

CSCI537
Introduction to Distributed Computing

Report on
RPCgen A3

**Dept. of Computer & Information Science,
IUPUI**

Name: Keyur Kirti Mehta

Contents

1. Introduction.....	3
2. Design Decisions and Implementation	3
1. RPCgen	3
2. Operation.....	3
1. Host Name:	3
2. MergeSort:.....	4
3. Encrypted Echo:.....	4
4. List the files:.....	4
5. Add Complex numbers:.....	4
3. Screenshots.....	4
1. Host Name	4
2. Merge and Sort 2 list	5
3. Encrypt the String.....	5
4. List all the File	5
5. Add Complex Number	6
6. Multi-Threading.....	6
4. Conclusion	7
References	7

1. Introduction

The assignment is to implement different operations using the RPCgen library. This library will be implemented in C programming language which will make remote procedure calls to the server to perform the operation.

The assignment is implemented using multi-threaded server so as multiple client can be connected and request for file transmission.

2. Design Decisions and Implementation

1. RPCgen

RPCgen tool generates the code in C programming language which implements the RPC protocol. This involves 5 components as:

- Client – It calls the client stub. This is local call made by the client. It sends the parameters along with the call.
- Client Stub – Stub consolidates the parameters together called as marshalling and sends to the runtime.
- RPC Communication Package (aka RPCRuntime) – This will send the call to the server machine. It actually transfer the call to the server stub.
- Server Stub – It unpacks the parameters sends by the client and send the request to actual server. This is also called as unmarshalling.
- Server – The actual processing of the request take place on the server. And it follows the same process to send response back to the client.

RPCgen generates header file which will be common for bother client and server code. The client and server uses this header file for the code. The XDR file defines the data structure mentioned in the header file. It will only be generated when different data structure is used. The client stub and server stub is used to marshal and unmarshal the parameters and actual call. The client code makes the call using the client stub for the server code.

In order to generate the stubs, the .x file with all the signature of the functions and data structure used in it needs to be created. Once the .x file is created, then execute the `rpcgen -C -a <.x>` command in order to generate all the files. It will create header file, XDR file, client and server stubs and client and server code template.

2. Operation

The assignment implements 5 operations using RPC which are as follows:

1. Host Name:

This will return the host name of the server to the client. It will return the string with name of the server which is running.

2. MergeSort:

This operation will accept 2 arrays from the user and marshal them in the structure type. This is passed to the server. Server will merge and sort the both the integer list and return the sorted 1 list back to the client. I have implemented the insertion sort to sort the list of integer in ascending order.

3. Encrypted Echo:

This operation will take the string as an input from the client and encrypt the string on server side using any simple technique. The server will return the encrypted version of the string back to the client. I have used simple encryption technology which will perform the operation on the ASCII value of each character of the string. It will multiply the ASCII value of the char by 3 and add 10 to it.

```
outputEcho[i] = (char) (((*argp)[i] * 3) + 10);
```

4. List the files:

This operation will return the list of files present in the current directory of the server. It will return the string which contains the name of all the files separated by '\n' char.

5. Add Complex numbers:

The user needs to enter the 2 complex number for this operation. Client stub will marshal these numbers in one structure object and send across the server. Server will perform the addition of the 2 complex numbers and send the result back to the client.

3. Screenshots

1. Host Name

```

[mehtake@in-csci-rp001 Assignment3]$ make -f Makefile.RPC_Assign
cc -g -o RPC_Assign_xdr.o RPC_Assign_xdr.c
cc -g -o RPC_Assign_client.o RPC_Assign_client.o RPC_Assign_xdr.o -lnsl
cc -g -o RPC_Assign_svc.o RPC_Assign_svc.c
cc -g -o RPC_Assign_server.o RPC_Assign_server.c
cc -g -o RPC_Assign_server.o RPC_Assign_svc.o RPC_Assign_server.o RPC_Assign_xdr.o -lnsl
[mehtake@in-csci-rp001 Assignment3]$ ./RPC_Assign_server

****Host Name****
The Hostname: in-csci-rp001.cs.lupul.edu

[mehtake@in-csci-rp001 Assignment3]$ ./RPC_Assign_client in-csci-rp001.cs.lupul.edu

*****Welcome to RPC Program *****
1. Get hostname
2. Merge Sort
3. Encrypted Echo
4. List all files
5. Add complex number
6. Exit
Enter your choice:1
The host name is: in-csci-rp001.cs.lupul.edu

*****Welcome to RPC Program *****
1. Get hostname
2. Merge Sort
3. Encrypted Echo
4. List all files
5. Add complex number
6. Exit
Enter your choice:

```

2. Merge and Sort 2 list

The first terminal window shows the compilation of the Merge Sort program. The user runs `make` in the `Assignment3` directory, which compiles `RPC_Assign_client.c` and `RPC_Assign_server.c` into `RPC_Assign_client.o` and `RPC_Assign_server.o`, and then links them into `RPC_Assign_client` and `RPC_Assign_server` executables.

The second terminal window shows the execution of the `RPC_Assign_server` program. The user enters their choice as 2 (Merge Sort). The program prompts for two arrays of integers. The first array is `2 3 5 6 8 11 23 42 54 76` and the second array is `11 76 42 5`. The program then displays the merged and sorted list: `2 3 5 6 8 11 23 42 54 76`.

3. Encrypt the String

The first terminal window shows the execution of the `RPC_Assign_client` program. The user enters their choice as 3 (Encrypted Echo). The program prompts for a string. The user enters `my name is keyur`. The program then displays the encrypted string: `SwjV/S:jGajM:Web`.

The second terminal window shows the execution of the `RPC_Assign_server` program. The user enters their choice as 3 (Encrypted Echo). The program prompts for a string. The user enters `my name is keyur`. The program then displays the encrypted string: `SwjV/S:jGajM:Web`.

4. List all the File

The first terminal window shows the execution of the `RPC_Assign_client` program. The user enters their choice as 4 (List of Files). The program prompts for a directory. The user enters `.`. The program then displays the list of files in the current directory: `Makefile.RPC_Assign RPC_Assign.h RPC_Assign.x RPC_Assign_client.o RPC_Assign_client.c RPC_Assign_server.o RPC_Assign_server.c RPC_Assign_svc.o RPC_Assign_wtr.o RPC_Assign_wtr.c RPC_Assign_wtr.o RPC_Assign_wtr.o RPC_Assign_client RPC_Assign_client.o RPC_Assign_server.o RPC_Assign_server`.

The second terminal window shows the execution of the `RPC_Assign_server` program. The user enters their choice as 4 (List of Files). The program prompts for a directory. The user enters `.`. The program then displays the list of files in the current directory: `Makefile.RPC_Assign RPC_Assign.h RPC_Assign.x RPC_Assign_client.o RPC_Assign_client.c RPC_Assign_server.o RPC_Assign_server.c RPC_Assign_svc.o RPC_Assign_wtr.o RPC_Assign_wtr.c RPC_Assign_wtr.o RPC_Assign_wtr.o RPC_Assign_client RPC_Assign_client.o RPC_Assign_server.o RPC_Assign_server`.

5. Add Complex Number

The left terminal window shows the compilation of the program using `makefile.RPC_Assign`. The right terminal window shows the execution of the program, which displays a menu with options 1 through 6. Option 5, 'Add complex number', is selected. The program prompts for the real and imaginary parts of two complex numbers. The first number is 3 + 5i and the second is 7 + (-10)i. The program then outputs the result: 'The addition of 2 complex no is : 10 + (-5)i'.

```

makefile.RPC_Assign
RPC_Assign.h
RPC_Assign.s
RPC_Assign_client.c
RPC_Assign_client.o
RPC_Assign_server.c
RPC_Assign_server.o
RPC_Assign_exe.c
RPC_Assign_exe.o
RPC_Assign_exe
RPC_Assign_client.o
RPC_Assign_server.o
RPC_Assign_exe

****Addition of Complex Number****
The 1st complex no is: 3 + 5i
The 2nd complex no is: 7 + (-10)i
The addition of 2 complex no is : 10 + (-5)i

*****Welcome to RPC Program *****
1. Get hostname
2. Merge Sort
3. Encrypted Echo
4. List all files
5. Add complex number
6. Exit
Enter your choice:5
Enter the 1st real part: 3
Enter the 1st imp part: 5
Enter the 2nd real part: 7
Enter the 2nd imp part: -10
The addition of 2 complex no is : 10 + (-5)i

*****Welcome to RPC Program *****
1. Get hostname
2. Merge Sort
3. Encrypted Echo
4. List all files
5. Add complex number
6. Exit
Enter your choice:

```

6. Multi-Threading

The left terminal window shows the compilation of the program using `makefile.RPC_Assign`. The middle terminal window shows the execution of the program, which displays a menu with options 1 through 6. Option 3, 'Encrypted Echo', is selected. The program prompts for an element of a list array. The user enters 2. The program then outputs the merged sorted list: 'The Merged sorted list is 1 2 2 3 3 4 5 6 6 6'. The right terminal window shows the execution of the program, which displays a menu with options 1 through 6. Option 1, 'Get hostname', is selected. The program outputs the host name: 'The host name is: in-csci-rpc01.cs.lupui.edu'.

```

makefile.RPC_Assign
RPC_Assign.h
RPC_Assign.s
RPC_Assign_client.c
RPC_Assign_client.o
RPC_Assign_server.c
RPC_Assign_server.o
RPC_Assign_exe.c
RPC_Assign_exe.o
RPC_Assign_exe
RPC_Assign_client.o
RPC_Assign_server.o
RPC_Assign_exe

****Merge Sort****
Enter your choice:3
Enter the element of list array: 2
2
3
4
5
6
7
8
9
10
The Merged sorted list is
1 2 2 3 3 4 5 6 6 6

*****Welcome to RPC Program *****
1. Get hostname
2. Merge Sort
3. Encrypted Echo
4. List all files
5. Add complex number
6. Exit
Enter your choice:1
The host name is: in-csci-rpc01.cs.lupui.edu

*****Welcome to RPC Program *****
1. Get hostname
2. Merge Sort
3. Encrypted Echo
4. List all files
5. Add complex number
6. Exit
Enter your choice:

```

4. Conclusion

The assignment implements the RPCgen library successfully in C. It provide the knowledge of the Remote Procedure Call with help of auto generated stubs.

Advantage:

1. RPCgen generates all the stubs for client and server which actually performs all the communication between client and server. So, user does not need to worry about remote calls. It is more efficient and makes simpler for programmer to code.
2. Programmer just need to implement the server logic of operation and client control flow.
3. It also supports the multithreading on server side. So, multiple client can send concurrent requests.

Disadvantage:

1. It can pass only 1 arguments during the call. So, all the parameters must be enclosed in structure and pass to the server call.

References

1. <https://docs.oracle.com/cd/E19253-01/816-1435/rpcgenpguide-24243/index.html>
2. https://en.wikipedia.org/wiki/Remote_procedure_call