Compiling and Running Java RMI Programs

Here we will use the Calculator example from the Java RMI tutorial. Note that the original example did not use any packages. First we need to make a package. Let us call the server package *cal.server*.

1. Create a directory structure like this in your home directory on one of the Vulcan machines:

   ```
   cs631
       homework2
               classes
               lib
   ```

   Change to the directory homework2 and download the Calculator example source files from the directory ~puri/cs631/source.tar.gz and extract it in the directory homework2 using the following command:

   `tar xfz ~puri/cs631/source.tar.gz` (you should be in the homework2 directory)

   Your directory structure should look like this after you extract the files:

   ```
   source
               cal
                       server
                               Calculator.java
                               CalculatorImpl.java
                               CalculatorServer.java
                       client
                               CalculatorClient.java
   ```

2. To compile the server source files, change to the directory homework2/source/cal/server and enter the following command (you may have change the directories in the classpath according to your installation):

   `export CLASSPATH=$HOME/cs631/homework2/classes` (for Bash shell users)
   OR
   `setenv CLASSPATH $HOME/cs631/homework2/classes` (for C shell users)

   `javac -d ../../../classes/ Calculator.java CalculatorImpl.java CalculatorServer.java`

   Note: You must have the directory structure created as shown above.

3. To create stubs, change to the directory homework2/classes and enter the following command:

   ```
   rmic cal.server.CalculatorImpl
   ```

4. Create a jar file of the interface definitions and stubs classes using the following command from the homework2/classes directory:

   ```
   jar cvf cal.jar cal/server/Calculator.class cal/server/CalculatorImpl_Stub.class
   ```

5. Copy the jar file cal.jar from the classes directory to the lib directory using the following command:

   ```
   cp cal.jar ../lib/
   ```

6. To compile the client source file, change to the directory homework2/source/cal/client and enter the following command:

   ```
   javac -d ../../../classes/ -classpath ../../../lib/cal.jar:. CalculatorClient.java
   ```

   Note: There is period after cal.jar:

7. To start the server, from the classes directory first start the rmiregistry (using rmiregistry <rmiport>) and then start the server using the following command:

   ```
   java -classpath . cal.server.CalculatorServer <rmiport>
   ```

   Note: Use the server hostname and one of the ports from the port range assigned in Homework-1 to start the rmiregistry and the server process.

8. In another window, start the client from the directory homework3/classes using the following command:

   ```
   java -classpath ../lib/cal.jar:. cal.client.CalculatorClient <servername> <rmiport>
   ```

   Note: There is period after cal.jar:. Use the client hostnames assigned in Homework-1 to start the client processes.