Lecture **Distributed Systems**
Fall 2010

Prof. Abraham Bernstein, Ph.D.

ASSIGNMENT 1 - SOCKETS & RPC (10 POINTS)

Due date: Oct. 26, 2010, 14:00 (CET)

Rules

- Assumed programming language is *Java* in version 1.6
- Code that is handed in and does not compile will NOT be graded. So please make sure to test your implementation properly.
- Assignments have to be solved individually. It is ok and also desired to discuss problems with peers, whereas copying code is not. As a result, plagiarism will lead to 0 points for the particular assignment for both parties.
- The due date is a hard deadline. E-mails that arrive after this deadline will be discarded and therefore the contained solution not graded.

All the above rules are final and no matter for further discussions!

Assume you have a website that allows students to sell their books when they don't use them anymore (like for example "semesterbooks.de").

The website uses a central server which stores in a database the available products and the information regarding the sellers.

The data on the server is stored in a text file in the following form:

<username>,<e-mail>,<ISBN>, <book title>, <author name>, <book price>

<username>,<e-mail>,<ISBN>, <book title>, <author name>, <book price>

...

Where the username and e-mail are the information of the student who wants to sell the book. For simplicity if a book has several authors only the first author is listed. The price is in CHF. Since the title can have commas should be put in "".

Find a sample data file for download under:

http://www.ifi.uzh.ch/ddis/fileadmin/teaching/FAll10/DistSystems10/assignments/A1_sample_data.txt

For simplicity we assumed the data is stored in a file, but if the students want they can use a MySQL DB with several tables (users, books, authors, etc.). The following requests are necessary in every-day life:

- displayBookByTitle(bookTitle)

This means the client sends a request to the server with the book name. What the client receives back is the book information: author name and book price.

- displayBookByAuthor(authorName)

The client sends a request to the server with the author name and it receives back the details regarding the book.

- displayBookByPrice(bookPrice)

The client sends a request to the server with the price and it receives back all the books which have the price <= bookPrice.

- getUserNames()

The client asks for the list of all usernames and it receives back their user names as well as their e-mail address.

- displayBookByUserName(userName)

The client asks for the books that a given user wants to sell. The result should be a list containing the book information : book title, author name and book price.

Now your tasks:

1) Implement a server/client system using Java Sockets and Threads that solves the above problem. You must not assume that a client necessarily asks for an existing author, book or user, so take care of appropriate error handling and feedback. The server system must be able to accept several client connections at a time (therefore the use of threading). For client-server communication you have to use the following protocol:

Socket parameters:
IP of the server and portNumber

Message format:
methodName [tab] arg1 [tab] arg2 .... argN

For example for method displayBookByTitle you have to use the following protocol:
displayBookByTitle title

(**5 Points**)

2) Implement the very same server/client system using Apache XML-RPC. (**3 Points**)

3) Create a short documentation in which you briefly describe your implementation, such that somebody who has not seen your code can understand what you did. (**2 Points**)

The whole documentation should not be longer than 1 page.

Grading:

Grading will be based on

a) the correctness of your code, i.e. does it solve the given task? and appropriate error handling.

b) readability/structure of your code (including appropriate comments).

c) clarity of your documentation, i.e. does it really describe what you implemented and how well can it be understood by somebody who has not written or read the code.

What to hand in and how:

- Create a zip file named <your_student_id>_<first name>_<last name>_assignment01.zip (e.g. 1234567_John_Doe_assignment01.zip). This zip file should contain two source code folders, one folder for the socket/thread part and one for the RPC part. It should also contain your documentation as *.pdf file.

In case your code requires any special treatment to compile, you have to enclose a README describing the necessary steps.

If you use MySQL please include the script used to create the DB and also a backup of your DB.

- Send this zip archive on time via email to Floarea Serban (*serban|at|ifi.uzh.ch*) or to Cosmin Basca (*basca|at|ifi.uzh.ch*). The email subject should start with ***[DS 2010]*** .

Helpful reading

Java Sockets

http://java.sun.com/docs/books/tutorial/networking/sockets/

Java Threads

http://java.sun.com/docs/books/tutorial/essential/concurrency/

Apache XML-RPC

http://ws.apache.org/xmlrpc/