CSE 660 - Lab 5

Amit Lawanghare

10 May 10, 2017

Part 1.

1. *What is XDR and what is it for?*

XDR is the routines of library for "external data representation". It allows us to explain "arbitrary data structures in a machine-independent fashion", and it uses for Remote Procedure Call.

2. *How do you compile an input file into XDR routine:*

To compile file we use the operation –c

3. *$rpcgen -C -a rand.x explanation*

In this command, -C generates code in ANSI C, this option also generates code that could be compiled with the C++ compiler. –a generates all the files including sample code for client and server side.

Writing own random number generator

Write our own 20 Random numbers Generator:

Code:

In Client:

```
//random_client.c
#include "random.h"
double radn_prog_1(char *host)
{
          CLIENT *clnt; void *result_1;
          long initialize_random_1_arg;
          double *result_2;
          char *get_next_random_1_arg;
          #ifndefDEBUG
                    clnt = clnt_create(host, RADN_PROG, RAND_VERS, "udp");
          if (clnt == NULL) {
                    clnt_pcreateerror(host);
                    exit(1);
          }
#endif /* DEBUG */
          result_1 = initialize_random_1(&initialize_random_1_arg, clnt);
          if (result_1 == (void *)NULL) {
                    clnt_perror(clnt, "call failed");
          }
          result_2 = get_next_random_1((void*)&get_next_random_1_arg, clnt);
          if (result_2 == (double *)NULL) {
                    clnt_perror(clnt, "call failed");
          }
          #ifndefDEBUG
                    clnt_destroy(clnt);
```

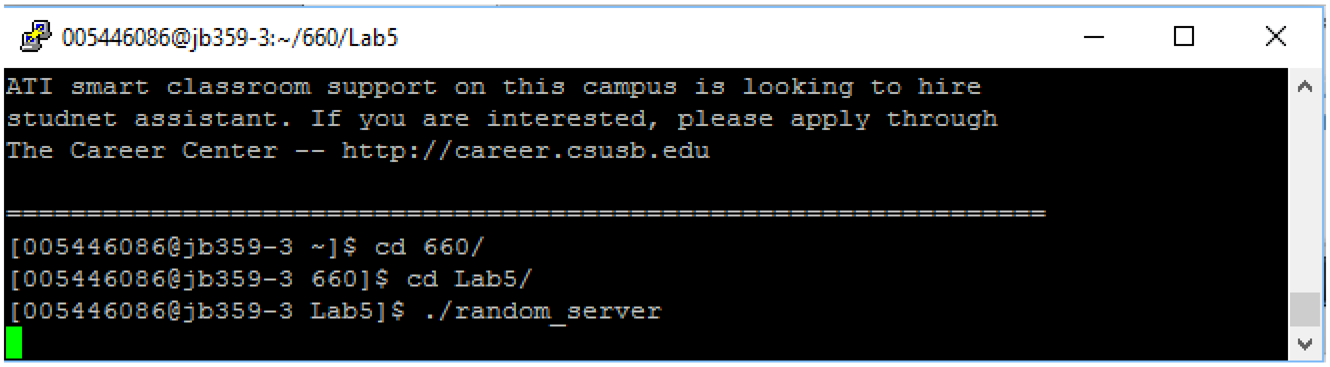
```c
#endif /* DEBUG */
	return *result_2;
}
int main(int argc, char *argv[])
{
	char *host;
	if (argc < 2) {
		printf("usage: %s server_host\n", argv[0]);
		exit(1);
	}
	host = argv[1];
	radn_prog_1(host);
	double x;
	int i;
	printf("\n twenty random numbers ");
	for (i = 0; i < 20; ++i) {
		x = radn_prog_1(host);
		printf(" %f, ", x);
	}
	exit(0);
}
```

In Server:

```c
//random_server.c
#include "random.h"
#include <time.h>
void * initialize_random_1_svc(long *argp, struct svc_req *rqstp)
{
	static char * result;
	/*
	* insert server code here
	*/
	return (void *)&result;
}
double * get_next_random_1_svc(void *argp, struct svc_req *rqstp)
{
	static double result;
	/*
	* insert server code here
	*/
	srand(time(NULL));
	result += rand();
	// if ( result >= 1.0 )
	result -= 0.713;
	return &result;
}
```

Ouput :

Server program is running on machine *jb359-3* as given below

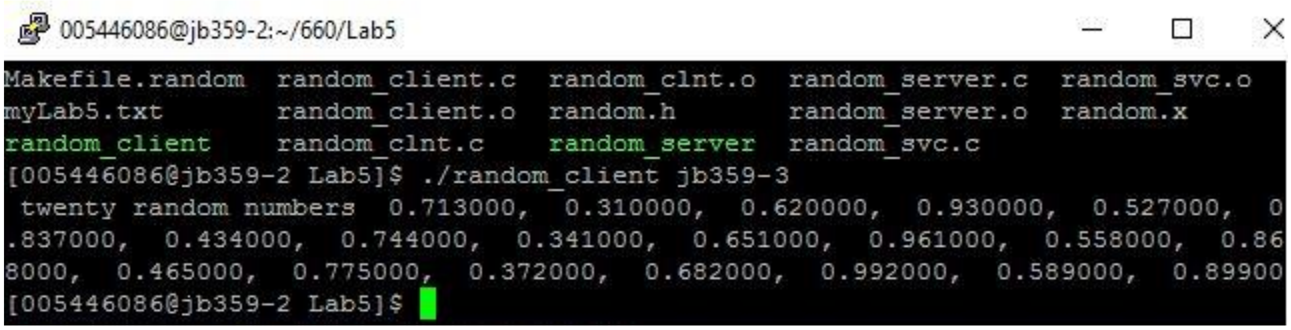


005446086@jb359-3:~/660/Lab5

```
ATI smart classroom support on this campus is looking to hire
studnet assistant. If you are interested, please apply through
The Career Center -- http://career.csusb.edu

=================================================================
[005446086@jb359-3 ~]$ cd 660/
[005446086@jb359-3 660]$ cd Lab5/
[005446086@jb359-3 Lab5]$ ./random_server
```

Client program is running on machine *jb359-2* and invokes server process which is present on *jb359-3* and it returned random numbers generated at server side (*jb359-3*)



005446086@jb359-2:~/660/Lab5

```
Makefile.random  random_client.c  random_clnt.o   random_server.c  random_svc.o
myLab5.txt       random_client.o  random.h        random_server.o  random.x
random_client    random_clnt.c    random_server   random_svc.c
[005446086@jb359-2 Lab5]$ ./random_client jb359-3
 twenty random numbers  0.713000,  0.310000,  0.620000,  0.930000,  0.527000,  0
.837000,  0.434000,  0.744000,  0.341000,  0.651000,  0.961000,  0.558000,  0.86
8000,  0.465000,  0.775000,  0.372000,  0.682000,  0.992000,  0.589000,  0.89900
[005446086@jb359-2 Lab5]$
```

Part 2

Java Remote Method Invocation (RMI)

Server jb359-3

Client jb359-2

Source Code –

Client.java

```
package rmi;

//Client.java
import java.rmi.Naming;
import java.util.Scanner;

public class Client {
	private static Scanner reader;

	public static void main(String[] argv) {
		reader = new Scanner(System.in);
		System.out.print("Enter the first number: ");
		double firstNumber = reader.nextDouble();
		System.out.print("Enter the second number: ");
		double secondNumber = reader.nextDouble();
		double result;
		try {
			MyAdditionInterface addtionRemoteObject = (MyAdditionInterface) Naming
					.lookup("rmi://" + argv[0] + "/RmiAddition");
			result = addtionRemoteObject.add(firstNumber, secondNumber);
			System.out.printf("The sum is: %f\n", result);
		} catch (Exception e) {
			System.out.println("RMI Addition exception: " + e);
		}
	}
}
```

Interface

```
package rmi;

//RmiAdditionInterface.java
import java.rmi.*;

public interface MyAdditionInterface extends Remote {
	/**
	 *
	 * @return the sum of the two numbers
	 * @exception RemoteException
	 *                 if the remote invocation fails.
	 */
	public double add(double firstNumber, double secondNumber) throws RemoteException;
}
```

RmiAddition

```
package rmi;

//RmiAddition.java
import java.rmi.*;
import java.rmi.server.*;

public class RmiAddition extends UnicastRemoteObject implements MyAdditionInterface {
        private static final long serialVersionUID = 1L;

        /**
         * Construct a remote object
         */
        public RmiAddition() throws RemoteException {
        }

        /**
         * Implementation of the remotely method.
         *
         * @return the sum of the remote object
         * @exception RemoteException
         *                  if the remote invocation fails.
         */
        @Override
        public double add(double firstNumber, double secondNumber) throws RemoteException {
                return firstNumber + secondNumber;
        }
}
```

```
Server code

package rmi;

//Server.java
import java.rmi.*;

public class Server {
        public static void main(String[] argv) {
                try {
                        Naming.rebind("RmiAddition", new RmiAddition());
                        System.out.println("Rmi Addition Server is ready to use.");
                } catch (Exception e) {
                        System.out.println("Rmi Addition Server has been failed: " + e);
                }
        }
}
```

Steps for execution –

1. Compile programs with java compiler
2. Search for rmiregistry and add it in path as below

```
[005446086@jb359-2 rmi]$ locate rmiregistry | grep bin
/opt/Xilinx/.xinstall/DocNav/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/.xinstall/Vivado_2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/.xinstall/xic/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java/lin/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java/lin64/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java6/lin/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java6/lin64/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/PlanAhead/tps/lnx32/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/PlanAhead/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/Vivado/2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/Vivado_HLS/2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/xic/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx.bak/Downloads/2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/android-studio/jre/bin/rmiregistry
/opt/android-studio/jre/jre/bin/rmiregistry
/opt/scilab-5.5.2/thirdparty/java/bin/rmiregistry
/usr/java/jdk1.8.0_121/bin/rmiregistry
/usr/java/jdk1.8.0_121/jre/bin/rmiregistry
/usr/java/jdk1.8.0_60/bin/rmiregistry
/usr/java/jdk1.8.0_60/jre/bin/rmiregistry
/usr/java/jdk1.8.0_65/bin/rmiregistry
/usr/java/jdk1.8.0_65/jre/bin/rmiregistry
/usr/java/jdk1.8.0_66/bin/rmiregistry
/usr/java/jdk1.8.0_66/jre/bin/rmiregistry
/usr/java/jdk1.8.0_72/bin/rmiregistry
/usr/java/jdk1.8.0_72/jre/bin/rmiregistry
/usr/java/jdk1.8.0_77/bin/rmiregistry
```

```
/usr/java/jdk1.8.0_77/jre/bin/rmiregistry
/usr/java/jre1.8.0_71/bin/rmiregistry
/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.141-2.6.10.1.el7_3.x86_64/jre/bin/rmiregistry
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/bin/rmiregistry
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/jre/bin/rmiregistry
/usr/local/MATLAB/R2015b/sys/java/jre/glnxa64/jre/bin/rmiregistry
[005446086@jb359-2 rmi]$ /usr/java/jre1.8.0_71/bin/rmiregistry &
```

3. Start Server.
4. Ping from Client machine (JB 359-3) to Server (JB 359-2) machine and get IP address

```
[005446086@jb359-3 rmi]$ ping jb359-2
PING jb359-2.cse.csusb.edu (139.182.148.122) 56(84) bytes of data.
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=1 ttl=64 time=0.433 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=2 ttl=64 time=0.200 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=3 ttl=64 time=0.238 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=4 ttl=64 time=0.213 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=5 ttl=64 time=0.182 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=6 ttl=64 time=0.279 ms
^C
--- jb359-2.cse.csusb.edu ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 0.182/0.257/0.433/0.085 ms
```

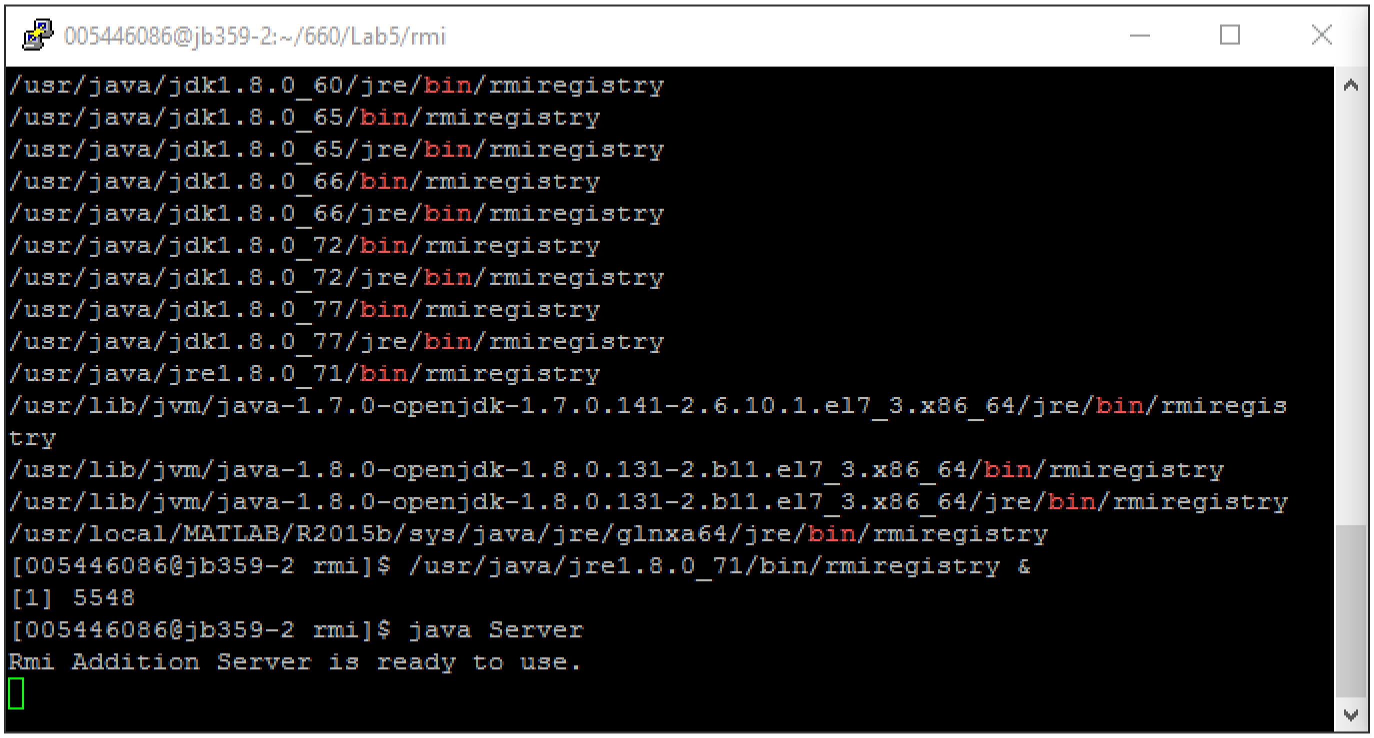
5. Connect client to server machine.

```
[005446086@jb359-3 rmi]$ java Client 139.182.148.122
Enter the first number: 10
Enter the second number: 15
The sum is: 25.000000
[005446086@jb359-3 rmi]$
```

So sum was calculated correctly by server (jb359-2) and utilized by client (jb359-3)

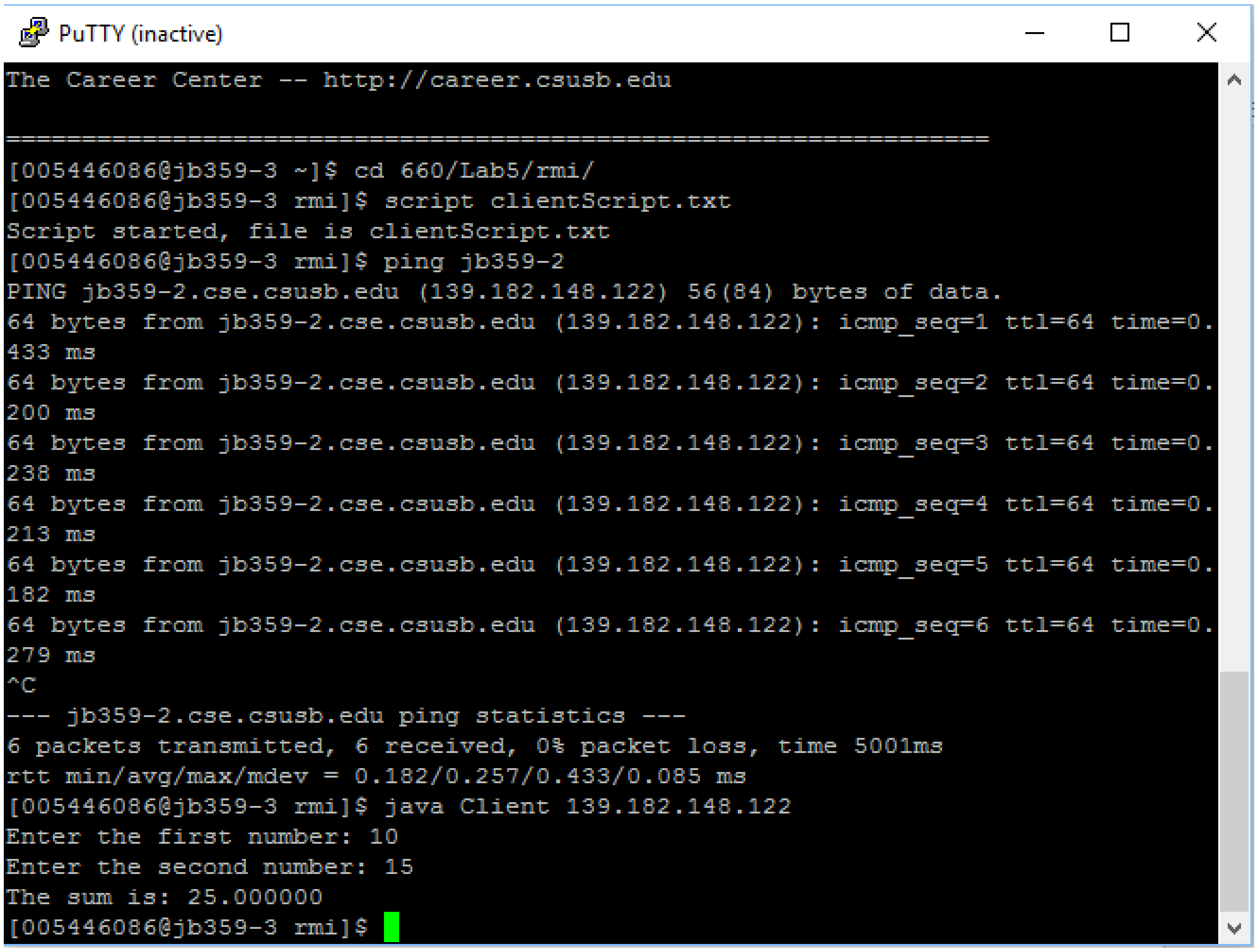
Output

Server (jb359-2)



```
/usr/java/jdk1.8.0_60/jre/bin/rmiregistry
/usr/java/jdk1.8.0_65/bin/rmiregistry
/usr/java/jdk1.8.0_65/jre/bin/rmiregistry
/usr/java/jdk1.8.0_66/bin/rmiregistry
/usr/java/jdk1.8.0_66/jre/bin/rmiregistry
/usr/java/jdk1.8.0_72/bin/rmiregistry
/usr/java/jdk1.8.0_72/jre/bin/rmiregistry
/usr/java/jdk1.8.0_77/bin/rmiregistry
/usr/java/jdk1.8.0_77/jre/bin/rmiregistry
/usr/java/jre1.8.0_71/bin/rmiregistry
/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.141-2.6.10.1.el7_3.x86_64/jre/bin/rmiregis
try
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/bin/rmiregistry
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/jre/bin/rmiregistry
/usr/local/MATLAB/R2015b/sys/java/jre/glnxa64/jre/bin/rmiregistry
[005446086@jb359-2 rmi]$ /usr/java/jre1.8.0_71/bin/rmiregistry &
[1] 5548
[005446086@jb359-2 rmi]$ java Server
Rmi Addition Server is ready to use.
```

Client (jb359-3)



PuTTY (inactive)

```
The Career Center -- http://career.csusb.edu

=================================================================
[005446086@jb359-3 ~]$ cd 660/Lab5/rmi/
[005446086@jb359-3 rmi]$ script clientScript.txt
Script started, file is clientScript.txt
[005446086@jb359-3 rmi]$ ping jb359-2
PING jb359-2.cse.csusb.edu (139.182.148.122) 56(84) bytes of data.
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=1 ttl=64 time=0.
433 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=2 ttl=64 time=0.
200 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=3 ttl=64 time=0.
238 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=4 ttl=64 time=0.
213 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=5 ttl=64 time=0.
182 ms
64 bytes from jb359-2.cse.csusb.edu (139.182.148.122): icmp_seq=6 ttl=64 time=0.
279 ms
^C
--- jb359-2.cse.csusb.edu ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 0.182/0.257/0.433/0.085 ms
[005446086@jb359-3 rmi]$ java Client 139.182.148.122
Enter the first number: 10
Enter the second number: 15
The sum is: 25.000000
[005446086@jb359-3 rmi]$
```

Part 3

Server jb359-3

Client jb359-2

Source Code

Client.java

```
package randomN;

//Client.java
import java.rmi.Naming;

public class Client {
        public static void main(String[] argv) {
                /* getting the argument from the command line */
                int numberArgument = argv.length;
                if (numberArgument != 2 && numberArgument != 4) {
                        System.out.println("Usage (check arguments): Server <host> <number of outputs> <Min> <Max>");
                        System.exit(0);
                }
                int[] randomArray = new int[Integer.parseInt(argv[1])];
                try {
                        RmiRandomNumberCreatorInterface randomGenRemoteObject = (RmiRandomNumberCreatorInterface) Naming
                                        .lookup("rmi://" + argv[0] + "/RmiRandomCr");
                        if (numberArgument == 2) {
                                randomArray = randomGenRemoteObject.nRandGen(Integer.parseInt(argv[1]));
                        }
                        /* if the range is provided */
                        if (numberArgument == 4) {
                                randomArray = randomGenRemoteObject.nRandGenWithRange(Integer.parseInt(argv[1]),
                                                Integer.parseInt(argv[2]), Integer.parseInt(argv[3]));
                        }

                        for (int i = 0; i < Integer.parseInt(argv[1]); i++) {
                                System.out.printf("%d\n", randomArray[i]);
                        }
                } catch (Exception e) {
                        System.out.println("RMI Random Number creator exception: " + e);
                }
        }
}
```

Server.java

```
package randomN;

//Server.java
import java.rmi.*;

public class Server {
        public static void main(String[] argv) {
                try {
                        Naming.rebind("RmiRandomCr", new RmiRandomNumberCreator());
                        System.out.println("Random Number creator Server is ready.");
                } catch (Exception e) {
                        System.out.println("Rmi Random Number creator Server failed: " + e);
                }
        }
}
```

RmiRandomNumberCreatorInterface.java

```java
package randomN;

//RmiRandGenInterface.java
import java.rmi.*;

public interface RmiRandomNumberCreatorInterface extends Remote {
        /**
         *
         * @return array of n random numbers
         * @exception RemoteException
         *                 if the remote invocation fails.
         */
        public int[] nRandGen(int n) throws RemoteException;

        /**
         *
         * @return array of n random numbers within the range min, max
         * @exception RemoteException
         *                 if the remote invocation fails.
         */
        public int[] nRandGenWithRange(int n, int min, int max) throws RemoteException;
}
```

RmiRandomNumberCreator.java

```java
package randomN;

//RmiRandGen.java
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.util.Random;

public class RmiRandomNumberCreator extends UnicastRemoteObject implements RmiRandomNumberCreatorInterface {
	private static final long serialVersionUID = 1L;

	/**
	 * Construct a remote object
	 *
	 * @param the
	 *           message of the remote object
	 *
	 * @exception RemoteException
	 *                if the object handle cannot be constructed.
	 *
	 */
	public RmiRandomNumberCreator() throws RemoteException {
	}

	/**
	 * Implementation of the remotely method.
	 *
	 * @return an int array with n random number
	 * @exception RemoteException
	 *                if the remote invocation fails.
	 */
	public int[] nRandGen(int n) throws RemoteException {
		int[] randArray = new int[n];
		Random randomGen = new Random();
		for (int i = 0; i < n; i++) {
			randArray[i] = (int) randomGen.nextInt(10000000);
		}
		return randArray;
	}

	/**
	 * Implementation of the remotely method.
	 *
	 * @return array of n random numbers within the range min, max
	 * @exception RemoteException
	 *                if the remote invocation fails.
	 */
	@Override
	public int[] nRandGenWithRange(int n, int min, int max) throws RemoteException {
		int[] randArray = new int[n];
		Random randomGen = new Random();
		for (int i = 0; i < n; i++) {
			randArray[i] = min + (int) randomGen.nextInt((max - min) + 1);
		}
		return randArray;
	}
}
```

Steps

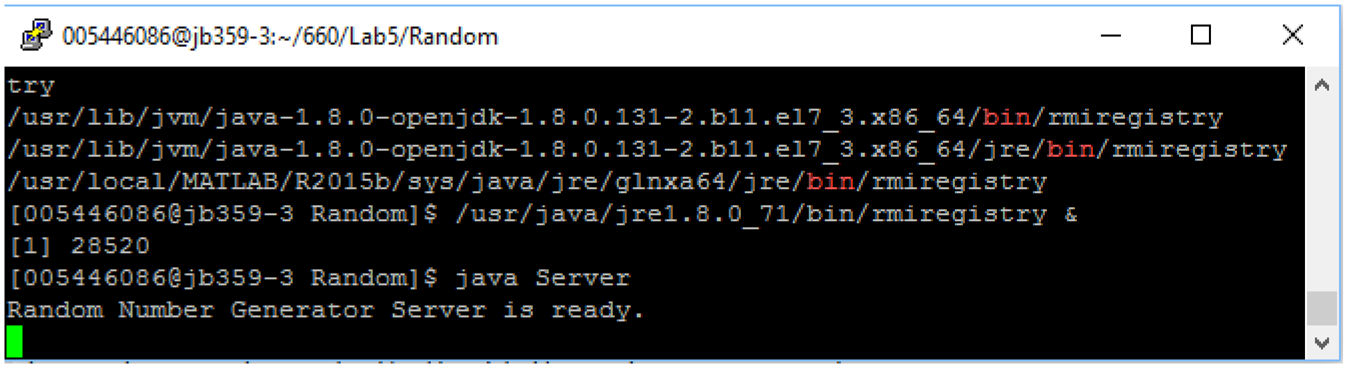
1. Compile all files.

```
[005446086@csex11 Random]$ javac Client.java
[005446086@csex11 Random]$ javac RmiRandGen.java
[005446086@csex11 Random]$ javac RmiRandGenInterface.java
[005446086@csex11 Random]$ javac Server.java
```

2. Search for rmiregistry and add it in path as below

```
[005446086@jb359-2 rmi]$ locate rmiregistry | grep bin
/opt/Xilinx/.xinstall/DocNav/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/.xinstall/Vivado_2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/.xinstall/xic/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java/lin/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java/lin64/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java6/lin/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/ISE/java6/lin64/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/PlanAhead/tps/lnx32/jre/bin/rmiregistry
/opt/Xilinx/14.7/ISE_DS/PlanAhead/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/Vivado/2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/Vivado_HLS/2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx/xic/tps/lnx64/jre/bin/rmiregistry
/opt/Xilinx.bak/Downloads/2016.4/tps/lnx64/jre/bin/rmiregistry
/opt/android-studio/jre/bin/rmiregistry
/opt/android-studio/jre/jre/bin/rmiregistry
/opt/scilab-5.5.2/thirdparty/java/bin/rmiregistry
/usr/java/jdk1.8.0_121/bin/rmiregistry
/usr/java/jdk1.8.0_121/jre/bin/rmiregistry
/usr/java/jdk1.8.0_60/bin/rmiregistry
/usr/java/jdk1.8.0_60/jre/bin/rmiregistry
/usr/java/jdk1.8.0_65/bin/rmiregistry
/usr/java/jdk1.8.0_65/jre/bin/rmiregistry
/usr/java/jdk1.8.0_66/bin/rmiregistry
/usr/java/jdk1.8.0_66/jre/bin/rmiregistry
/usr/java/jdk1.8.0_72/bin/rmiregistry
/usr/java/jdk1.8.0_72/jre/bin/rmiregistry
/usr/java/jdk1.8.0_77/bin/rmiregistry
/usr/java/jdk1.8.0_77/jre/bin/rmiregistry
/usr/java/jre1.8.0_71/bin/rmiregistry
/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.141-2.6.10.1.el7_3.x86_64/jre/bin/rmiregistry
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/bin/rmiregistry
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/jre/bin/rmiregistry
/usr/local/MATLAB/R2015b/sys/java/jre/glnxa64/jre/bin/rmiregistry
[005446086@jb359-2 rmi]$ /usr/java/jre1.8.0_71/bin/rmiregistry &
[1] 28520
```

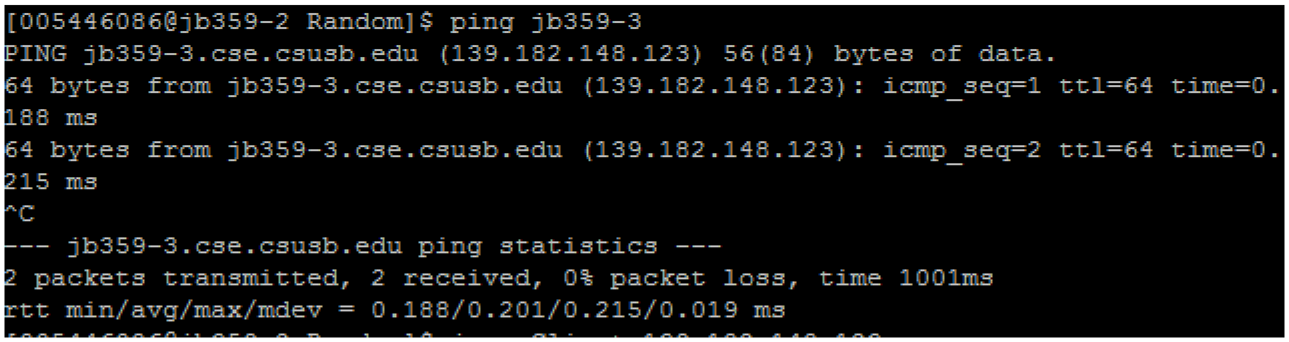
3. Start Server



005446086@jb359-3:~/660/Lab5/Random

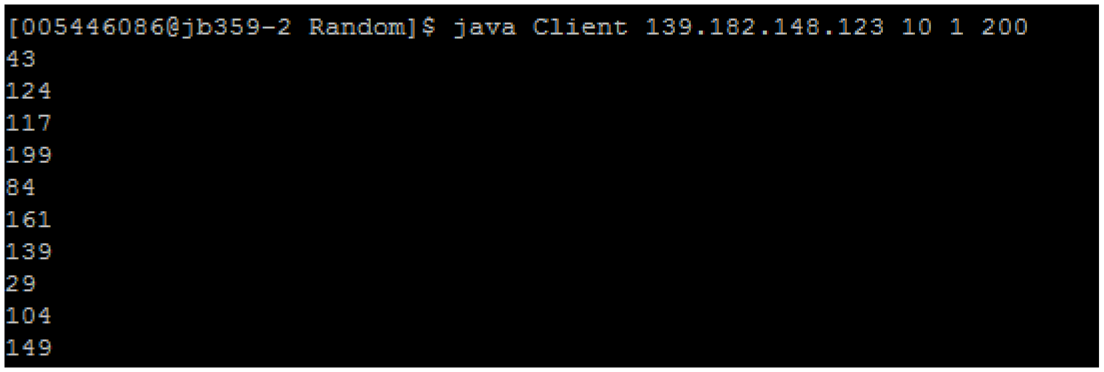
```
try
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/bin/rmiregistry
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/jre/bin/rmiregistry
/usr/local/MATLAB/R2015b/sys/java/jre/glnxa64/jre/bin/rmiregistry
[005446086@jb359-3 Random]$ /usr/java/jre1.8.0_71/bin/rmiregistry &
[1] 28520
[005446086@jb359-3 Random]$ java Server
Random Number Generator Server is ready.
```

4. Search remote machine IP



```
[005446086@jb359-2 Random]$ ping jb359-3
PING jb359-3.cse.csusb.edu (139.182.148.123) 56(84) bytes of data.
64 bytes from jb359-3.cse.csusb.edu (139.182.148.123): icmp_seq=1 ttl=64 time=0.
188 ms
64 bytes from jb359-3.cse.csusb.edu (139.182.148.123): icmp_seq=2 ttl=64 time=0.
215 ms
^C
--- jb359-3.cse.csusb.edu ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.188/0.201/0.215/0.019 ms
```

5. Execute with 10 numbers between 1 and 100.



```
[005446086@jb359-2 Random]$ java Client 139.182.148.123 10 1 200
43
124
117
199
84
161
139
29
104
149
```