# KADI SARVA VISHWAVIDYALAYA

## LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH

## GANDHINAGAR



## Department of

## Computer Engineering and Information Technology

## Subject: Distributed Systems (CT704A-N)

## Laboratory Manual

Prepared By

**Jain Shreyas Sunilkumar**
**19BECE30183**
**7 CE-E div**

# LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH

## GANDHINAGAR

DEPARTMENT OF COMPUTER ENGINEERING
&
INFORMATION TECHNOLOGY

# CERTIFICATE

Mr./~~Miss~~ _____

of Enrolment No_____

Exam No,_____has satisfactorily completed his/~~her~~ term work in **Distributed Systems (CT704A-N)** for the term ending in **Nov-2022**.

Date: _____

Prof. RASHMIKA PATEL
**Subject Coordinator**

Dr. Sandeep Modha
**HOD-CE**

# INDEX

**Practical: 1**
**Write a program to implement “Hello World!” using RMI**

**Rmihellointer.java**

```
import java.rmi.Remote;
import java.rmi.RemoteException;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student1
 */
public interface rmihellointer extends Remote{
   public String say1() throws RemoteException;
}
```

**Rmihelloserver.java**

```
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student1
 */
public class rmihelloserver extends UnicastRemoteObject implements rmihellointer{

   private String messg;

   public rmihelloserver(String msgg) throws RemoteException
   {
      messg=msgg;
   }
      @Override
   public String say1() throws RemoteException {
      return messg;
   }

   public static void main(String[] args) throws RemoteException{
      try
      {
         Registry re1=LocateRegistry.createRegistry(9999);
         re1.rebind("hi everyone",new rmihelloserver("hello world"));
         System.out.println("Server is ready");
      }
      catch(RemoteException e)
      {
         System.out.println(e);
```

```
        }
    }
}
```

**Rmihelloclient.java**

```
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student1
 */
public class rmihelloclient {
    public static void main(String[] args) throws RemoteException{
        rmihelloclient rm1=new rmihelloclient();
        rm1.remoteconnection();
    }

    private void remoteconnection() throws RemoteException
    {
        try
        {
            Registry  re11=LocateRegistry.getRegistry("Localhost",9999);
            rmihellointer rhi=(rmihellointer) re11.lookup("hi everyone");
            System.out.println(rhi.say1());
        }
        catch(NotBoundException e)
        {
            System.out.println(e);
        }
    }
}
```

# OUTPUT

To compile the Java source files, run the javac command as follows:

javac -d $HOME/myclasses
Hello.java HelloImpl.java HelloClient.java

**Practical:2**

**Write a program to implement Calculator using RMI**

**Intercalc.java**

```
import java.rmi.Remote;
import java.rmi.RemoteException;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student1
 */
public interface intercalc extends Remote{
   public long addi(long x,long y) throws RemoteException;
   public long subt(long x,long y) throws RemoteException;
   public long mult(long x,long y) throws RemoteException;
   public long divi(long x,long y) throws RemoteException;
}
```

**Servercalc.java**

```
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student1
 */
public class servercalc extends UnicastRemoteObject implements intercalc{

   public servercalc() throws RemoteException
   {
      super();
   }
   @Override
   public long addi(long x, long y) throws RemoteException {
      System.out.println("addition of" + x +"and "+ y);
      return x+y;
   }

   @Override
   public long subt(long x, long y) throws RemoteException {
      System.out.println("Subtraction of" + x +"and "+ y);
      return x-y;
   }

   @Override
   public long mult(long x, long y) throws RemoteException {
      System.out.println("Multiplication of" + x +"and "+ y);
      return x*y;
   }

   @Override
```

```
    public long divi(long x, long y) throws RemoteException {
        System.out.println("Division of" + x +"and "+ y);
        return x/y;
    }

    public static void main(String[] args) throws RemoteException{
        try
        {
            Registry regis=LocateRegistry.createRegistry(9999);
            regis.rebind("service calc",new servercalc());
            System.out.println("Server is ready");
        }
        catch(RemoteException e)
        {
            System.out.println(e);
        }
    }
}
```

**Clientcalc.java**

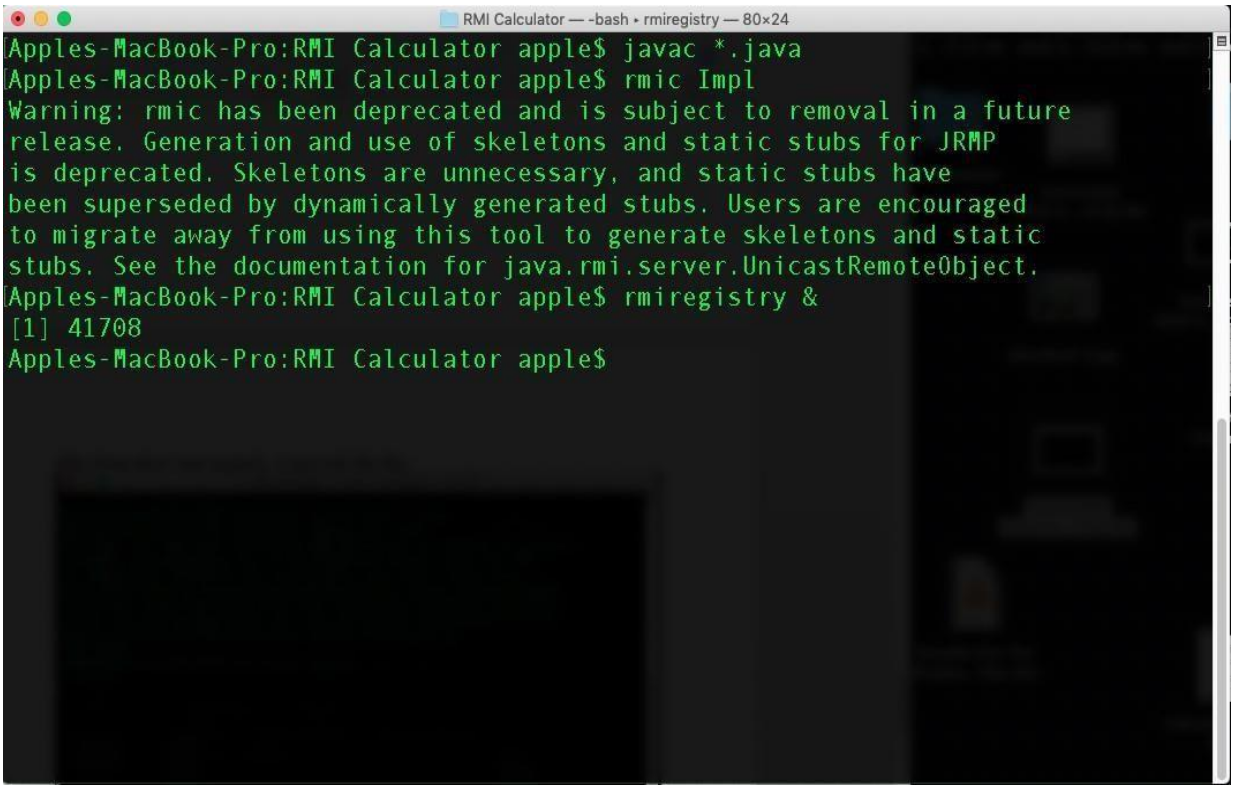
```
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student1
 */
public class clientcalc  {
   public static void main(String[] args) throws RemoteException{
      clientcalc c15=new clientcalc();
      c15.remoteme();
   }

   private void remoteme() throws RemoteException
   {
      try
      {
    Registry regst=LocateRegistry.getRegistry("localhost",9999);
    intercalc itc=(intercalc) regst.lookup("service calc");
      System.out.println("addition is "+itc.addi(3,4) );
      System.out.println("Subtraction is "+itc.subt(3,4) );
      System.out.println("Multiplicatio is "+itc.mult(3,4) );
      System.out.println("Division is "+itc.divi(3,4) );
      }
      catch(NotBoundException e)
      {
         System.out.println(e);
      }
   }
}
```

## OUTPUT


RMI Calculator — -bash ▸ rmiregistry — 80×24
Apples-MacBook-Pro:RMI Calculator apple$ javac *.java
Apples-MacBook-Pro:RMI Calculator apple$ rmic Impl
Warning: rmic has been deprecated and is subject to removal in a future
release. Generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are encouraged
to migrate away from using this tool to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
Apples-MacBook-Pro:RMI Calculator apple$ rmiregistry &
[1] 41708
Apples-MacBook-Pro:RMI Calculator apple$

**Practical -3**

## Write a program to calculate interest rate using RMI.

**interrmi3.java**

```
import java.rmi.Remote;

import java.rmi.RemoteException;


/*

 * To change this license header, choose License Headers in Project Properties.

 * To change this template file, choose Tools | Templates

 * and open the template in the editor.

 */


/**

 *

 * @author Bhavesh Patel

 */

public interface interrmi3 extends Remote {

    public Double getInterestAmount(double interestRate,int numberOfYears,double loanAmount) throws RemoteException;

   public Double getTotalAmount(double loanAmount,double interestAmount) throws RemoteException;

}
```

**rmiserver3.java**

```
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Bhavesh Patel
 */
public class rmiserver3 extends UnicastRemoteObject implements interrmi3{

    public rmiserver3() throws RemoteException
    {
        super();
    }

    @Override
    public Double getInterestAmount(double interestRate, int numberOfYears, double loanAmount)
throws RemoteException {
        double interestAmount = (interestRate * numberOfYears * loanAmount)/100;
```

```
        return(interestAmount);
    }

    @Override
    public Double getTotalAmount(double loanAmount, double interestAmount) throws
RemoteException {
        double totalAmount = loanAmount + interestAmount;
        return(totalAmount);
    }

    public static void main(String[] args) throws RemoteException{

        try
        {
            Registry reg=LocateRegistry.createRegistry(9999);
            reg.rebind("interestrate service",new rmiserver3() );
            System.out.println("Server is Ready");
        }
        catch(RemoteException e)
        {
            System.out.println(e);
        }
    }
}
```

**clientrmi3.java**

```
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Bhavesh Patel
 */
public class clientrmi3 {
    public static void main(String[] args) throws RemoteException{
        clientrmi3 c1=new clientrmi3();
        c1.rmconn();
    }
    private void rmconn() throws RemoteException
    {
        try
        {
            Registry reg1=LocateRegistry.getRegistry("Localhost",9999);
```

```
        interrmi3 ir1=(interrmi3) reg1.lookup("interestrate service");
        double interestAmount = ir1.getInterestAmount(0.5,2,10000);
        System.out.println("Interest rate is"+ interestAmount );
        double totalAmount=ir1.getTotalAmount(10000,interestAmount);
        System.out.println("Totla Amount is" + totalAmount);

      }
      catch(NotBoundException e)
      {
        System.out.println(e);
      }
    }
}
```

## OUTPUT

```
C:\myrmi\javac Mortgage.java
C:\myrmi\java Mortgage
Usage: java Mortgage principal annualInterest years

For example: java Mortgage 80000 .065 15

You will get the output like the following:

The principal is $80000
The annual interest rate is 6.5%
The term is 15 years
Your monthly payment is $696.89

C:\myrmi>java Mortgage 150000 .060 15
The principal is $150000
The annual interest rate is 6.0%
The term is 15 years
Your monthly payment is $1265.79

C:\myrmi>
```

**Practical-4**

**AIM:** Write a program to implement Time Service using RMI.

## Fourinter.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Bhavesh Patel
 */
public interface fourinter extends Remote{
   public String timeget() throws RemoteException;
}
```

**Fourserver.java**

```
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
import java.util.Calendar;
import java.util.GregorianCalendar;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Bhavesh Patel
 */
public class fourserver extends UnicastRemoteObject implements fourinter{

    public fourserver() throws RemoteException
    {
        super();
    }

    @Override
    public String timeget() throws RemoteException {
```

```
        //return new java.util.Date().toString();
            }

    public static void main(String[] args) throws RemoteException{

        try
        {
            Registry dg=LocateRegistry.createRegistry(9999);
            dg.rebind("timeservice",new fourserver() );
            System.out.println("Server is Ready");
        }
        catch(RemoteException e)
        {
            System.out.println(e);
        }
    }

}
```

**Fourclient.java**

```
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Bhavesh Patel
 */
public class fourclient {
    public static void main(String[] args) throws RemoteException{
        fourclient fc=new fourclient();
        fc.connremo();
    }

    private void connremo() throws RemoteException
    {
        try
        {
            Registry reg=LocateRegistry.getRegistry("Localhost",9999);
```

```
            fourinter sa=(fourinter) reg.lookup("timeservice");
            System.out.println("the time get form server is " + sa.timeget());
        }
        catch(NotBoundException e)
        {
            System.out.println(e);
        }
    }
}
```

**Practical-5**

**Aim:** Write a program to implement “Hello World!” using RPC

**Hello.x**

```
program DISPLAY_PRG

{
   version DISPLAY_VER

    {

                int print_hello( void ) = 1;

    } = 1;

} = 0x20000001;
```

**Hello_Client.c**

```
#include <stdio.h>

#include "hello.h"

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

{

      CLIENT *client;

      int   *return_value, filler;

      char  *server;

      if (argc != 2)
      {

            fprintf(stderr, "Usage: %s host_name\n",
*argv);

            exit(1);

      }



      server = argv[1];

      if
((client=clnt_create(server,DISPLAY_PRG,DISPLAY_VER,
"udp")) == (CLIENT *) NULL)
      {

            clnt_pcreateerror(server);

            exit(2);

      }
```

```
	printf("client : Calling function.\n");

	return_value = print_hello_1((void *) &filler,
client);

	if (*return_value)

		printf("client : Mission
accomplished.\n");

	else

		printf("client : Unable to display
message.\n");

	return 0;

}
```

**Hello_Server.c**

```
#include <stdio.h>

#include "hello.h"

int *print_hello_1_svc(void * filler, struct svc_req
* req)
{

      static int  ok;

      ok = printf("server : Hello, welcome to
LDRP.\n");

      return (&ok);

}
```

## OUTPUT

```
client: Calling function.
client: Mission accomplished.
server: Hello, welcome to LDRP.
```

**AIM:** Write a program to implement Arithmetic operations using RPC.

**calculate.x**

```
struct inputs{
 float num1;
 float num2;
 char operator;
};

program CALCULATE_PROG{
 version CALCULATE_VER{
 float ADD(inputs)=1;
 float SUB(inputs)=2;
 float MUL(inputs)=3;
 float DIV(inputs)=4;

 }=1;
}=0x2fffffff;
```

**calculate_client.c**

```
#include <stdio.h>
#include "calculate.h"


float calculate_prog_1(char *host,float n1,float
n2,char opr,CLIENT *clnt)
{

 float *result_1;
 inputs add_1_arg;
 float *result_2;
 inputs sub_1_arg;
 float *result_3;
 inputs mul_1_arg;
 float *result_4;
 inputs div_1_arg;

if(opr=='+'){

 add_1_arg.num1=n1;
 add_1_arg.num2=n2;
 add_1_arg.operator=opr;

 result_1 = add_1(&add_1_arg, clnt);
 if (result_1 == (float *) NULL) {
 clnt_perror (clnt, "call failed");
 }
 return *result_1;
}

else if(opr=='-'){

 sub_1_arg.num1=n1;
 sub_1_arg.num2=n2;
 sub_1_arg.operator=opr;

 result_2 = sub_1(&sub_1_arg, clnt);
 if (result_2 == (float *) NULL) {
 clnt_perror (clnt, "call failed");
 }
```

```
 return *result_2;
}

else if(opr=='*'){

 mul_1_arg.num1=n1;
 mul_1_arg.num2=n2;
 mul_1_arg.operator=opr;

 result_3 = mul_1(&mul_1_arg, clnt);
 if (result_3 == (float *) NULL) {
 clnt_perror (clnt, "call failed");
 }
 return *result_3;
}

else if(opr=='/'){

 div_1_arg.num1=n1;
 div_1_arg.num2=n2;
 div_1_arg.operator=opr;

 if(n2 == 0){
 printf("Division by zero is not valid.\n");
 exit(0);
 }else{

 result_4 = div_1(&div_1_arg, clnt);
 if (result_4 == (float *) NULL) {
 clnt_perror (clnt, "call failed");
 }
 return *result_4;
 }
}
}

int main (int argc, char *argv[])
{
 char *host;
 float a,b;
 char op;
 CLIENT *clnt;

 if (argc < 2) {
```

```
 printf ("usage: %s server_host\n", argv[0]);
 exit (1);
 }

 printf("Welcome to Quick Cal!!!\n");
 printf("+ for Addition\n- for Substraction\n* for
Multiplication\n/ for Division\n");
 printf("Enter number 1 :\n");
 scanf("%f",&a);
 printf("Enter number 2 :\n");
 scanf("%f",&b);
 printf("Enter the Operator :\n");
 scanf("%s",&op);

 host = argv[1];

 clnt = clnt_create (host, CALCULATE_PROG,
CALCULATE_VER, "udp");

 if (clnt == NULL) {
 clnt_pcreateerror (host);
 exit (1);
 }

 printf("The Answer = %f\n",calculate_prog_1
(host,a,b,op,clnt));

 clnt_destroy (clnt);

exit (0);
}
```

**calculate_server.c**

```
#include <stdio.h>
#include "calculate.h"

float * add_1_svc(inputs *argp, struct svc_req
*rqstp)
{
 static float result;

 result = argp->num1+argp->num2;
 printf("Got Request : Adding %f and %f\n",argp-
>num1,argp->num2);
 printf("Sent Response : %f\n",result);

 return &result;
}

float * sub_1_svc(inputs *argp, struct svc_req
*rqstp)
{
 static float result;

 result = argp->num1-argp->num2;
 printf("Got Request : substituting %f from
%f\n",argp->num2,argp->num1);
 printf("Sent Response : %f\n",result);

 return &result;
}

float * mul_1_svc(inputs *argp, struct svc_req
*rqstp)
{
 static float result;

 result = argp->num1*argp->num2;
```

```
 printf("Got Request : Multiplying %f by %f\n",argp-
>num1,argp->num2);
 printf("Sent Response : %f\n",result);

 return &result;
}

float * div_1_svc(inputs *argp, struct svc_req
*rqstp)
{
 static float result;

 result = argp->num1/argp->num2;
 printf("Got Request : Dividing %f by %f\n",argp-
>num1,argp->num2);
 printf("Sent Response : %f\n",result);

 return &result;
}
```

**Practical:7**

Write a program to implement date service using RPC.

**date.x**

```
struct Date
{
    char a[64];
};
program Date_PRG
{
    version Date_VER
    {
        Date DATE(Date)=1;
    } = 1;
} = 0x20000007;
```

**date_client.c**

```
#include "date.h"

void date_prg_1(char *host)

{

CLIENT *clnt;

Date *result_1;

Date  date_1_arg;


#ifndef DEBUG

    clnt = clnt_create (host, Date_PRG, Date_VER,
"udp");

if (clnt == NULL)

{

    clnt_pcreateerror (host);

    exit (1);

}

#endif /* DEBUG */


result_1 = date_1(&date_1_arg, clnt);

if (result_1 == (Date *) NULL)

{

    clnt_perror (clnt, "call failed");

}

else

{
```

```
    printf("Server's Reply= %s\n",result_1->a);
}


#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];
date_prg_1 (host);
exit (0);
}
```

**Date_server.c**

```
#include "date.h"
#include <stdio.h>
#include <time.h>

Date *date_1_svc(Date *argp, struct svc_req *rqstp)
{
static Date result;
result=*argp;
time_t t = time(NULL);
struct tm *tm = localtime(&t);
char s[64];
strftime(s, sizeof(s), "%c", tm);
strcpy(result.a,s);
return &result;
}
```

## Output:

**Server Side:** Gcc –o server echo_server.c

**Client Side:**
Gcc –o client echo_client.c
./client 127.0.0.1
Enter the Message:Hello

Reply=>Hello

# Practical-8

## Write a program to implement Echo server using RPC

## echo.x

```
struct echo_ser
{


char a[20];
};
program echo_PRG
{
version echo_VER
{
        echo_ser ECHO(echo_ser) = 1;
} = 1;
} = 0x20000009;
```

**echo_client.c**

```
#include "echo.h"
void echo_prg_1(char *host)
{
CLIENT *clnt;
echo_ser *result_1;
echo_ser  echo_1_arg;

#ifndef DEBUG
        clnt = clnt_create (host, echo_PRG, echo_VER, "udp");
        if (clnt == NULL)
        {
                clnt_pcreateerror (host);
                exit (1);
        }
#endif /* DEBUG */
printf("Enter the string:");
scanf("%s",echo_1_arg.a);
result_1 = echo_1(&echo_1_arg, clnt);
if (result_1 == (echo_ser *) NULL)
{
```

```
clnt_perror (clnt, "call failed");
}
else
{
}
printf("SERVER REPLIES : %s \n",result_1->a);

#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];
        echo_prg_1 (host);
        exit (0);
```

```
}
```

**echo_server.c**

```
#include "echo.h"


echo_ser *echo_1_svc(echo_ser *argp, struct svc_req *rqstp)
{
static echo_ser result;
result = *argp;
return &result;
}
```

**Practical:9**

Write a program to implement Chat server using RPC

**chat.x**

```
struct operand {
        char myname[100];
        char message[100] ;
        char yourname[100] ;
};
program MESSAGEPROG {
  version PRINTMESSAGEVERS {
    operand PRINTMESSAGE(operand) = 1;
  } = 1;
} = 0x20000001;
```

**chat_client.c**

```
#include "chat.h"
#include<ctype.h>
#include<string.h>
#include<pthread.h>
void *chatting(void *);
int main (int argc, char *argv[])
{
	if (argc != 3 ) {
		// printf ("usage: %s server_host\n", argv[0]);
		exit (1);
	}
	operand printmessage_1_arg;
	operand printmessages_1_arg;
	strcpy(printmessage_1_arg.myname,argv[1]);
	strcpy(printmessage_1_arg.message,argv[2]);
	strcpy(printmessages_1_arg.myname,argv[1]);
	strcpy(printmessages_1_arg.message,argv[2]);
	int choice;
	printf("What do u want to do \n1.Group Chat 2. Indivual Chat\n");
	scanf("%d", &choice);
	if(choice==2)
	{ // FILE *fp ; //fp = fopen("messages","a+");
		CLIENT *clnt;
		char *server;
		server=(char*)malloc(100*sizeof(char)) ;
		operand *result_1;
		printf("enter ip of frnd\n") ;
		scanf("%s",server) ;
		char *namefrnd ;
		namefrnd = (char*)malloc(100*sizeof(char)) ;
		printf("enter his name\n") ;
		scanf("%s",namefrnd) ;
		printmessage_1_arg.yourname[0]='\0' ;
		strcat(printmessage_1_arg.yourname,namefrnd) ;

		clnt = clnt_create (server, MESSAGEPROG, PRINTMESSAGEVERS,"tcp");

		if (clnt == NULL) {
			//clnt_pcreateerror (server);
			exit (1);
		}
		while(1)
		{
		result_1 = (operand*)printmessage_1(&printmessage_1_arg, clnt);
```

```
		if (result_1 == (operand *) NULL)
		{
			clnt_perror (clnt, "call failed");
		}

		printf("%s : %s \n",result_1->myname,result_1->message) ;
		// fprintf(fp,"%s : %s \n",result_1->myname,result_1->message);
	//fclose(fp) ; printf("REPLY\n") ;
		scanf("%s",printmessage_1_arg.message);
		if(strcmp(printmessage_1_arg.message,"abuse")==0)
			exit(1) ;
	}

	clnt_destroy (clnt);

		}
		else
		{
			printf("enter number of people u want to add in");
			int n;
			scanf("%d",&n);
	int iret1 ; int iret2 ;
  pthread_t threads1, threads2;
	iret1 = pthread_create(&threads1,NULL,chatting,&printmessage_1_arg);
	sleep(10) ;
	iret2 = pthread_create(&threads2,NULL,chatting,&printmessages_1_arg);

		pthread_join(threads1,NULL);
		pthread_join(threads2,NULL);
}
	}
void *chatting(void *optn) {
	operand opt = *((operand*)(optn)) ;
	CLIENT *clnt;
	char *server;
	server=(char*)malloc(100*sizeof(char)) ;
	operand *result_1;
	printf("enter ip of frnd\n") ;
	scanf("%s",server) ;
	char *namefrnd ;
	namefrnd = (char*)malloc(100*sizeof(char)) ;
	printf("enter his name\n") ;
	scanf("%s",namefrnd) ;
	opt.yourname[0]='\0' ;
	strcat(opt.yourname,namefrnd) ;

	clnt = clnt_create (server, MESSAGEPROG, PRINTMESSAGEVERS,"udp");

	if (clnt == NULL) {
```

```
                //clnt_pcreateerror (server);
                exit (1);
        }
        while(1) {
        result_1 = (operand*)printmessage_1(&opt, clnt);
        if (result_1 == (operand *) NULL) {
                clnt_perror (clnt, "call failed");
}

        printf("%s : %s \n",result_1->myname,result_1->message) ;

        printf("REPLY\n") ;
    scanf("%s",opt.message);
        if(strcmp(opt.message,"abuse")==0)
                exit(1) ;
}

        clnt_destroy (clnt);

}
```

**chat_server.c**

```
#include "chat.h"
operand * printmessage_1_svc(operand *argp, struct svc_req *rqstp)
{
	static operand result;
	FILE *fp ;
	fp=fopen("messages","a+") ;
	char *input1 ;
	char *input2 ;
	input1=(char*)malloc(100*sizeof(char));
	input2=(char*)malloc(100*sizeof(char));
	printf("%s : %s \n",argp->myname,argp->message) ;
	fprintf(fp,"%s : %s \n",argp->myname,argp->message);
	result.message[0]='\0' ;
	result.myname[0]='\0' ;
	result.yourname[0]='\0' ;
	strcat(result.yourname,argp->myname) ;
	strcat(result.myname,argp->yourname) ;

	printf("REPLY\n") ;
	scanf("%s",input2) ;
	strcat(result.message,input2) ;
	fclose(fp) ;
	return &result;
}
```

## Practical: 10

## Design a Distributed Application for performing arithmetic operations using Message passing Interface (MPI) for remote computation.

### Server.c

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;
     socklen_t clilen;
     char buffer[256];
     struct sockaddr_in serv_addr, cli_addr;
     int n;
     if (argc < 2) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
     if (bind(sockfd, (struct sockaddr *) &serv_addr,
              sizeof(serv_addr)) < 0)
              error("ERROR on binding");
     listen(sockfd,5);
     clilen = sizeof(cli_addr);
     newsockfd = accept(sockfd,
                 (struct sockaddr *) &cli_addr,
                 &clilen);
     if (newsockfd < 0)
          error("ERROR on accept");
     bzero(buffer,256);
     n = read(newsockfd,buffer,255);
```

```
     if (n < 0) error("ERROR reading from socket");
     printf("Here is the message: %s\n",buffer);
     n = write(newsockfd,"I got your message",18);
     if (n < 0) error("ERROR writing to socket");
     close(newsockfd);
     close(sockfd);
     return 0;
}
```

**Client.c**

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);
```

```
    if (connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr)) <
0)
        error("ERROR connecting");
    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
    if (n < 0)
         error("ERROR writing to socket");
    bzero(buffer,256);
    n = read(sockfd,buffer,255);
    if (n < 0)
         error("ERROR reading from socket");
    printf("%s\n",buffer);
    close(sockfd);
    return 0;
}
```

**Practical: 11**

Write a program to implement Echo SOCKET in JAVA

**EchoServer.java**

```
import java.io.*;
import java.net.*;

public class EchoServer
{
	public EchoServer(int portnum)
	{
		try
		{
			server = new ServerSocket(portnum);
		}

		catch (Exception err)
		{
			System.out.println(err);
		}
	}

	public void serve()
	{
		try
		{
			while (true)
			{
				Socket client = server.accept();
				BufferedReader r = new BufferedReader(new
InputStreamReader(client.getInputStream()));
				PrintWriter w = new PrintWriter(client.getOutputStream(), true);
				w.println("This is EchoServer Program. Type 'bye' to close.");
				String line;
				do
				{
					line = r.readLine();
					if ( line != null )
						w.println(" Echo message from server is: "+ line);
				}
				while ( !line.trim().equals("bye") );
				client.close();
			}
		}
		catch (Exception err)
```

```
		{
			System.err.println(err);
		}
	}

	public static void main(String[] args)
	{
		EchoServer s = new EchoServer(9999);
		s.serve();
	}

	private ServerSocket server;
}
```

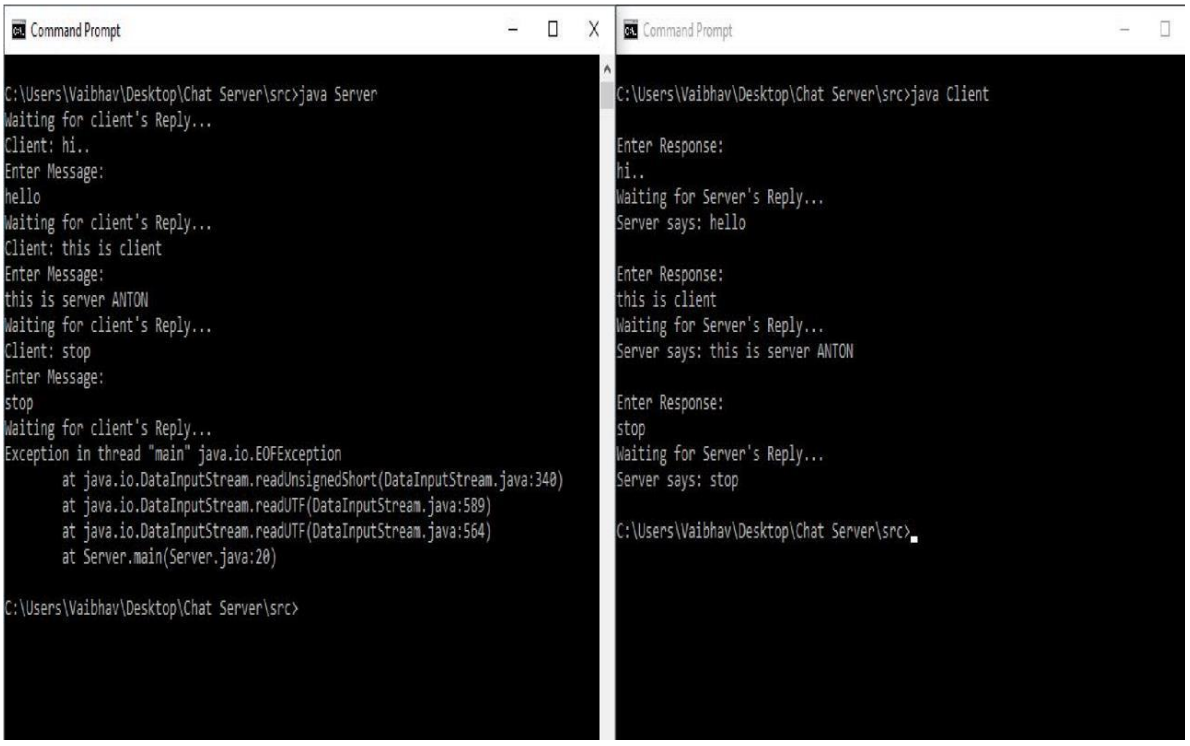
**EchoClient.java**

```
import java.io.*;
import java.net.*;

public class EchoClient
{
	public static void main(String[] args)
	{
		try
		{
			Socket s = new Socket("127.0.0.1", 9999);
			BufferedReader r = new BufferedReader(new
InputStreamReader(s.getInputStream()));
			PrintWriter w = new PrintWriter(s.getOutputStream(), true);
			BufferedReader con = new BufferedReader(new
InputStreamReader(System.in));
			String line;
			do
			{
				line = r.readLine();
				if ( line != null )
					System.out.println(line);
				line = con.readLine();
				w.println(line);
			}
			while ( !line.trim().equals("bye") );
		}
		catch (Exception err)
		{
```

```
                    System.err.println(err);
                }
            }
}
```

# OUTPUT



**Practical:12**

**Implement a program to solve the producer-consumer problem using Thread in JAVA**

```
class Q
{
        int n;
        boolean valueSet=false;

        synchronized int get()
        {
                if(!valueSet)
                try
                {
                        wait();
                }
                catch(InterruptedException e)
                {
                        System.out.println("Interrupted");
                }
    System.out.println("Got" + n);
                valueSet=false;
                notify();
                return n;
  }

        synchronized void put(int n)
        {
                if(valueSet)
                try
                {
                        wait();
                }
                catch(InterruptedException e)
                {
                        System.out.println("Interrupted");
                }
    this.n=n;
                valueSet=true;
    System.out.println("Put" + n);
                notify();
  }
```

```
}

class Producer implements Runnable
{
	Q q;
	Producer(Q q)
	{
		this.q=q;
		new Thread(this,"Producer").start();
	}
	public void run()
	{
		int i=0;
		while(true)
		{
			q.put(i++);
		}
	}
}

class Consumer implements Runnable
{
	Q q;
	Consumer(Q q)
	{
		this.q=q;
		new Thread(this,"Consumer").start();
	}
	public void run()
	{
		while(true)
		{
			q.get();
		}
	}
}

class PCFixed
{
	public static void main(String args[])
	{
```

```
        Q q=new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Ctrl+C to Stop");
    }
}
```