RA1911003010386
MAYANK SINHA

SERVER:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
//headers for socket and related functions
#include <sys/types.h>
#include <sys/socket.h>
//for including structures which will store information needed
#include <netinet/in.h>
#include <unistd.h>
//for gethostbyname
#include "netdb.h"
#include "arpa/inet.h"
#define MAX 1000
#define BACKLOG 5 // how many pending connections queue will hold
int main()
{
char serverMessage[MAX];
char clientMessage[MAX];
//create the server socket
int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
struct sockaddr_in serverAddress;
serverAddress.sin_family = AF_INET;
serverAddress.sin_port = htons(9086);
serverAddress.sin_addr.s_addr = INADDR_ANY;
//calling bind function to oir specified IP and port
bind(socketDescriptor, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
listen(socketDescriptor, BACKLOG);
//starting the accepting
int clientSocketDescriptor = accept(socketDescriptor, NULL, NULL);
while (1)
{
printf("\ntext message here .. :");
scanf("%s", serverMessage);
send(clientSocketDescriptor, serverMessage, sizeof(serverMessage) , 0);
//recieve the data from the server
recv(clientSocketDescriptor, &clientMessage, sizeof(clientMessage), 0) ;
//recieved data from the server successfully then printing the data obtained from the server
printf("\nCLIENT: %s", clientMessage);
}
```

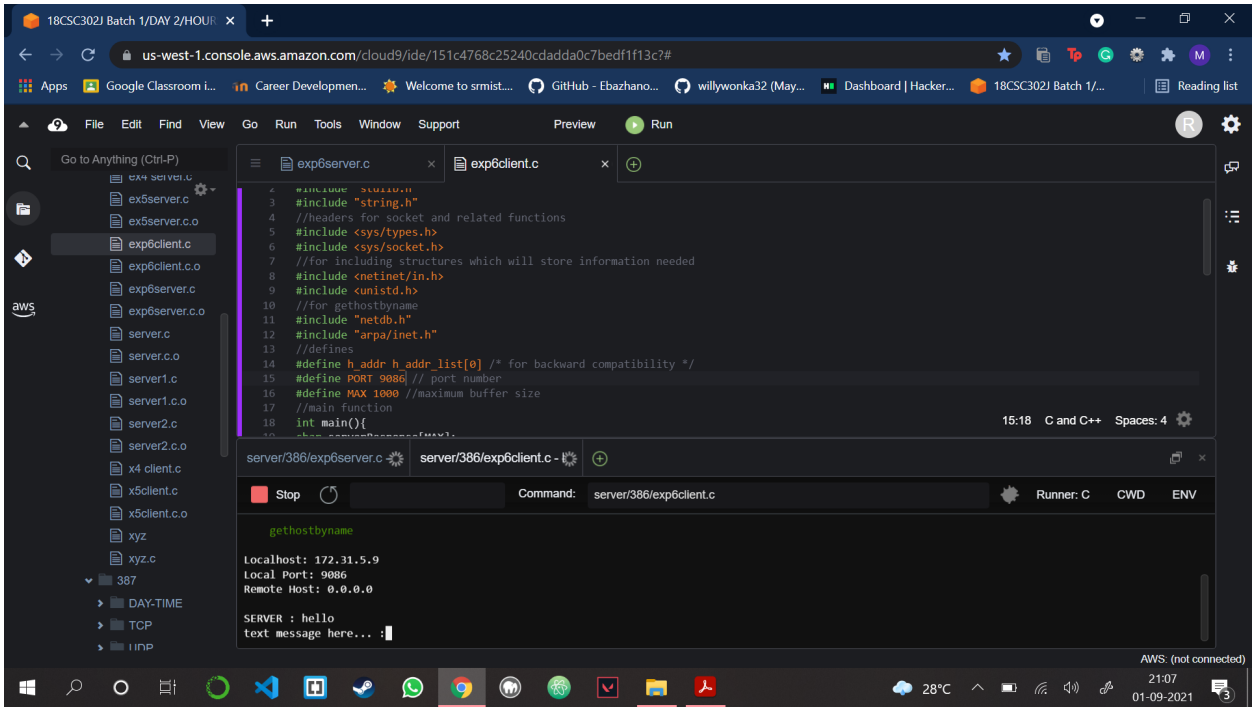
```
//close the socket
close(socketDescriptor);
return 0;
}
```

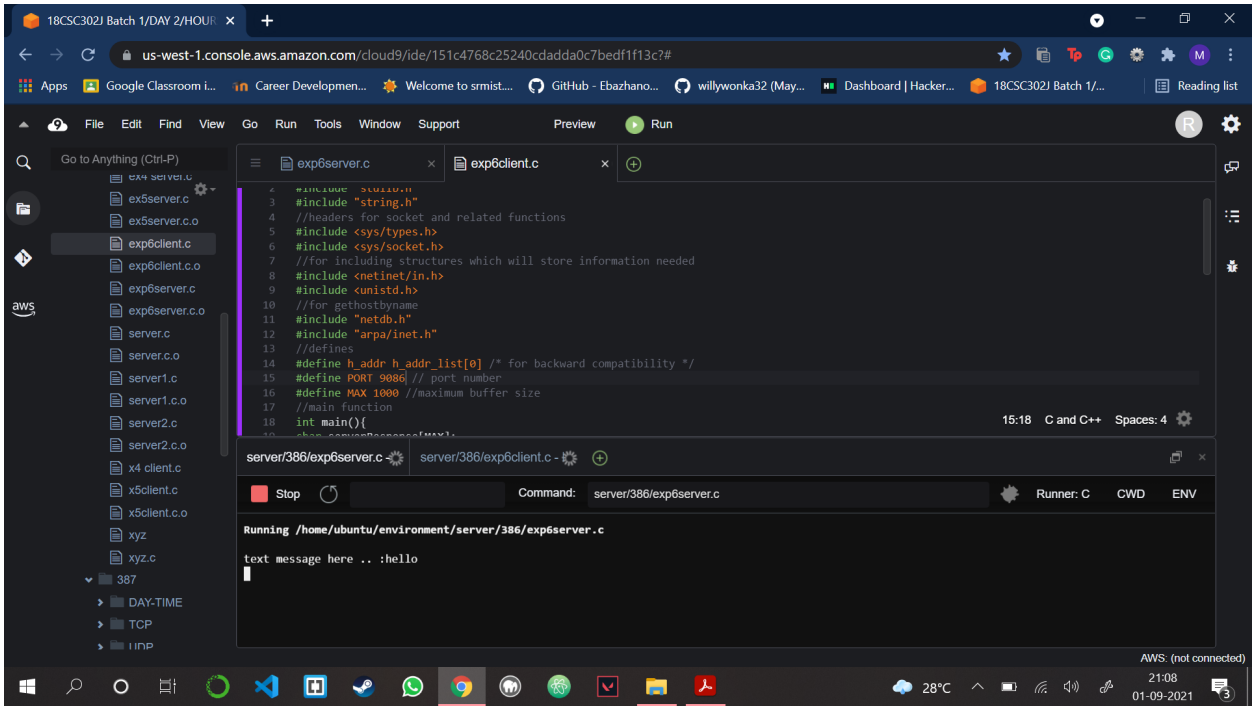
CLIENT:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
//headers for socket and related functions
#include <sys/types.h>
#include <sys/socket.h>
//for including structures which will store information needed
#include <netinet/in.h>
#include <unistd.h>
//for gethostbyname
#include "netdb.h"
#include "arpa/inet.h"
//defines
#define h_addr h_addr_list[0] /* for backward compatibility */
#define PORT 9086 // port number
#define MAX 1000 //maximum buffer size
//main function
int main(){
char serverResponse[MAX];
char clientResponse[MAX];
//creating a socket
int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
//placeholder for the hostname and my ip address
char hostname[MAX], ipaddress[MAX];
struct hostent *hostIP; //placeholder for the ip address
//if the gethostname returns a name then the program will get the ip address
if(gethostname(hostname,sizeof(hostname))==0){
hostIP = gethostbyname(hostname);//the netdb.h fucntion gethostbyname
}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));
// getting the address port and remote host
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)
{ //recieve the data from the server
recv(socketDescriptor, serverResponse, sizeof(serverResponse), 0);
//recieved data from the server successfully then printing the data obtained from the server
printf("\nSERVER : %s", serverResponse);
printf("\ntext message here... :");
scanf("%s", clientResponse);
send(socketDescriptor, clientResponse, sizeof(clientResponse), 0);
}
//closing the socket
close(socketDescriptor);
return 0;
}
```

OUTPUT:




18CSC302J Batch 1/DAY 2/HOUR
us-west-1.console.aws.amazon.com/cloud9/ide/151c4768c25240cdadda0c7bedf1f13c?#
Apps
Google Classroom i...
Career Developmen...
Welcome to srmist....
GitHub - Ebazhano...
willywonka32 (May...
Dashboard | Hacker...
18CSC302J Batch 1/...
Reading list
File Edit Find View Go Run Tools Window Support
Preview
Run
Go to Anything (Ctrl-P)
ex5server.c
ex5server.c.o
exp6client.c
exp6client.c.o
exp6server.c
exp6server.c.o
server.c
server.c.o
server1.c
server1.c.o
server2.c
server2.c.o
x4 client.c
x5client.c
x5client.c.o
xyz
xyz.c
387
DAY-TIME
TCP
UDP
exp6server.c
exp6client.c
#include "string.h"
//headers for socket and related functions
#include <sys/types.h>
#include <sys/socket.h>
//for including structures which will store information needed
#include <netinet/in.h>
#include <unistd.h>
//for gethostbyname
#include "netdb.h"
#include "arpa/inet.h"
//defines
#define h_addr h_addr_list[0] /* for backward compatibility */
#define PORT 9086 // port number
#define MAX 1000 //maximum buffer size
//main function
int main(){
15:18 C and C++ Spaces: 4
server/386/exp6server.c -
server/386/exp6client.c -
Stop
Command: server/386/exp6server.c
Runner: C CWD ENV
Running /home/ubuntu/environment/server/386/exp6server.c
text message here .. :hello
AWS: (not connected)
28°C
21:08
01-09-2021