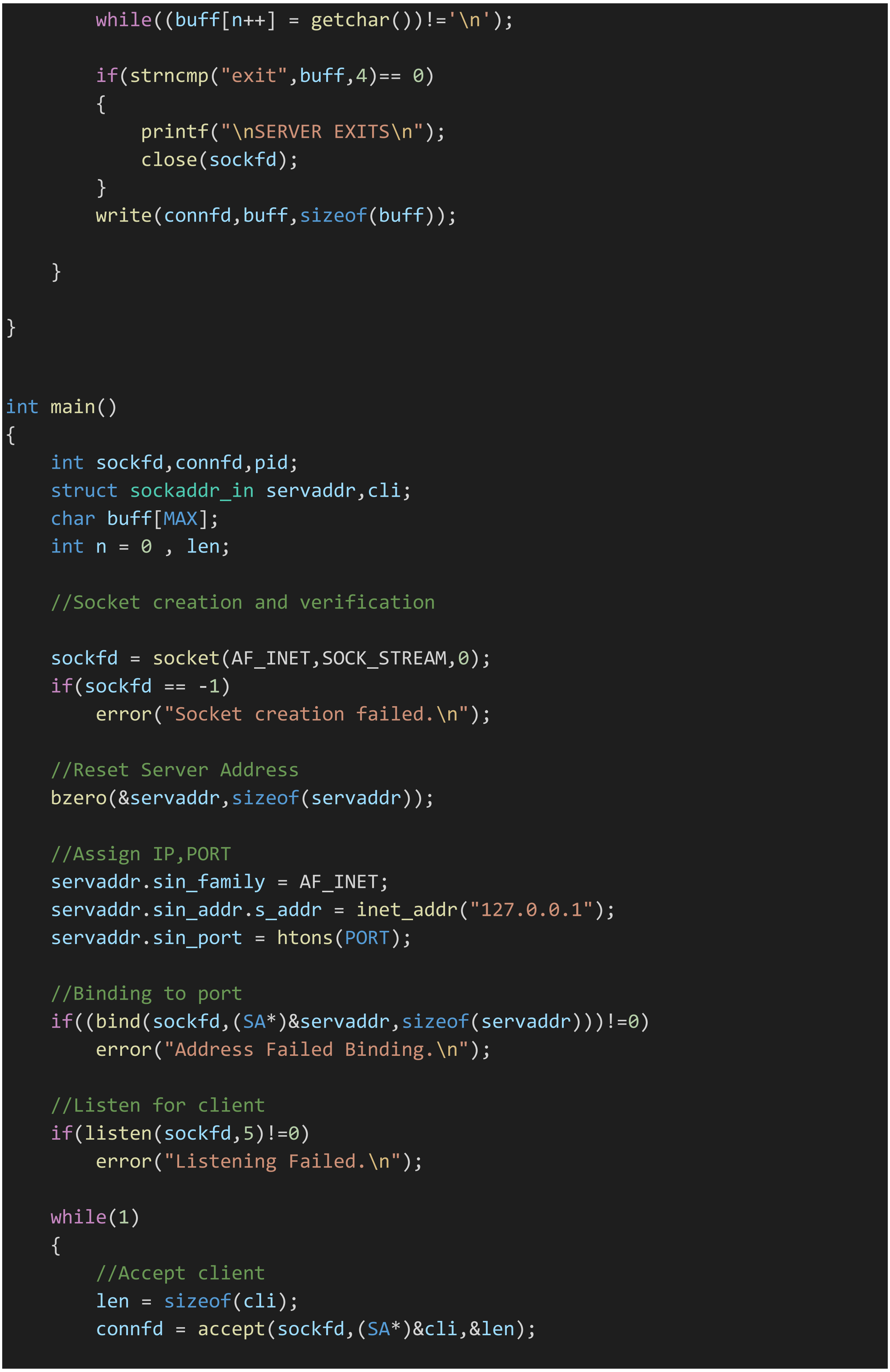
Name: Swetha Saseendran

Register Number: 185001183

Code:

SERVER.C

```
#include<netdb.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/socket.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr

int clientCount = 1;

void error(char *msg)
{
    perror(msg);
    exit(1);
}

void chat(int sockfd,int connfd,int clientIndex)

{
    char buff[MAX];
    int n;
    while(1)
    {
        //Clear buffer
        bzero(buff,sizeof(buff));

        //Read message from client
        read(connfd,buff,sizeof(buff));
        if(strncmp("exit",buff,4)== 0)
        {
            printf("\nCLIENT %d LEFT THE CHAT\n",clientIndex);
            break;
        }
        else
            printf("\nClient %d: %s\n",clientIndex,buff);

        //Write message to client
        n = 0;
        bzero(buff,MAX);
        printf("Server: ");
```

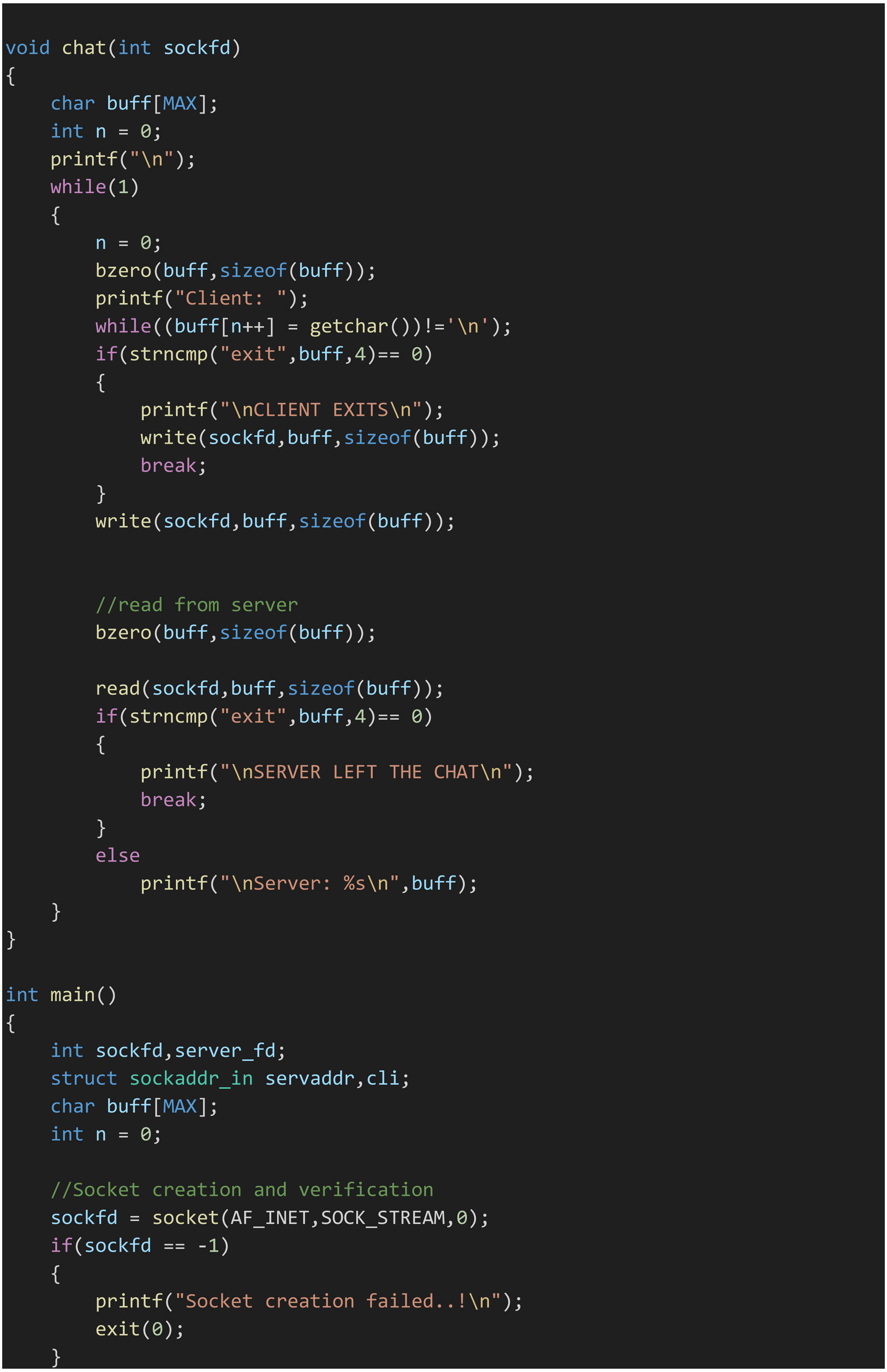


```
        while((buff[n++] = getchar())!='\n');

        if(strncmp("exit",buff,4)== 0)
        {
            printf("\nSERVER EXITS\n");
            close(sockfd);
        }
        write(connfd,buff,sizeof(buff));

    }

}


int main()
{
    int sockfd,connfd,pid;
    struct sockaddr_in servaddr,cli;
    char buff[MAX];
    int n = 0 , len;

    //Socket creation and verification

    sockfd = socket(AF_INET,SOCK_STREAM,0);
    if(sockfd == -1)
        error("Socket creation failed.\n");

    //Reset Server Address
    bzero(&servaddr,sizeof(servaddr));

    //Assign IP,PORT
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);

    //Binding to port
    if((bind(sockfd,(SA*)&servaddr,sizeof(servaddr)))!=0)
        error("Address Failed Binding.\n");

    //Listen for client
    if(listen(sockfd,5)!=0)
        error("Listening Failed.\n");

    while(1)
    {
        //Accept client
        len = sizeof(cli);
        connfd = accept(sockfd,(SA*)&cli,&len);
```
]

```
       if(connfd<0)
            error("Not Accepted.\n");
        else
            printf("\nCLIENT %d ENTERED THE CHAT\n",clientCount);

        pid = fork();
        if (pid == -1) //error
        {
            close(connfd);
            error("Error in creating process\n");
            continue;
        }
        else if(pid > 0)
        {
            close(connfd);
            clientCount++;
            continue;
        }
        else if(pid ==0){
            //Chat function
            chat(sockfd,connfd,clientCount);
            //close(connfd);
            bzero(buff,MAX);
            break;
        }

        close(sockfd);
    }

}
```

CLIENT.C

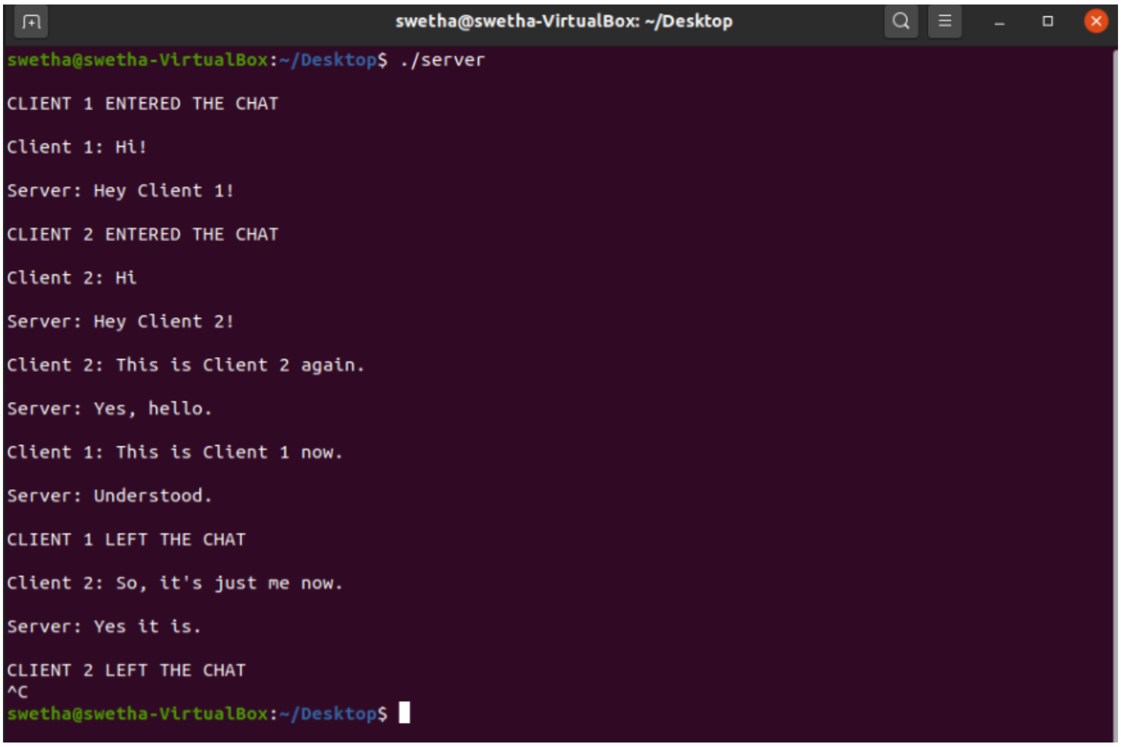
```
#include<netdb.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/socket.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr


void error(char *msg)
{
    perror(msg);
    exit(1);
}
```



```
void chat(int sockfd)
{
    char buff[MAX];
    int n = 0;
    printf("\n");
    while(1)
    {
        n = 0;
        bzero(buff,sizeof(buff));
        printf("Client: ");
        while((buff[n++] = getchar())!='\n');
        if(strncmp("exit",buff,4)== 0)
        {
            printf("\nCLIENT EXITS\n");
            write(sockfd,buff,sizeof(buff));
            break;
        }
        write(sockfd,buff,sizeof(buff));


        //read from server
        bzero(buff,sizeof(buff));

        read(sockfd,buff,sizeof(buff));
        if(strncmp("exit",buff,4)== 0)
        {
            printf("\nSERVER LEFT THE CHAT\n");
            break;
        }
        else
            printf("\nServer: %s\n",buff);
    }
}

int main()
{
    int sockfd,server_fd;
    struct sockaddr_in servaddr,cli;
    char buff[MAX];
    int n = 0;

    //Socket creation and verification
    sockfd = socket(AF_INET,SOCK_STREAM,0);
    if(sockfd == -1)
    {
        printf("Socket creation failed..!\n");
        exit(0);
    }
    bzero(&servaddr,sizeof(servaddr));

    //Assign IP,PORT

    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);

    //Connection

    if(connect(sockfd,(SA*)&servaddr,sizeof(servaddr))!=0)
    {
        printf("Connection Failed..\n");
        exit(0);
    }


    //Chat function
    chat(sockfd);

    close(sockfd);
}
```

Sample Input Output:

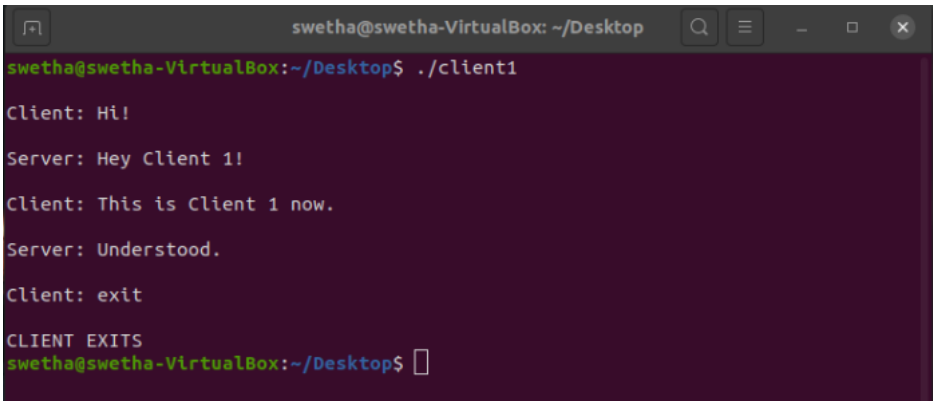
SERVER SIDE:



```
swetha@swetha-VirtualBox:~/Desktop$ ./server

CLIENT 1 ENTERED THE CHAT

Client 1: Hi!

Server: Hey Client 1!

CLIENT 2 ENTERED THE CHAT

Client 2: Hi

Server: Hey Client 2!

Client 2: This is Client 2 again.

Server: Yes, hello.

Client 1: This is Client 1 now.

Server: Understood.

CLIENT 1 LEFT THE CHAT

Client 2: So, it's just me now.

Server: Yes it is.

CLIENT 2 LEFT THE CHAT
^C
swetha@swetha-VirtualBox:~/Desktop$
```

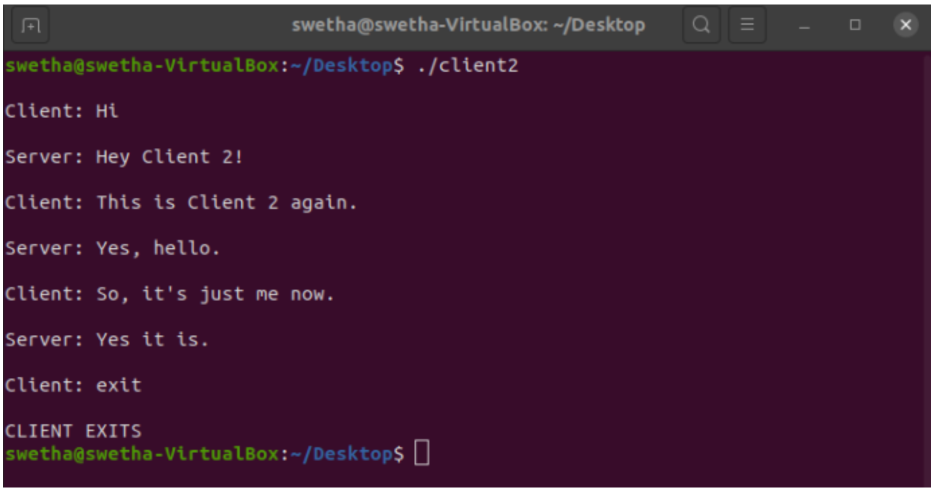
CLIENT SIDE:

Client 1:



```
swetha@swetha-VirtualBox:~/Desktop$ ./client1

Client: Hi!

Server: Hey Client 1!

Client: This is Client 1 now.

Server: Understood.

Client: exit

CLIENT EXITS
swetha@swetha-VirtualBox:~/Desktop$ 
```

Client 2:



```
swetha@swetha-VirtualBox:~/Desktop$ ./client2

Client: Hi

Server: Hey Client 2!

Client: This is Client 2 again.

Server: Yes, hello.

Client: So, it's just me now.

Server: Yes it is.

Client: exit

CLIENT EXITS
swetha@swetha-VirtualBox:~/Desktop$ 
```