**GCD of two numbers**

Vi gcd.x

struct num

{

long a;

long b;

};

program gcd\_prog{

version gcd\_vers{

long gcd\_fn(num)=1;

}=1;

}=0x30000001;

**[khaja@mars distsyslab]$ rpcgen gcd.x**

**[khaja@mars distsyslab]$ ls**

**client Echos.h Echos.x gcd\_svc.c Makefile.Echos**

**Echos\_client.c Echos\_server.c gcd\_clnt.c gcd.x server**

**Echos\_clnt.c Echos\_svc.c gcd.h gcd\_xdr.c**

**GCD CLIENT CODE**

**Vi gcd\_client.c**

**#include "gcd.h"**

**void**

**gcd\_prog\_1(char \*host,num number)**

**{**

**CLIENT \*clnt;**

**long \*result\_1;**

**num gcd\_fn\_1\_arg;**

**gcd\_fn\_1\_arg.a=number.a;**

**gcd\_fn\_1\_arg.b=number.b;**

**#ifndef DEBUG**

**clnt = clnt\_create (host, gcd\_prog, gcd\_vers, "udp");**

**if (clnt == NULL) {**

**clnt\_pcreateerror (host);**

**exit (1);**

**}**

**#endif /\* DEBUG \*/**

**result\_1 = gcd\_fn\_1(&gcd\_fn\_1\_arg, clnt);**

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

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

**}**

**printf("gcd is %d",\*result\_1);**

**#ifndef DEBUG**

**clnt\_destroy (clnt);**

**#endif /\* DEBUG \*/**

**}**

**int**

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

**{**

**char \*host;**

**num n;**

**if (argc < 2) {**

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

**exit (1);**

**}**

**host = argv[1];**

**n.a=atol(argv[2]);**

**n.b=atol(argv[3]);**

**gcd\_prog\_1 (host,n);**

**exit (0);**

**}**

**Server code**

**Vi gcd\_server.c**

int gcd(int a ,int b){

if (b==0)

return a;

return gcd(b,a%b);}

#include "gcd.h"

long \*

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

{

static long result;

/\*

\* insert server code here

\*/

result=gcd((\*argp).a,(\*argp).b);

return &result;

}

Execution :-

$cc -o gcd\_server gcd\_server.c gcd\_svc.c gcd\_xdr.c –lnsl

$cc -o gcd\_client gcd\_client.c gcd\_clnt.c gcd\_xdr.c –lnsl

$./gcd\_server &

[4] 5086

$ ./gcd\_client 10.2.0.5 10 12

gcd is 2[khaja@mars gcd]$

**Simple message program**

**Vi msg.x**

program msg\_prg{

version msg\_ver{

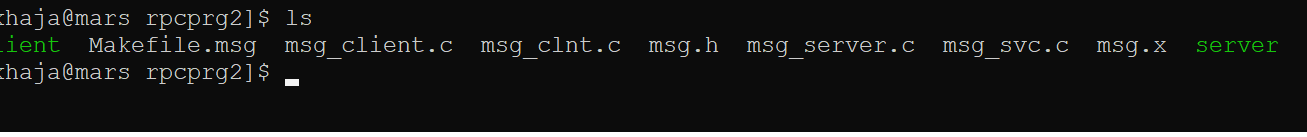
string message() = 1;

} = 1;

} = 0x20000001;

Step -2

Compile the program :- rpcgen –a msg.x



Open vi msg\_client.c

#include "msg.h"

void

msg\_prg\_1(char \*host)

{

CLIENT \*clnt;

char \* \*result\_1;

char \*message\_1\_arg;

#ifndef DEBUG

clnt = clnt\_create (host, msg\_prg, 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

printf("\n Server Response is %s \n",\*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];

msg\_prg\_1 (host);

exit (0);

}

**Vi msg\_server.c**

#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 both programs :-

cc msg\_client.c msg\_clnt.c -o client

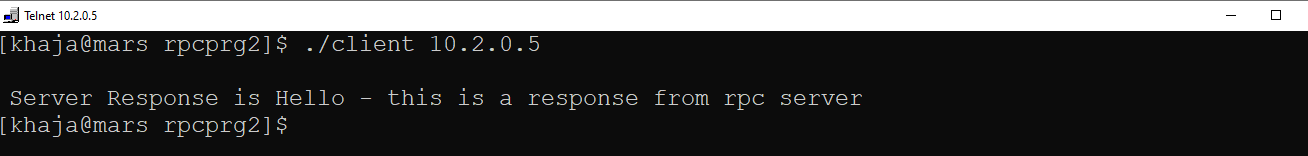
cc msg\_server.c msg\_svc.c -o server

Next run the server

./server &



output



**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;

**Vi add\_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 "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");**

**scanf("%d %d",&addition\_1\_arg.a,&addition\_1\_arg.b);**

**result\_1=(int \*) malloc(sizeof(int));**

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

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

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

}

#ifndef DEBUG

**printf("\n the sum 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;**

return &result;

}