

## RPC/RMI based Network Management Application

This client-server programming project is an actual Sun RPC or Java RMI. The network management tool of project 1 will be written using RPC or RMI. All requirements for the RPC/RMI project are the same as that of project 1. The intention of this project is to be able to compare the RPC/RMI paradigm of client-server development with the socket API based approach.

For RPC, you will need to provide 3 files:

netmgt.x     The RPC specification file

rnetmgt.c    The client program

netproc.c    The server procedures that are called from the client (in effect, called by the server stub)

For RMI, you will need to provide:

RMI interface class

Client source code/classes

Server source code/classes

Make your server concurrent by having it fork a new process/thread to handle each client request. Have your server print out diagnostic messages about what it is doing (e.g. forking a child process, accepting a new connection, etc.)

You will be required to provide a demo to the instructor. The client-server application will be tested for compliance with all requirements listed above. Your code will be expected to deal with invalid user commands. You will turn in binaries and source code of your client and server programs at the time of the demo.

### Paper

The paper must be well written and must have the sections listed below.

#### Abstract

Describe what your project is about; state its goals (i.e. comparison of distributed application development paradigms), experiments conducted etc. The abstract must be concise. Try not to exceed 150 words.

#### Introduction

First describe your project in greater depth, i.e. explain your goals, and include an explanation of the client-server set-up.

### Distributed Application Development Paradigms

You will describe both the paradigms in this section. Be sure to provide references when you refer to texts, papers, and websites etc. to describe these paradigms. Include a description of the major socket calls used (e.g. socket(), bind() etc) and the major RPC/RMI calls used.

### Results and Comparisons

First describe the studies carried out. Then provide the results by referring the reader to your graph(s). Explain your results. Then compare the socket API and SUN RPC technologies qualitatively. One metric by which to compare is how transparent RPC makes the underlying network by performing the data conversions, eliminating the socket calls etc.

### Conclusions

Provide your conclusions, insights that you have gained in this section. E.g. which technology shows better performance and why, which technology is easier to learn, which technology has a richer set of services to offer, which technology is better used in which circumstance.

### References

Use the following IEEE standard when listing references:

[1] Ahuja, Sanjay, P., Quintao, R., and Basavanahally, S., *An Empirical Analysis of the Performance of CORBA and Java RMI: Distributed Open Object Architectures for Internet Based Applications*, Proceedings of the 2001 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2001), Orlando, FL, July 2001.