Lab 09: RPC Programming

OSL, Mon, Oct 22, 2012

Objective:

1. Get familiarized with network RPC programming.

General instructions:

1. This lab is to be done in **groups of two students.**
2. You are **not allowed to exchange code snippets or anything** across groups. In case of any doubts, you are allowed to consult any Internet resource, books, or the course TAs.
3. You will need to use C for coding.

Lab Instructions

You have to implement a client-server application using RPC. The client invokes a function call with appropriate arguments. The server implements the function call on the arguments and returns result to the client. For example if the client invokes “factorial (4)”, the server should return “24”.

Exercise 1: Warm up

1. Go through pages 1-6 of the RPC Programming Guide at: http://www.cse.iitb.ac.in/~sri/cs348/cs378-lab09-rpc-guide.pdf
2. Write the code for *msg.x* (see page 3 of the guide).
3. Execute the command 'rpcgen -a msg.x'
4. What are the files that get created? What is the functionality difference between msg_client.c and msg_clnt.c? Note these in your file lab09.txt.
5. Modify msg_server.c to implement your server (see 'msg_proc.c' on page 3 of the guide).
6. Modify msg_client.c to implement your client (see 'rprintmsg.c' on page 5 of the guide).
7. Compile and create the executables. You can simply rename the 'Makefile.msg' to 'Makefile' and run *make*.
8. Run your server and client (see last few lines on page 6 of the guide). You can also run the server on 'localhost'. The *portmap* service needs to be running! */etc/init.d/portmap start*

Exercise 2: Arithmetic Service

You have to implement a server that provides the services of computing factorial (n) and power (a, b), where *n* is an integer, *a* and *b* are double.

1. Write the code for *arith.x.*
   You can appropriately modify the *msg.x* used above. For another example, see slides 37-49 in http://www.cse.iitb.ac.in/~sri/cs348/cs348-lec29-30-RPC-2012.pdf
2. Generate the stubs and templates of the client and server as in Exercise 1.
3. Modify arith_client.c and arith_server.c to implement your client and server.
4. Compile and execute the client and server as in Exercise 1. Try a few test cases.
5. Show a demo to one of the TAs. Ensure that the TA has noted your roll numbers, if you want to get full credit for this exercise!

Submission guidelines

Create a directory <rollnumber1>_<rollnumber2>_lab09. This should contain the following files:

1. lab09.txt
2. Your modified version of msg_server.c and msg_client.c
3. arith.x
4. Your modified version of arith_server.c and arith_client.c

Now tar it as follows:
tar -zcvf <rollnumber1>_<rollnumber2>_lab09.tgz <rollnumber1>_<rollnumber2>_lab09/

Submit the file <rollnumber1>_<rollnumber2>_lab09.tgz via moodle for grading.