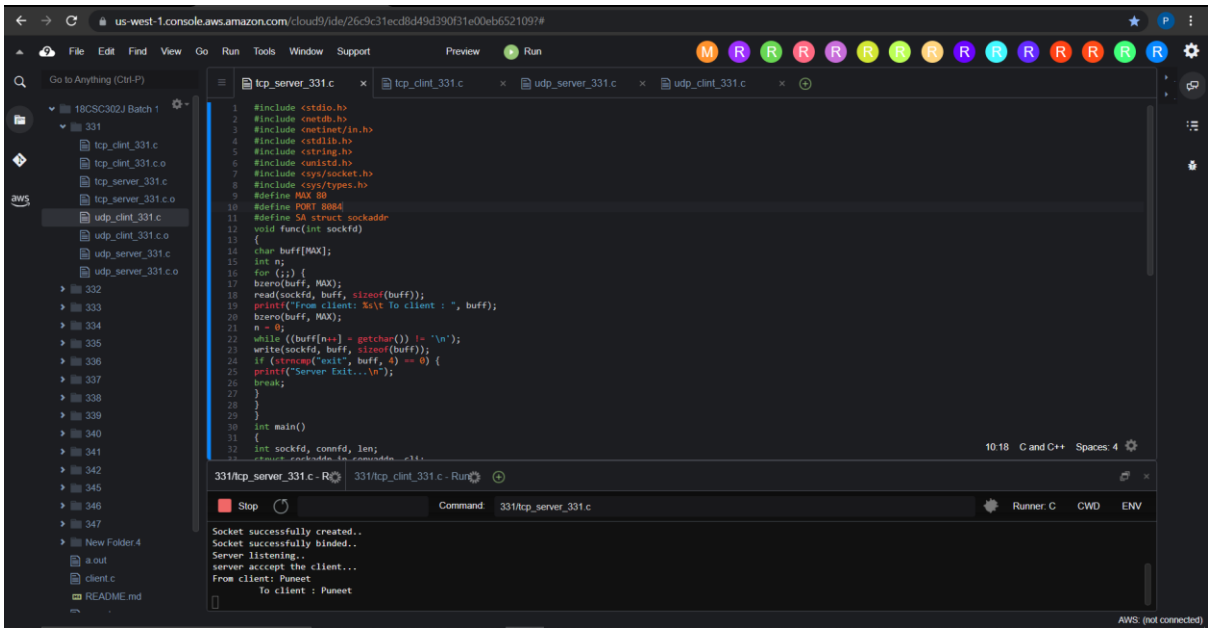
18CSC302J (Computer Networks Lab)
Lab session 3 - Simple TCP /IP Client -Server Communication

Name :- Puneet Sharma
Reg. No. :- RA1911003010331
Class :-CSE F1

TCP SERVER:
SERVER CODE:

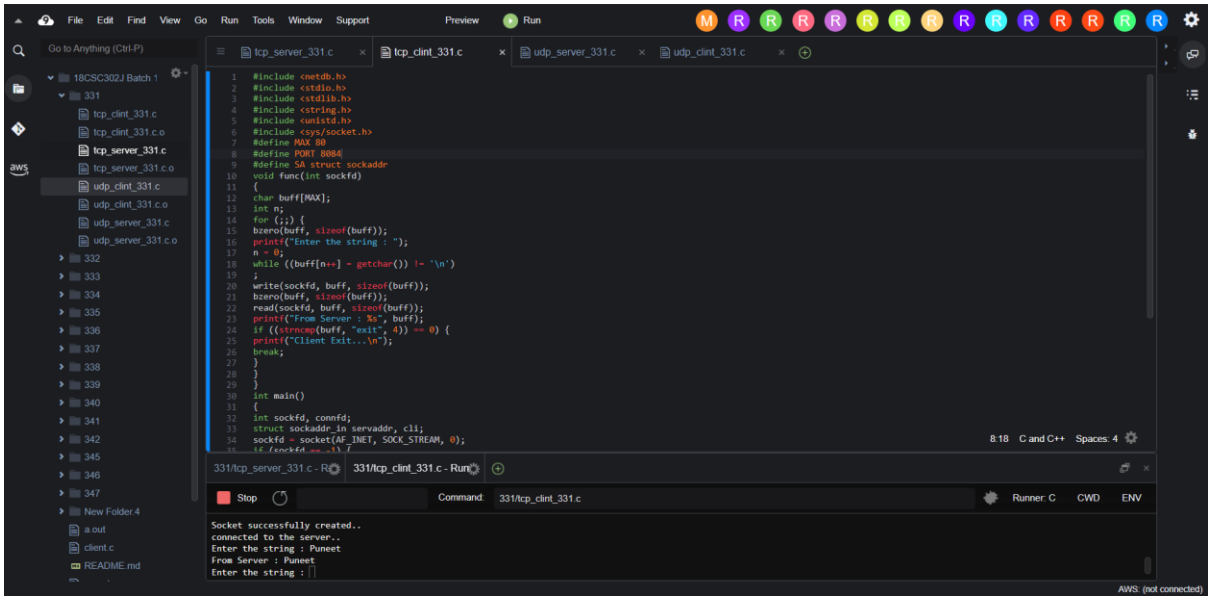
```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#define MAX 80
#define PORT 8084
#define SA struct sockaddr
void func(int sockfd)
{
char buff[MAX];
int n;
for (;;) {
bzero(buff, MAX);
read(sockfd, buff, sizeof(buff));
printf("From client: %s\t To client : ", buff);
bzero(buff, MAX);
n = 0;
while ((buff[n++] = getchar()) != '\n');
write(sockfd, buff, sizeof(buff));
if (strncmp("exit", buff, 4) == 0) {
printf("Server Exit...\n");
break;
}
}
}
int main()
{
int sockfd, connfd, len;
struct sockaddr_in servaddr, cli;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd == -1) {
printf("socket creation failed...\n");
exit(0);
}
else
printf("Socket successfully created..\n");
```

```
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
servaddr.sin_port = htons(PORT);
if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
printf("socket bind failed...\n");
exit(0);
}
else
printf("Socket successfully binded..\n");
if ((listen(sockfd, 5)) != 0) {
printf("Listen failed...\n");
exit(0);
}
else
printf("Server listening..\n");
len = sizeof(cli);
connfd = accept(sockfd, (SA*)&cli, &len);
if (connfd < 0) {
printf("server acccept failed...\n");
exit(0);
}
else
printf("server acccept the client...\n");
func(connfd);
close(sockfd);
}
```



TCP CLIENT:
CLIENT CODE

```
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#define MAX 80
#define PORT 8084
#define SA struct sockaddr
void func(int sockfd)
{
char buff[MAX];
int n;
for (;;) {
bzero(buff, sizeof(buff));
printf("Enter the string : ");
n = 0;
while ((buff[n++] = getchar()) != '\n')
;
write(sockfd, buff, sizeof(buff));
bzero(buff, sizeof(buff));
read(sockfd, buff, sizeof(buff));
printf("From Server : %s", buff);
if ((strncmp(buff, "exit", 4)) == 0) {
printf("Client Exit...\n");
break;
}
}
}
int main()
{
int sockfd, connfd;
struct sockaddr_in servaddr, cli;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd == -1) {
printf("socket creation failed...\n");
exit(0);
}
else
printf("Socket successfully created..\n");
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
servaddr.sin_port = htons(PORT);
if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr)) != 0) {
printf("connection with the server failed...\n");
exit(0);
}
else
printf("connected to the server..\n");
func(sockfd);
close(sockfd);}
```


File Edit Find View Go Run Tools Window Support
Preview
Run
Go to Anything (Ctrl-P)
18CSC302J Batch 1
331
tcp_clint_331.c
tcp_clint_331.c.o
tcp_server_331.c
tcp_server_331.c.o
udp_clint_331.c
udp_clint_331.c.o
udp_server_331.c
udp_server_331.c.o
332
333
334
335
336
337
338
339
340
341
342
345
346
347
New Folder:4
a.out
client.c
README.md
tcp_server_331.c
tcp_clint_331.c
udp_server_331.c
udp_clint_331.c
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#define MAX 80
#define PORT 8084
#define SA struct sockaddr
void func(int sockfd)
{
char buff[MAX];
int n;
for (;;) {
bzero(buff, sizeof(buff));
printf("Enter the string : ");
n = 0;
while ((buff[n++] = getchar()) != '\n')
;
write(sockfd, buff, sizeof(buff));
bzero(buff, sizeof(buff));
read(sockfd, buff, sizeof(buff));
printf("From Server : %s", buff);
if ((strncmp(buff, "exit", 4)) == 0) {
printf("Client Exit...\n");
break;
}
}
}
int main()
{
int sockfd, connfd;
struct sockaddr_in servaddr, cli;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
8:18 C and C++ Spaces: 4
331/tcp_server_331.c - R
331/tcp_clint_331.c - Run
Stop
Command: 331/tcp_clint_331.c
Runner: C CWD ENV
Socket successfully created..
connected to the server..
Enter the string : Puneet
From Server : Puneet
Enter the string :
AWS: (not connected)