**A4 :Write a program using TCP socket for wired network for following**

**a. Say Hello to Each other ( For all students) b. File transfer ( For all students)**

**c. Calculator (Arithmetic)**

**Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.**

**A. Say Hello to Each other ( For all students)**

**CODE:**

**SERVER:**

**#include**<stdio.h>

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

**#include**<arpa/inet.h>

**#include**<string.h>

**#include**<stdlib.h>

**#include**<unistd.h>

**#define** DATA "hello from server"

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

{

**int** sock;

**struct** sockaddr\_in server;

**int** mysock;

**char** buffer[1024];

**int** rval;

//creating socket

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

**if**(sock<0)

{

**perror**("faild to creat socket");

**exit**(1);

}

server.sin\_family=AF\_INET;

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

server.sin\_port=**htons**(5050);

//call bind

**if**(**bind**(sock,(**struct** sockaddr\*)&server,**sizeof**(server)))

{

**perror**("bind faild");

**exit**(1);

}

//listen

**listen**(sock,5);

//accept

**do**

{

mysock=**accept**(sock,(**struct** sockaddr\*)0,0);

**if**(mysock==-1)

{

**perror**("accept faild");

}

**else**

{

**memset**(buffer,0,**sizeof**(buffer));

//receive data from client

**if**(**recv**(mysock,buffer,**sizeof**(buffer),0)<0)

{

**perror**("receive faild");

**exit**(1);

}

**printf**("receive message:%s\n",buffer);

**printf**("sending message:%s\n",DATA);

//sendinf data to client

**if**(**send**(mysock,DATA,**sizeof**(DATA),0)<0)

{

**perror**("send faild");

**close**(sock);

**exit**(1);

}

**close**(mysock);

}

}**while**(1);

**return** 0;

}

**output:**

**client:**

**#include<stdio.h>**

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

**#include**<netdb.h>

**#include**<string.h>

**#include**<stdlib.h>

**#include**<unistd.h>

**#define** DATA "hello from client"

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

{

**int** sock;

**struct** sockaddr\_in server;

**struct** hostent \*hp;

**char** buffer[1024];

//creating socket

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

**if**(sock<0)

{

**perror**("faild to creat socket");

**close**(0);

**exit**(1);

}

server.sin\_family=AF\_INET;

//accept command line argument as IP address

hp=**gethostbyname**(argv[1]);

**if**(hp==0)

{

**perror**("gethostbyname faild");

**close**(sock);

**exit**(1);

}

**memcpy**(&server.sin\_addr,hp->h\_addr,hp->h\_length);

server.sin\_port=**htons**(5050);

//connect to server

**if**(**connect**(sock,(**struct** sockaddr\*)&server,**sizeof**(server))<0)

{

**perror**("connect failed");

**close**(sock);

**exit**(1);

}

//send data to server

**if**(**send**(sock,DATA,**sizeof**(DATA),0)<0)

{

**perror**("SEND FAILD");

**close**(sock);

**exit**(1);

}

**printf**("send %s\n",DATA);

//RECEIVE DATA FROM SERVER

**recv**(sock,buffer,**sizeof**(buffer),0);

**printf**("message:%s\n",buffer);

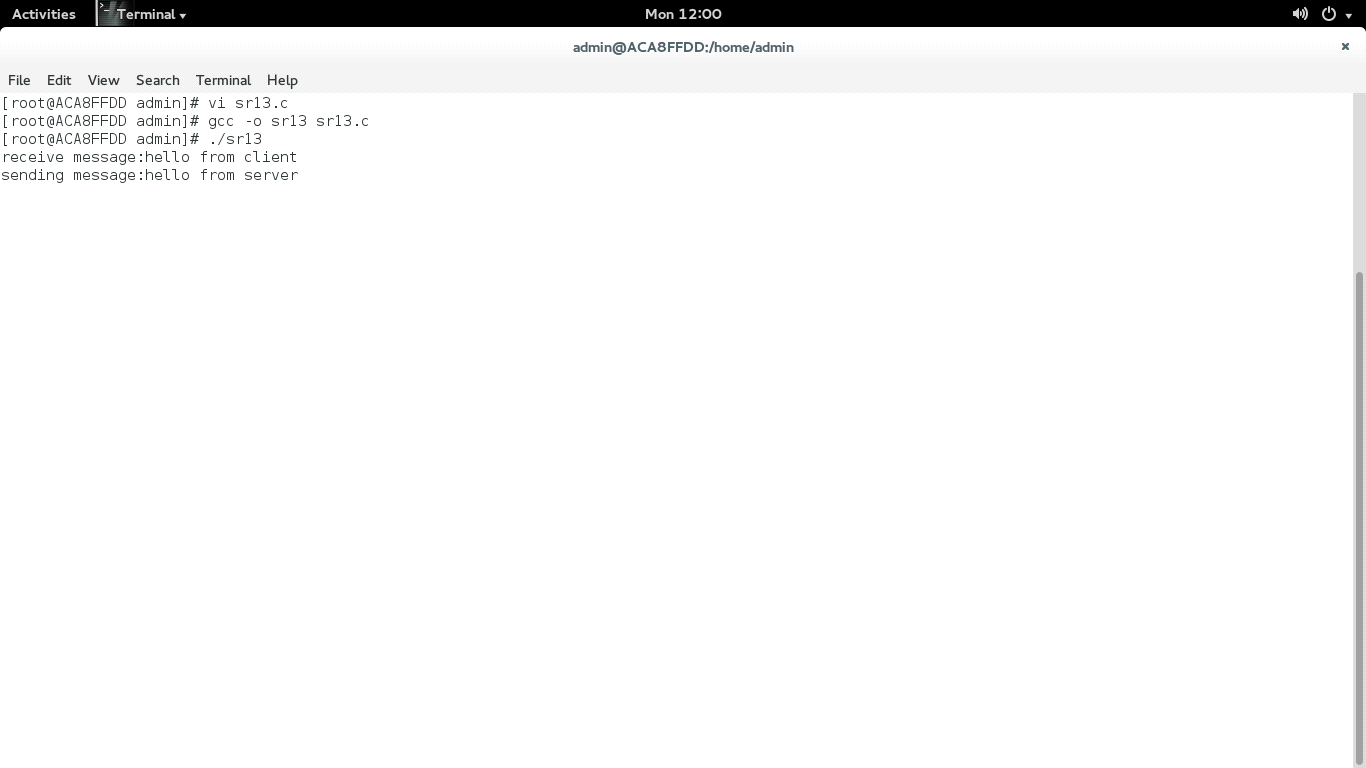
**close**(sock);

**return** 0;

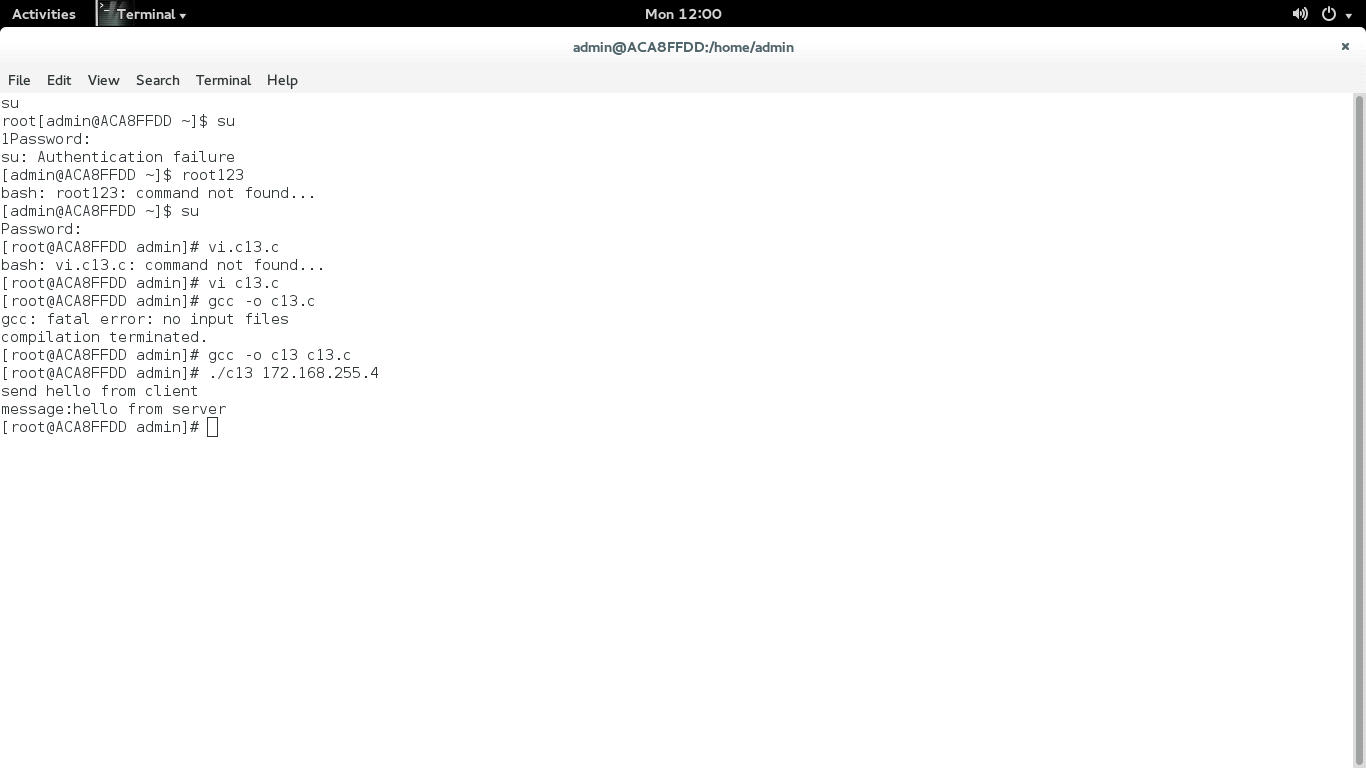
}

**OUTPUT:**

**server**



client:

**B. File transfer ( For all students)**

**CODE:**

**SERVER:**

**#include<stdio.h>**

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

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

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

**#include<string.h>**

**#include<stdlib.h>**

**#include <arpa/inet.h>**

**#include<unistd.h>**

**#include <string.h>**

**//To check if the file exist**

**int exists(const char \*fname)**

**{**

**FILE \*file;**

**if(file = fopen(fname, "r"))**

**{**

**fclose(file);**

**return 1;**

**}**

**return 0;**

**}**

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

**{**

**/\*Variables\*/**

**int sock,csock;**

**struct sockaddr\_in server;**

**struct sockaddr\_in client;**

**int sin\_size;**

**char filename[1024],filedata[1024],buffer[1024];**

**FILE \*fp;**

**/\*Socket\*/**

**if((sock= socket(AF\_INET, SOCK\_STREAM, 0))<0)**

**{**

**perror("Failed to Create Socket");**

**exit(1);**

**}**

**server.sin\_family = AF\_INET;**

**server.sin\_addr.s\_addr = INADDR\_ANY;**

**server.sin\_port = htons(5000);**

**/\*bind\*/**

**if(bind(sock, (struct sockaddr \*)&server, sizeof(server)))**

**{**

**perror("Bind Failed");**

**exit(1);**

**}**

**/\*listen\*/**

**if(listen(sock, 5) == -1)**

**{**

**perror("Listen failed");**

**exit(1);**

**}**

**/\*Accept\*/**

**while(1)**

**{**

**sin\_size = sizeof(client);**

**csock = accept(sock,(struct sockaddr \*)&client, &sin\_size);**

**if(csock == -1)**

**{**

**perror("Accept Failed");**

**}**

**printf("Connetion Received from: %s:%d\n",inet\_ntoa(client.sin\_addr),ntohs(client.sin\_port));**

**memset(filename,0, sizeof(filename));**

**memset(filedata,0, sizeof(filedata));**

**memset(buffer,0, sizeof(buffer));**

**/\*Receive the file name from the client\*/**

**if(recv(csock, filename, sizeof(filename), 0)<0)**

**{**

**perror("Reveive Failed");**

**exit(1);**

**}**

**printf("Filename Check:%s\n",filename);**

**/\*Call exists function to check if file exists\*/**

**if(exists(filename))**

**{**

**//Sending true for file exists as an acknowledgement**

**strcpy(buffer,"True");**

**if(send(csock,buffer,sizeof(buffer),0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**}**

**//Opening the file in read mode**

**fp=fopen(filename, "r");**

**usleep(100000);**

**/\*Reading the file in chunks\*/**

**while (fread(filedata, strlen(filedata)+1, 1, fp) == 1)**

**{**

**/\*Sending the read file data to the client\*/**

**if(send(csock,filedata,sizeof(filedata),0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**}**

**}**

**usleep(100000);**

**if (feof(fp))**

**{**

**/\*Sending the last chunk of file\*/**

**if(send(csock,filedata,sizeof(filedata),0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**usleep(100000);**

**}**

**/\*Sending end as data after the file ends\*/**

**strcpy(filedata,"end");**

**if(send(csock,filedata,sizeof(filedata),0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**usleep(100000);**

**}**

**printf("File written successfully\n");**

**}**

**else**

**{**

**printf("File not read/written successfully\n");**

**}**

**fclose(fp);**

**}**

**else**

**{**

**/\*Sending False if file doesn't exist\*/**

**strcpy(buffer,"False");**

**if(send(csock,buffer,sizeof(buffer),0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**}**

**}**

**/\*Close\*/**

**close(csock);**

**}**

**return 0;**

**}**

**CLIENT:**

**#include<stdio.h>**

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

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

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

**#include<unistd.h>**

**#include<string.h>**

**#include<stdlib.h>**

**#include<netdb.h>**

**#include <string.h>**

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

**{**

**/\*Variables\*/**

**int sock,cnt=0;**

**char buffer[1024],filedata[1024],filename[1024];**

**struct hostent \*host;**

**struct sockaddr\_in server;**

**FILE \*fp;**

**host = gethostbyname(argv[1]);**

**/\*Socket\*/**

**/\*\*\*\*\*\*\*\*\*\*\*Define Socket Here\*\*\*\*\*\*\*\*\*\*/**

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

**if(sock == -1)**

**{**

**perror("Socket Failed");**

**exit(1);**

**}**

**server.sin\_family = AF\_INET;**

**server.sin\_port = htons(5000);**

**memcpy(&server.sin\_addr,host->h\_addr,host->h\_length);**

**/\*Connect\*/**

**if(connect(sock, (struct sockaddr \*) &server, sizeof(server))<0)**

**{**

**perror("Connect Failed");**

**exit(1);**

**}**

**printf("Enter Filename:");**

**scanf("%s",filename);**

**//Send the filename of the file to receive**

**if(send(sock,filename,sizeof(filename), 0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**}**

**//Receive the file status. True if file exists else False**

**if(recv(sock, buffer, sizeof(buffer), 0)<0)**

**{**

**perror("Receive Failed");**

**exit(1);**

**}**

**printf("\nFile status Received: %s\n",buffer);**

**//Checking if received value is True i.e. File exists**

**if (strcmp(buffer,"True") == 0)**

**{**

**printf("File Exists.\n");**

**//strcat(filename,"1");**

**//Creating the file**

**fp=fopen(filename, "w");**

**//Initialize filedata with some random value**

**strcpy(filedata,"random");**

**//Receiving file data in packets till end of file**

**while(strcmp(filedata,"end") != 0)**

**{**

**//Receiving File Data**

**if(recv(sock, filedata, sizeof(filedata), 0)<0)**

**{**

**perror("Receive Failed");**

**exit(1);**

**}**

**//Writing received data into a file**

**/\*\*\*\*\*\*\*\*\*\*\*Define fwrite to write to file Here\*\*\*\*\*\*\*\*\*\*/**

**//fwrite(filedata , 1 , sizeof(filedata) , fp );**

**if(strcmp(filedata,"end") != 0)**

**{**

**fprintf(fp,filedata);**

**}**

**printf("Received:%d\n",cnt);**

**cnt++;**

**}**

**fclose(fp);**

**}**

**else**

**printf("File Doesn't Exist.\n");**

**return 0;**

**}**

**OUTPUT:**

client side:

[admin@localhost ~]$ vi fc.c

[admin@localhost ~]$ gcc -o fc fc.c

[admin@localhost ~]$ ./fc 172.168.252.64

Enter Filename:sk.txt

File status Received: True

File Exists.

Received:0

Received:1

server side:

[admin@localhost ~]$ vi fs.c

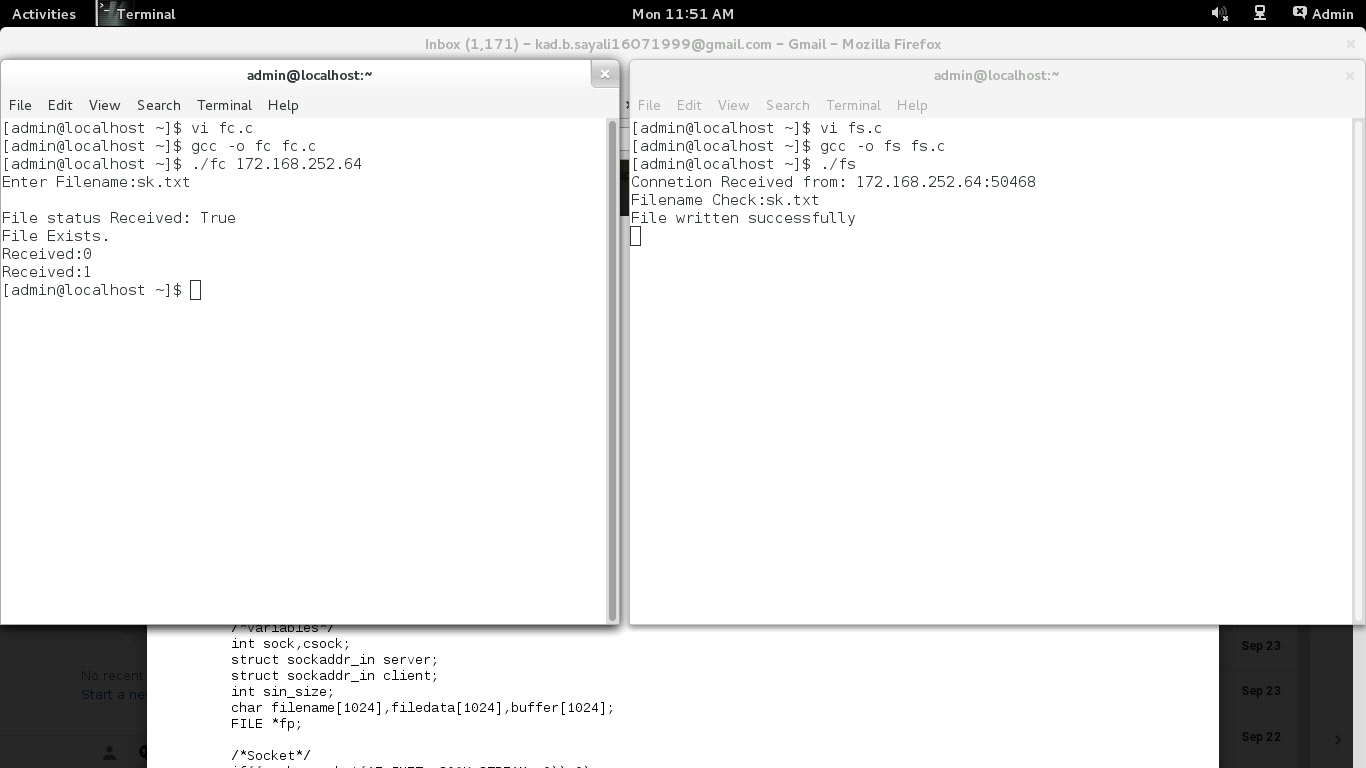
[admin@localhost ~]$ gcc -o fs fs.c

[admin@localhost ~]$ ./fs

Connetion Received from: 172.168.252.64:50468

Filename Check:sk.txt

File written successfully



**C. Calculator (Arithmetic)**

**CODE:**

**server:**

**#include<stdio.h>**

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

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

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

**#include<string.h>**

**#include<stdlib.h>**

**#include <arpa/inet.h>**

**#include<unistd.h>**

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

**{**

**/\*Variables\*/**

**int sock,csock;**

**struct sockaddr\_in server;**

**struct sockaddr\_in client;**

**int sin\_size,num1,num2,result;**

**char buffer[1024],buffer1[1024];**

**/\*Socket\*/**

**if((sock = socket(AF\_INET,SOCK\_STREAM,0))<0)**

**{**

**perror("Failed to Create Socket");**

**exit(1);**

**}**

**server.sin\_family = AF\_INET;**

**server.sin\_addr.s\_addr = INADDR\_ANY;**

**server.sin\_port = htons(5000);**

**/\*bind\*/**

**if(bind(sock, (struct sockaddr \*)&server, sizeof(server)))**

**{**

**perror("Bind Failed");**

**exit(1);**

**}**

**/\*listen\*/**

**if(listen(sock,5) == -1)**

**{**

**perror("Listen failed");**

**exit(1);**

**}**

**/\*Accept\*/**

**while(1)**

**{**

**sin\_size = sizeof(client);**

**csock = accept(sock,(struct sockaddr \*)&client, &sin\_size);**

**if(csock == -1)**

**{**

**perror("Accept Failed");**

**}**

**//printf("\n I got a connection");**

**printf("Connetion Received from: %s:%d\n",inet\_ntoa(client.sin\_addr),ntohs(client.sin\_port));**

**//Receive Number 1**

**if(recv(csock,&num1,sizeof(int),0)<0)**

**{**

**perror("Receive 1 Fialed");**

**exit(1);**

**}**

**printf("Num1:%d\n",num1);**

**//Receive Number 2**

**if(recv(csock,&num2,sizeof(int),0)<0)**

**{**

**perror("Receive 2 Fialed");**

**exit(1);**

**}**

**printf("Num2:%d\n",num2);**

**//Perform Addition and store in result**

**result = num1+num2;**

**printf("Result:%d\n",result);**

**//Send result to the server**

**if(send(csock,&result,sizeof(int),0)<0)**

**{**

**perror("Send Failed");**

**exit(1);**

**}**

**/\*Close\*/**

**close(csock);**

**}**

**return 0;**

**}**

**client:**

**#include<stdio.h>**

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

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

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

**#include<unistd.h>**

**#include<string.h>**

**#include<stdlib.h>**

**#include<netdb.h>**

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

**{**

**/\*Variables\*/**

**int sock,num1,num2,result;**

**char buffer[1024],buffer1[1024];**

**struct hostent \*host;**

**struct sockaddr\_in server;**

**host = gethostbyname(argv[1]);**

**/\*Socket\*/**

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

**if(sock == -1)**

**{**

**perror("Socket Failed");**

**exit(1);**

**}**

**server.sin\_family = AF\_INET;**

**server.sin\_port = htons(5000);**

**memcpy(&server.sin\_addr,host->h\_addr,host->h\_length);**

**/\*Connect\*/**

**if(connect(sock,(struct sockaddr \*)&server,sizeof(server))<0)**

**{**

**perror("Connect Failed");**

**exit(1);**

**}**

**//Number 1 input**

**printf("Enter num1:");**

**scanf("%d",&num1);**

**//Number 2 input**

**printf("Enter num2:");**

**scanf("%d",&num2);**

**/\*Send\*/**

**//Sending Number 1**

**if(send(sock,&num1,sizeof(int),0)<0)**

**{**

**perror("send failed");**

**exit(1);**

**}**

**//Sending Number 2**

**if(send(sock,&num2,sizeof(int),0)<0)**

**{**

**perror("Send2 Failed");**

**exit(1);**

**}**

**//Receiving the addition resule**

**recv(sock,&result,sizeof(int),0);**

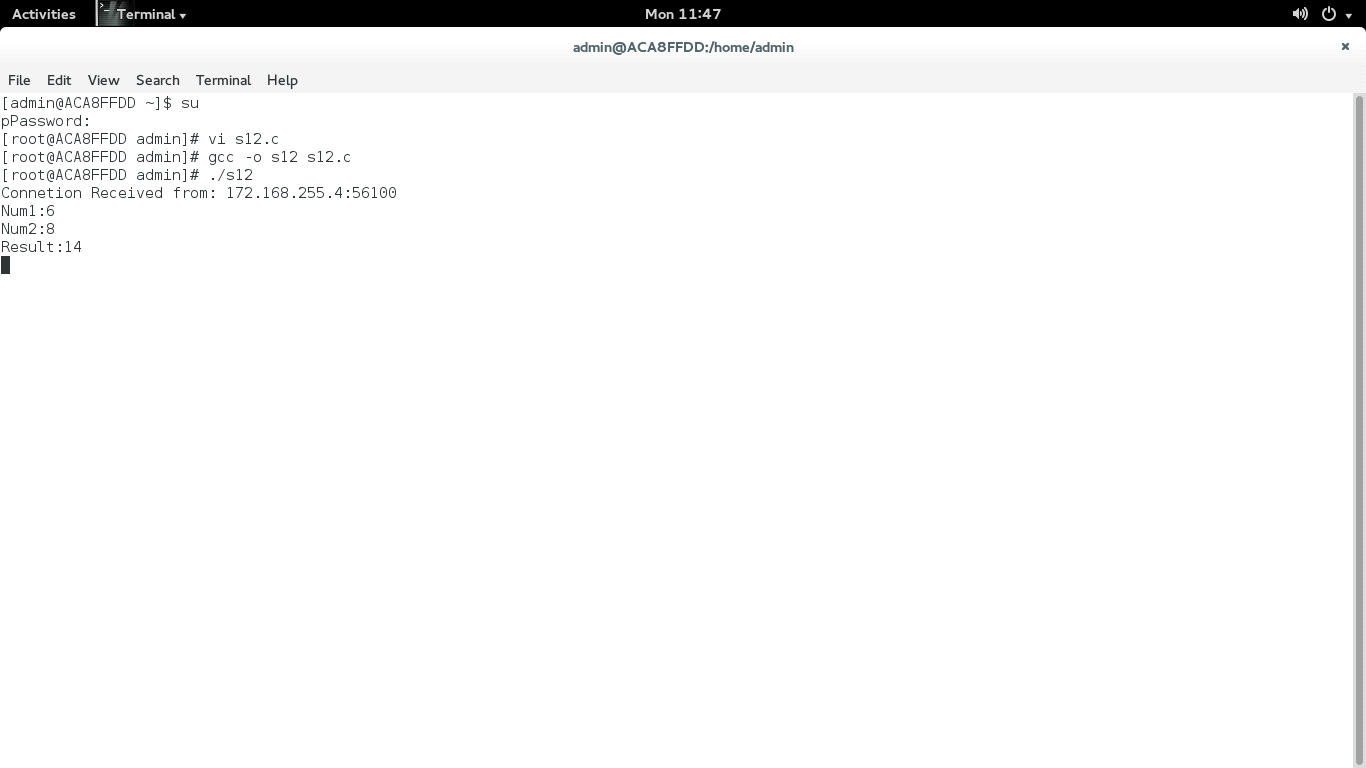
**printf("Answer is : %d\n",result);**

**return 0;**

**}**

**output:**

**server:**



client:

