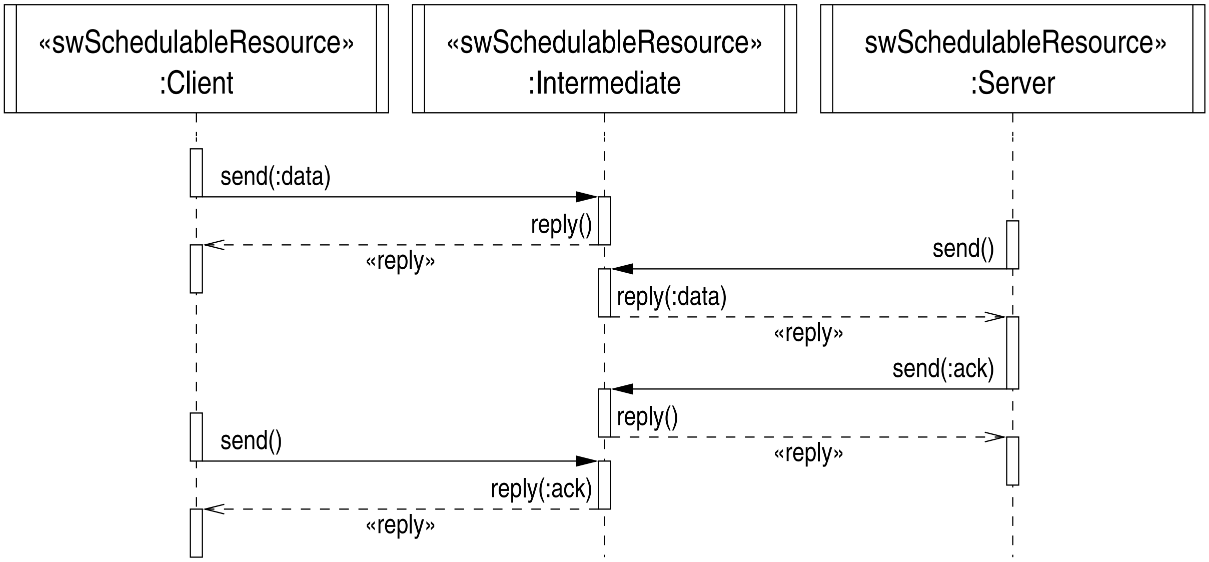
Carleton University
Department of Systems and Computer Engineering
SYSC 3303AC RealTime Concurrent Systems Winter 2024

Assignment 4 — Remote Procedure Calls

Background

The goal of this assignment is to change the three part system consisting of a client, an intermediate host, and a server from Assignment 2 and convert it to use synchronous communication to establish a two-way channel from the client to the server. In effect, you will now be using remote procedure calls (RPC) to transfer data from the client to the server using the intermediate host. The following sequence diagram shows how a basic write request takes place.



Specification

The algorithm is exactly the same as in the previous assignment. However, the Client now waits for the Intermediate task to accept the data and then reply rather than continuing on to wait for the acknowledgement packet directly. Also, note that the logic in the Server task is now reversed in that it now is making a request for data rather than waiting for data to arrive.

Hints

- Since you are using asynchronous UDP messages, you will need two for each RPC request, one to `send()` and another to `reply()`.
- You need to create an `rpc_send( :out, :in )` method to reuse in the client and the server. The first argument is used to pass the data to be sent, while the second is used to return the data received in a reply. This likely can't be done for the Intermediate task since the code between the `receive()` and `send()` is going to be different in general.
- You may wish to use more than one thread in the intermediate task, i.e., one for packets going from the Client to the Server, and one for packets going in the opposite direction.

- You may have to use more ports.
- The TAs will mark your assignments in the lab environment. It is your responsibility to ensure that your code works in that environment, and that any software required for viewing any text/diagrams is also present in the lab. You must use the IntelliJ IDE.

Questions

1. Why did I suggest that you use more than one thread for the implementation of the Intermediate task?
2. Is it necessary to use synchronized in the intermediate task? Explain.

Work Products

1. A “README.txt” file explaining the names of your files, set up instructions, and the answers to the two questions above.
2. One or more UML Class diagrams showing your system.
3. The source code for all three parts of the system, as well as any files required to run these files in IntelliJ. (You may submit test classes if you wrote them, but be sure to explain the files you have submitted – see #1 above.) Your code should demonstrate good programming style, and be well documented, etc.

For part 2, hand drawn scanned diagrams are acceptable, as long as they are neatly drawn and your handwriting is legible, and the software required to view them is present in the lab. As an alternative, you can use Violet UML.

Submitting Assignments

Assignments are to be submitted electronically using BrightSpace. Emailed submissions will not be accepted. See the course outline for the procedure to follow if illness causes you to miss the deadline.