

MAYANK SINHA

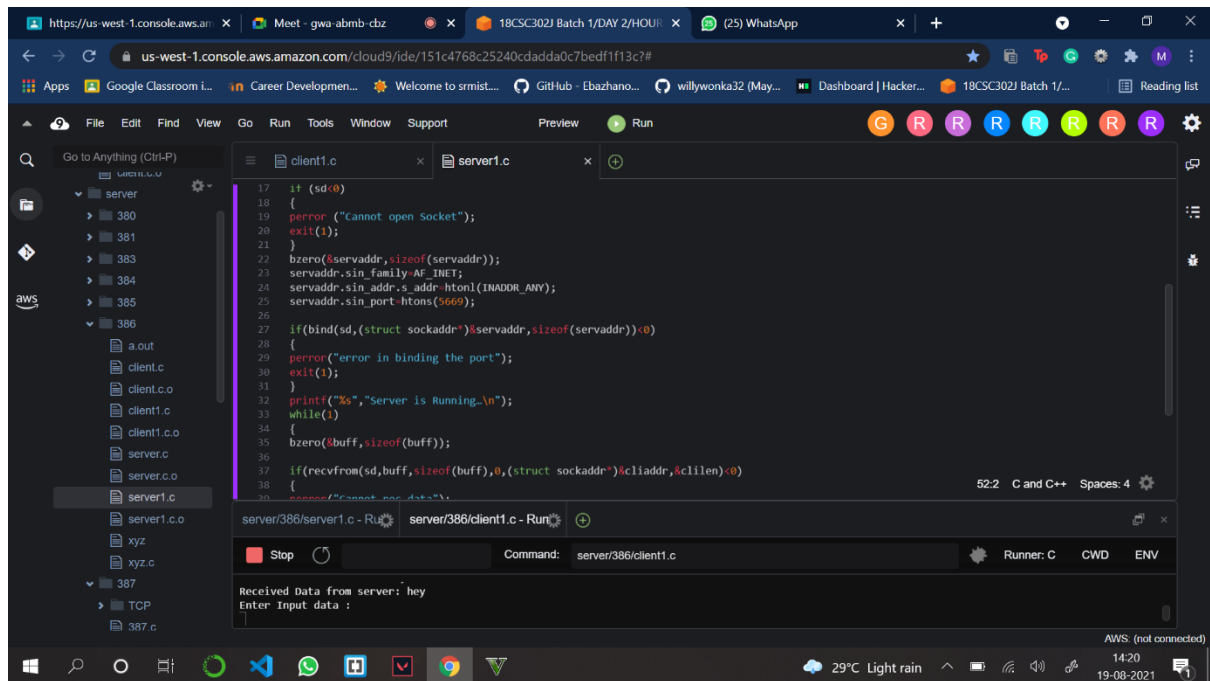
RA1911003010386

SERVER:

```
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<string.h>
#include<netinet/in.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<sys/types.h>
int main(int argc,char *argv[])
{
int sd;
char buff[1024];
struct sockaddr_in cliaddr,servaddr;
socklen_t clilen;
clilen=sizeof(cliaddr);
sd=socket(AF_INET,SOCK_DGRAM,0);
if (sd<0)
{
perror ("Cannot open Socket");
exit(1);
}
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(5669);
```

```
if(bind(sd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
{
perror("error in binding the port");
exit(1);
}
printf("%s","Server is Running...\n");
while(1)
{
bzero(&buff,sizeof(buff));

if(recvfrom(sd,buff,sizeof(buff),0,(struct sockaddr*)&cliaddr,&clilen)<0)
{
perror("Cannot rec data");
exit(1);
}
printf("Message is received \n",buff);
if(sendto(sd,buff,sizeof(buff),0,(struct sockaddr*)&cliaddr,clilen)<0)
{
perror("Cannot send data to client");
exit(1);
}
printf("Send data to UDP Client: %s",buff);
}
close(sd);
return 0;
}
```



CLIENT:

```
#include<sys/types.h>
```

```
#include<sys/socket.h>
```

```
#include<stdio.h>
```

```
#include<unistd.h>
```

```
#include<string.h>
```

```
#include<netinet/in.h>
```

```
#include<netdb.h>
```

```
int main(int argc,char*argv[])
```

```
{
```

```
int sd;
```

```
char buff[1024];
```

```
struct sockaddr_in servaddr;
```

```
socklen_t len;
```

```
len=sizeof(servaddr);
```

```
sd = socket(AF_INET,SOCK_DGRAM,0);
```

```
if(sd<0)
```

```

{
perror("Cannot open socket");
exit(1);
}

bzero(&servaddr,len);
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(5669);
while(1)
{
printf("Enter Input data : \n");
bzero(buff,sizeof(buff));
fgets(buff,sizeof (buff),stdin);
if(sendto (sd,buff,sizeof (buff),0,(struct sockaddr*)&servaddr,len)<0)
{
perror("Cannot send data");
exit(1);
}
printf("Data sent to UDP Server:%s",buff);
bzero(buff,sizeof(buff));
if(recvfrom (sd,buff,sizeof(buff),0,(struct sockaddr*)&servaddr,&len)<0)
{
perror("Cannot receive data");
exit(1);
}
printf("Received Data from server: %s",buff);
}
close(sd);
return 0;
}

```

us-west-1.console.aws.amazon.com/cloud9/ide/151c4768c25240cdadda0c7bedf1f13c?#

File Edit Find View Go Run Tools Window Support Preview Run

server

- 380
- 381
- 383
- 384
- 385
- 386
 - a.out
 - client.c
 - client.c.o
 - client1.c
 - client1.c.o
 - server.c
 - server.c.o
 - server1.c
 - server1.c.o
 - xyz
 - xyz.c
- 387
 - TCP
 - 387.c

```
27 {
28     printf("Enter Input data : \n");
29     bzero(buff,sizeof(buff));
30     fgets(buff,sizeof(buff),stdin);
31     if(sendto (sd,buff,sizeof (buff),0,(struct sockaddr*)&servaddr,len)<0)
32     {
33         perror("Cannot send data");
34         exit(1);
35     }
36     printf("Data sent to UDP Server:%s",buff);
37     bzero(buff,sizeof(buff));
38     if(recvfrom (sd,buff,sizeof (buff),0,(struct sockaddr*)&servaddr,&len)<0)
39     {
40         perror("Cannot receive data");
41         exit(1);
42     }
43 }
```

server/386/server1.c - Run server/386/client1.c - Run

Stop Command: server/386/client1.c Runner: C CWD ENV

Enter Input data :
hello
Data sent to UDP Server:hello
Received Data from server: hello
Enter Input data :
hey
Data sent to UDP Server:hey
Received Data from server: hey
Enter Input data :

47:2 C and C++ Spaces: 4

AWS: (not connected) 14:19 19-08-2021