Madhurya Mozumder

RA1911028010036

CSE-CC

I2

**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 the send result of the execution back to the Client.

**METHODOLOGY:**

**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.

**CODING:**

**Server:**

#include<stdio.h>

#include<arpa/inet.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#define SERV\_TCP\_PORT 5035

#define MAX 60

int i, j, tem;

char buff[4096], t;

FILE \*f1;

int main(int afg, char \*argv)

{

int sockfd, newsockfd, clength;

struct sockaddr\_in serv\_addr,cli\_addr;

char t[MAX], str[MAX];

strcpy(t,"exit");

sockfd=socket(AF\_INET, SOCK\_STREAM,0);

serv\_addr.sin\_family=AF\_INET;

serv\_addr.sin\_addr.s\_addr=INADDR\_ANY;

serv\_addr.sin\_port=htons(SERV\_TCP\_PORT);

printf("\nBinded");

bind(sockfd,(struct sockaddr\*)&serv\_addr, sizeof(serv\_addr));

printf("\nListening...");

listen(sockfd, 5);

clength=sizeof(cli\_addr);

newsockfd=accept(sockfd,(struct sockaddr\*) &cli\_addr,&clength);

close(sockfd);

read(newsockfd, &str, MAX);

printf("\nClient message\n File Name : %s\n", str);

f1=fopen(str, "r");

while(fgets(buff, 4096, f1)!=NULL)

{

write(newsockfd, buff,MAX);

printf("\n");

}

fclose(f1);

printf("\nFile Transferred\n");

return 0;

}

**Client:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#define SERV\_TCP\_PORT 5035

#define MAX 60

int main(int arg,char\*argv[])

{

int sockfd,n;

struct sockaddr\_in serv\_addr;

struct hostent\*server;

char send[MAX],recvline[MAX],s[MAX],name[MAX];

sockfd=socket(AF\_INET,SOCK\_STREAM,0);

serv\_addr.sin\_family=AF\_INET;

serv\_addr.sin\_port=htons(SERV\_TCP\_PORT);

serv\_addr.sin\_addr.s\_addr=INADDR\_ANY;

connect(sockfd,(struct sockaddr\*)&serv\_addr,sizeof(serv\_addr));

printf("\nEnter the source file name : \n");

scanf("%s",send);

write(sockfd,send,MAX);

while((n=read(sockfd,recvline,MAX))!=0)

{

printf("%s",recvline);

}

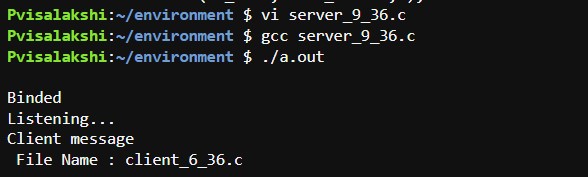
close(sockfd);

return 0;

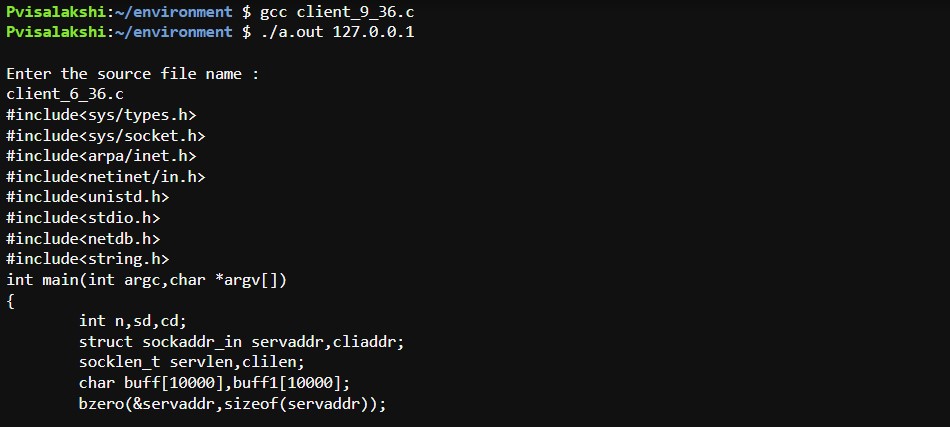
}

**OUTPUT:**

**Server:**

****

**Client:**

****

**INFERENCE:**

Thus the Remote Command Execution between the client and server is implemented.