CIS 620 Project 4 Spring 2020

(Due: Apr. 28)

Project Description

The purpose of this project is to write a distributed application using a Remote Procedure Call (RPC) protocol. The client program running on a Linux workstation provides the interface to users. The server running on another station provides two computational services. You are required to follow the eight steps of RPC distributed application implementation discussed in the class notes and then use the protocol translator rpcgen to implement the client/server programs.

The client program gets the server's name through the command line argument (i.e. `argv[1]`). Then, based on the next command line option (`-r` or `-u`), the client invokes the computation function `sumsqrt_range()` or `update_list()` on the server, gets the result back, and prints it out.

Two examples are shown below. The first example computes the sum of the square root from 2 to 6 (inclusive). The computation is executed on the machine degas. In the second example, the client calls the remote function `update_list()` which converts each value F in the linked list by using the formula $F * F/10.0$ and returns the updated linked list back to the client.

```
spirit> rcalc degas -r 2 6
9.831822

spirit> rcalc degas -u 5 10 13 14 25
2.5 10.0 16.9 19.6 62.5
```

The server provides two services (i.e. functions). The function `sumsqrt_range()` gets a range $N_1..N_2$ from the client via the function parameter and then calculates the sum of square roots (i.e. $\sqrt{N_1}+\sqrt{(N_1+1)}+\cdots+\sqrt{N_2}$). Similarly, the function `update_list()` gets a linked list of real numbers from the client and computes the values. Note that the server program should be run on another Linux workstation before the client starts.

Use the last 4-digit of your CSU-ID (e.g. `NNNN`) to be the RPC program number `0x2000NNNN` in the specification file `rcalc.x`. You can use the makefile in the `rdb` example as a template and modify it, such as changing the macro `APPN` value from `rdb` to `rcalc`, etc.

When the server is running, open a terminal session on the client. Issue the command `rpcinfo -p servername` to find the port number used by your RPC server. Take a screenshot and explain the response from rpcinfo in your report.

Turnin

Each group (at most two students) needs to submit this project using the following turnin command:

```
turnin -c cis620s -p proj4 makefile report.pdf rcalc.x rcalc.c rcalc_svc_proc.c
```

Don't forget that you have to turnin your report document `report.pdf` along with other source files. The document should include the description of your code, data structures, experiences in debugging/testing, screenshot/explanation, etc. The cover page should contain your name(s), picture(s), and the login-id you used to turnin.