COMPUTER NETWORKS

EXP 9

REMOTE COMMAND EXECUTION USING UDP

SEPTEMBER 27, 2021

ROEHIT RANGANATHAN
RA1911033010017 | L2

Aim:

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 the send result of the execution back to the Client.

Procedure:

Server:

-> Include the necessary header files.

-> Create a socket using 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 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 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 in 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

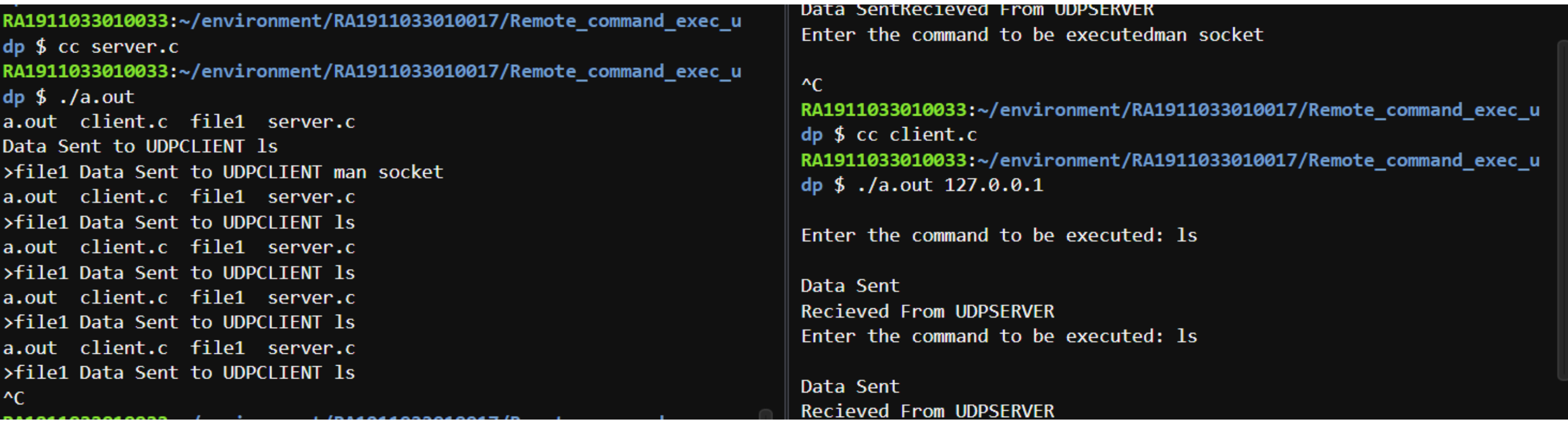
```
#include<sys/types.h>

#include<sys/socket.h>

#include<stdio.h>
```

```
#include<netdb.h>
#include<stdlib.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(3333);
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("\nData 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(3333);
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("\nRecieved From UDPSERVER %s",file);
}
return 0;
}
```

OUTPUT:



```
RA1911033010033:~/environment/RA1911033010017/Remote_command_exec_u
dp $ cc server.c
RA1911033010033:~/environment/RA1911033010017/Remote_command_exec_u
dp $ ./a.out
a.out  client.c  file1  server.c
Data Sent to UDPCLIENT ls
>file1 Data Sent to UDPCLIENT man socket
a.out  client.c  file1  server.c
>file1 Data Sent to UDPCLIENT ls
a.out  client.c  file1  server.c
>file1 Data Sent to UDPCLIENT ls
a.out  client.c  file1  server.c
>file1 Data Sent to UDPCLIENT ls
a.out  client.c  file1  server.c
>file1 Data Sent to UDPCLIENT ls
^C
```

```
Data SentRecieved From UDPSERVER
Enter the command to be executedman socket

^C
RA1911033010033:~/environment/RA1911033010017/Remote_command_exec_u
dp $ cc client.c
RA1911033010033:~/environment/RA1911033010017/Remote_command_exec_u
dp $ ./a.out 127.0.0.1

Enter the command to be executed: ls

Data Sent
Recieved From UDPSERVER
Enter the command to be executed: ls

Data Sent
Recieved From UDPSERVER
```