**RPC program to display message**

**Vi msg.x**

program msg\_prog{

version msg\_ver{

string message()=1;

}=1;

}=0x20000001;

Next compile the program by using command

$ rpcgen –a msg.x

**Vi msg\_client.c**

/\*

\* This is sample code generated by rpcgen.

\* These are only templates and you can use them

\* as a guideline for developing your own functions.

\*/

#include "msg.h"

void

msg\_prog\_1(char \*host)

{

CLIENT \*clnt;

char \* \*result\_1;

char \*message\_1\_arg;

#ifndef DEBUG

clnt = clnt\_create (host, msg\_prog, msg\_ver, "udp");

if (clnt == NULL) {

clnt\_pcreateerror (host);

exit (1);

}

#endif /\* DEBUG \*/

\*result\_1=(char\*) malloc (sizeof(char) \*25);

result\_1 = message\_1((void\*)&message\_1\_arg, clnt);

if (result\_1 == (char \*\*) NULL) {

clnt\_perror (clnt, "call failed");

}

#ifndef DEBUG

clnt\_destroy (clnt);

#endif /\* DEBUG \*/ }

int

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

{

char \*host;

if (argc < 2) {

printf ("usage: %s server\_host\n", argv[0]);

exit (1); }

host = argv[1];

msg\_prog\_1 (host);

exit (0); }

**Vi msg\_server.c**

/\*

\* This is sample code generated by rpcgen.

\* These are only templates and you can use them

\* as a guideline for developing your own functions.

\*/

#include "msg.h"

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

char \*\*

message\_1\_svc(void \*argp, struct svc\_req \*rqstp)

{

static char \* result;

/\*

\* insert server code here

\*/

result=(char \*) malloc(sizeof(char)\*25);

strcpy(result,"hello-this is a response from rpc server");

return &result;

}

Next compile the code using the command

$cc –o msg\_client.c msg\_clnt.c –lnsl msgclient

$cc –o msg\_server.c msg\_svc.c –lnsl msgserver

After successfully compilation execute the program by using command

$./msgserver &

$./msgclient 10.2.0.3

**RPC program to add two numbers**

**Vi add.x**

struct num{

int a;

int b;

};

program add\_prog{

version add\_ver{

int addition(num)=1;

}=1;

}=0x20000002;

Next compile the program

**$ rpcgen –a add.x**

**Vi add\_client.c**

#include "add.h"

void

add\_prog\_1(char \*host)

{

CLIENT \*clnt;

int \*result\_1;

num addition\_1\_arg;

#ifndef DEBUG

clnt = clnt\_create (host, add\_prog, add\_ver, "udp");

if (clnt == NULL) {

clnt\_pcreateerror (host);

exit (1); }

#endif /\* DEBUG \*/

printf("\n enter the two number to add...\n"); // reading 2 numbers for addition

scanf("%d%d",&addition\_1\_arg.a,&addition\_1\_arg.b); // assigned readed number

result\_1=(int \*) malloc(sizeof(int)); // allocate memroy

result\_1 = addition\_1(&addition\_1\_arg, clnt);

if (result\_1 == (int \*) NULL) {

clnt\_perror (clnt, "call failed"); }

#ifndef DEBUG

printf("\n the of %d\t%d is ..... %d\n",addition\_1\_arg.a,addition\_1\_arg.b,\*result\_1);

clnt\_destroy (clnt);

#endif /\* DEBUG \*/ }

int

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

{

char \*host;

if (argc < 2) {

printf ("usage: %s server\_host\n", argv[0]);

exit (1); }

host = argv[1];

add\_prog\_1 (host);

exit (0);

}

**Vi add\_server.c**

/\*

\* This is sample code generated by rpcgen.

\* These are only templates and you can use them

\* as a guideline for developing your own functions.

\*/

#include "add.h"

int \*

addition\_1\_svc(num \*argp, struct svc\_req \*rqstp)

{

static int result;

/\*

\* insert server code here

\*/

result = argp->a + argp->b; // adding numbers

return &result;

}

Next compile the code using the command

$cc –o add\_client.c add\_clnt.c –lnsl addclient

$cc –o add\_server.c add\_svc.c –lnsl addserver

After successfully compilation execut the program by using command

$./addserver &

$./addclient 10.2.0.3