Computer Networks Lab
Week-9

Sep 27, 2021

Venkata Naga Sai Ram Nomula
RA1911033010021
L2 - SWE

REMOTE COMMAND EXECUTION USING UDP

GIVEN REQUIREMENTS:

There are two hosts, Client and Server. The Client sends a command to the Server, which executes the command and sends the result back to the Client.

TECHNICAL OBJECTIVE:

Remote Command execution is implemented through this program using which Client is able to execute commands at the Server. Here, the Client sends the command to the Server for remote execution. The Server executes the command and sends the result of the execution back to the Client.

METHODOLOGY:

Server:

- Include the necessary header files.
- Create a socket using the socket function with family AF_INET, type as SOCK_DGRAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin_family to AF_INET, sin_addr to INADDR_ANY, sin_port to dynamically assigned port number.
- Bind the local host using the bind() system call.
- Within an infinite loop, receive the command to be executed from the client.
- Append text "> temp.txt" to the command.
- Execute the command using the "system()" system call.
- Send the result of execution to the Client using a file buffer.

Client:

- Include the necessary header files.
- Create a socket using the socket function with family AF_INET, type as SOCK_DGRAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin_family to AF_INET.
- Get the server IP address and the Port number from the console.
- Using the gethostbyname() function, assign it to a hostent structure, and assign it to sin_addr of the server address structure.
- Obtain the command to be executed on the server from the user.
- Send the command to the server.
- Receive the output from the server and print it on the console.

Code:

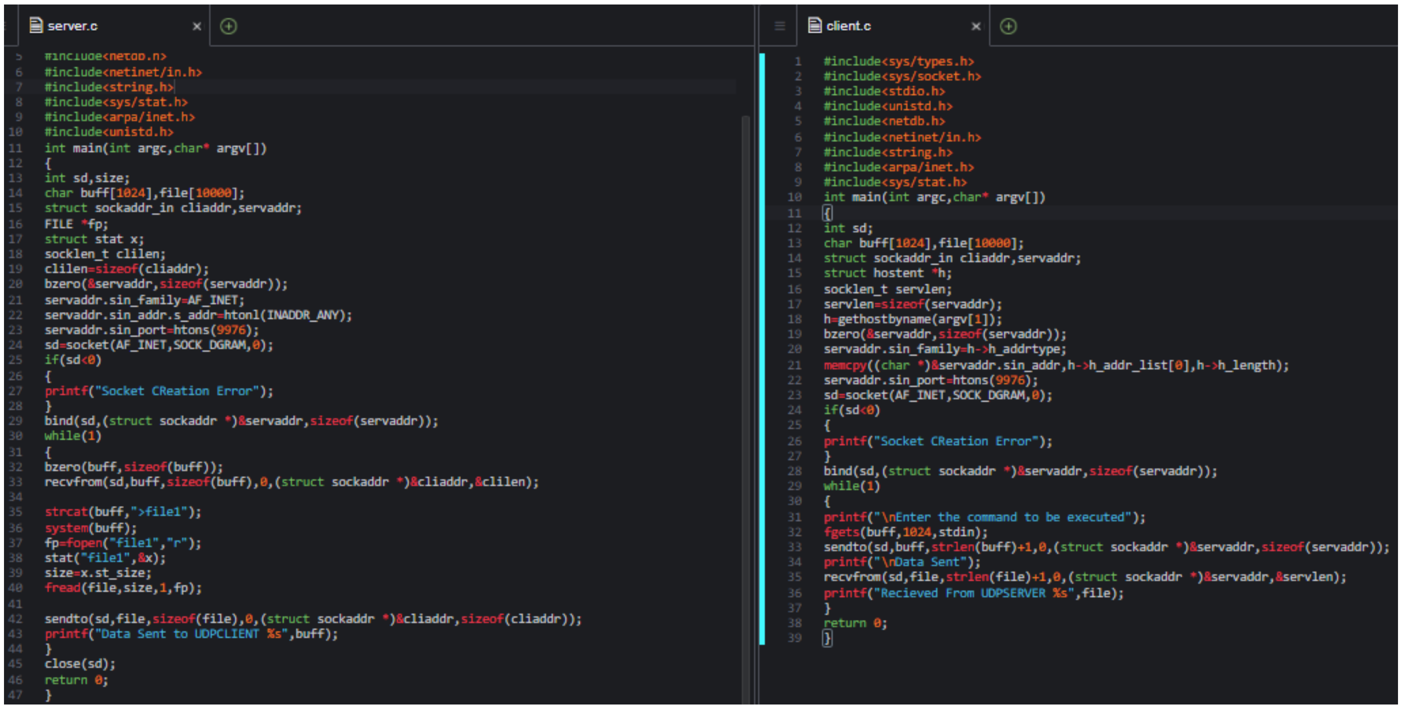
Server.c

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<stdlib.h>
#include<netdb.h>
#include<netinet/in.h>
#include<string.h>
#include<sys/stat.h>
#include<arpa/inet.h>
#include<unistd.h>
int main(int argc,char* argv[])
{
int sd,size;
char buff[1024],file[10000];
struct sockaddr_in cliaddr,servaddr;
FILE *fp;
struct stat x;
socklen_t clilen;
clilen=sizeof(cliaddr);
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(9976);
sd=socket(AF_INET,SOCK_DGRAM,0);
if(sd<0)
{
printf("Socket CReation Error");
}
bind(sd,(struct sockaddr *)&servaddr,sizeof(servaddr));
while(1)
{
bzero(buff,sizeof(buff));
recvfrom(sd,buff,sizeof(buff),0,(struct sockaddr *)&cliaddr,&clilen);

strcat(buff,">file1");
system(buff);
fp=fopen("file1","r");
stat("file1",&x);
size=x.st_size;
fread(file,size,1,fp);

sendto(sd,file,sizeof(file),0,(struct sockaddr *)&cliaddr,sizeof(cliaddr));
printf("Data Sent to UDPCLIENT %s",buff);
}
close(sd);
return 0;
}
```

Client.c

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<netdb.h>
#include<netinet/in.h>
```

```
#include<string.h>
#include<arpa/inet.h>
#include<sys/stat.h>
int main(int argc,char* argv[])
{
int sd;
char buff[1024],file[10000];
struct sockaddr_in cliaddr,servaddr;
struct hostent *h;
socklen_t servlen;
servlen=sizeof(servaddr);
h=gethostbyname(argv[1]);
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=h->h_addrtype;
memcpy((char *)&servaddr.sin_addr,h->h_addr_list[0],h->h_length);
servaddr.sin_port=htons(9976);
sd=socket(AF_INET,SOCK_DGRAM,0);
if(sd<0)
{
printf("Socket CReation Error");
}
bind(sd,(struct sockaddr *)&servaddr,sizeof(servaddr));
while(1)
{
printf("\nEnter the command to be executed");
fgets(buff,1024,stdin);
sendto(sd,buff,strlen(buff)+1,0,(struct sockaddr *)&servaddr,sizeof(servaddr));
printf("\nData Sent");
recvfrom(sd,file,strlen(file)+1,0,(struct sockaddr *)&servaddr,&servlen);
printf("Recieved From UDPSERVER %s",file);
}
return 0;
}
```



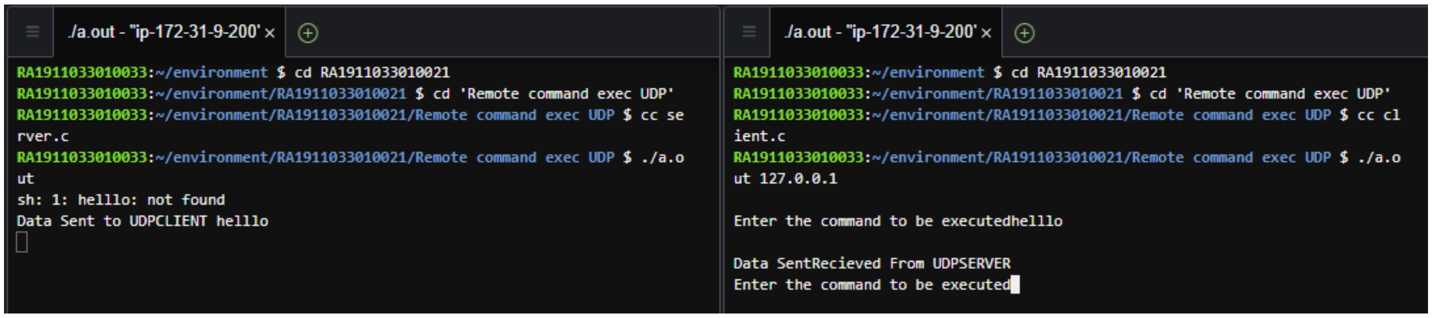
server.c

```c
#include<netdb.h>
#include<netinet/in.h>
#include<string.h>
#include<sys/stat.h>
#include<arpa/inet.h>
#include<unistd.h>
int main(int argc,char* argv[])
{
int sd,size;
char buff[1024],file[10000];
struct sockaddr_in cliaddr,servaddr;
FILE *fp;
struct stat x;
socklen_t clilen;
clilen=sizeof(cliaddr);
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(9976);
sd=socket(AF_INET,SOCK_DGRAM,0);
if(sd<0)
{
printf("Socket CReation Error");
}
bind(sd,(struct sockaddr *)&servaddr,sizeof(servaddr));
while(1)
{
bzero(buff,sizeof(buff));
recvfrom(sd,buff,sizeof(buff),0,(struct sockaddr *)&cliaddr,&clilen);

strcat(buff,">file1");
system(buff);
fp=fopen("file1","r");
stat("file1",&x);
size=x.st_size;
fread(file,size,1,fp);

sendto(sd,file,sizeof(file),0,(struct sockaddr *)&cliaddr,sizeof(cliaddr));
printf("Data Sent to UDPCLIENT %s",buff);
}
close(sd);
return 0;
}
```

client.c

```c
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<netdb.h>
#include<netinet/in.h>
#include<string.h>
#include<arpa/inet.h>
#include<sys/stat.h>
int main(int argc,char* argv[])
{
int sd;
char buff[1024],file[10000];
struct sockaddr_in cliaddr,servaddr;
struct hostent *h;
socklen_t servlen;
servlen=sizeof(servaddr);
h=gethostbyname(argv[1]);
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=h->h_addrtype;
memcpy((char *)&servaddr.sin_addr,h->h_addr_list[0],h->h_length);
servaddr.sin_port=htons(9976);
sd=socket(AF_INET,SOCK_DGRAM,0);
if(sd<0)
{
printf("Socket CReation Error");
}
bind(sd,(struct sockaddr *)&servaddr,sizeof(servaddr));
while(1)
{
printf("\nEnter the command to be executed");
fgets(buff,1024,stdin);
sendto(sd,buff,strlen(buff)+1,0,(struct sockaddr *)&servaddr,sizeof(servaddr));
printf("\nData Sent");
recvfrom(sd,file,strlen(file)+1,0,(struct sockaddr *)&servaddr,&servlen);
printf("Recieved From UDPSERVER %s",file);
}
return 0;
}
```

Result:



./a.out - "ip-172-31-9-200'

```
RA1911033010033:~/environment $ cd RA1911033010021
RA1911033010033:~/environment/RA1911033010021 $ cd 'Remote command exec UDP'
RA1911033010033:~/environment/RA1911033010021/Remote command exec UDP $ cc se
rver.c
RA1911033010033:~/environment/RA1911033010021/Remote command exec UDP $ ./a.o
ut
sh: 1: helllo: not found
Data Sent to UDPCLIENT helllo
```

./a.out - "ip-172-31-9-200'

```
RA1911033010033:~/environment $ cd RA1911033010021
RA1911033010033:~/environment/RA1911033010021 $ cd 'Remote command exec UDP'
RA1911033010033:~/environment/RA1911033010021/Remote command exec UDP $ cc cl
ient.c
RA1911033010033:~/environment/RA1911033010021/Remote command exec UDP $ ./a.o
ut 127.0.0.1

Enter the command to be executedhelllo

Data SentRecieved From UDPSERVER
Enter the command to be executed
```