CPSC4780

Assignment 2

Due: March 4 (Friday)

The total is 25 marks.

Written questions (Due at the beginning of the class)

(1) (4 marks) If you design a RPC protocol generator, such as rpcgen, propose a solution to handle arrays and pointers to memory objects. Also discuss what kind of restrictions you need to let the programmer know about using your generator.

(2) (3 marks) If a RMI client needs to download the client stub, where is the client stub eventually downloaded to and why Java arranges it this way.

(3) (4 marks) Briefly compare copy-on-reference and copy-on-write and discuss under what circumstances each of them is more suitable and more efficient.

(4) (4 marks) Briefly describe how security issues are tackled in Java programs, including in Java applets and Java applications. You may want to unfold your discussions around Java's SecurityManager.

Programming question (Due at 11:59pm on March 4)

(5) (10 marks) Programming using RPC. You are required to create a RPC application. You can use the sample codes posted on the course website as a start point. We have discussed them in class already.

The task is described as follows. You are about to design and implement a BigIntCalculator server, which will perform big integer operations. Assume that the server only needs to perform Add (+), Subtraction (-), Multiplication (*), and division (/) between two big integers.

A client provides necessary arguments and calls the one of above four remote functions. After obtaining the result, the client displays it.

There are three parts of an operation: The first operand, the operator (+, -, *, /), and the second operand. The three parts *are separated* by spaces. The output from the client should use the format as follows:

```
Please input your operation: 123 * -100
   123
* -100
----------
-12300
```

```
Please input your operation: 123 - 100

   123
-  100
-------
    23

Please input your operation: 100-12

There should be three parts for an operation!
Please input your operation:
```

Your interface file should be called BIC.x, your server implementation file should be called BICServerImpl.c(pp), your client implementation file should be called BICClientImpl.c(pp).

E-mail those files to me as attachments and in *Subject* field, specify “Assignment 1, your last name”. If you have anything special discussions, you can write them in the body of the e-mail.