Fall 2006: CS 431/631/731 Distributed Computing
Homework-3

200 points. Individual Work Only. Due November 6, 2006 before class.

Objective:
Design and implement a distributed file system using a multi-tier architecture with a Java Server Pages (JSP) based front-end.

Description:
The goal of this homework is to create logical names for physical files on different machines and develop a web-interface to access these files using the logical filenames. This is realized by using a multi-tier architecture. The JSP based front-end will provide a web-based user interface to perform the following operations on logical filenames:

1. Register a new host
2. Register a physical file (filename) residing on host (hostname) in the directory (dirname) to a logical filename (logicalfilename) (logicalfilename = hostname:/dirname/filename)
3. List registered files along with their details (hostname, directory, filename, filesize)
4. Download a registered file to the client
5. Replicate a registered file from one host to another host **(Graduate Students Only)**
6. Delete a registered file on a specific host

The logical filename along with the associated attributes are stored in a database in the middle-tier. Two separate tables will be used to keep track of these attributes. One for storing the various hostnames registered and one for storing the file attributes. You must use Java Database Connectivity (JDBC) API to connect to the database, to create tables, to insert rows, to update rows, and to delete rows. The list operation in the front-end requires that the contents of the file attributes table be displayed (along with the hostname instead of the hostid). The schema for these two tables along with some sample values is shown below:

Hostname Table Schema	
HOSTID INTEGER	HOSTNAME VARCHAR(64)
1	vulcan2.cis.uab.edu
2	vulcan3.cis.uab.edu

File Attributes Table Schema					
FILEID INTEGER	LOGICALNAME VARCHAR(64)	HOSTID INTEGER	DIRECTORY VARCHAR(256)	FILENAME VARCHAR(64)	FILESIZE INTEGER
1	hostsfile	1	/etc/	hosts	146
2	hostsfile	2	/etc/	hosts	188
3	myprofile	1	/mz/mb/	.bash_profile	456
4	mypicture	1	/mz/mb/puri/web/	puri.jpg	8452

Note that the same logical filename could be used to represent the same file on two different hosts (/etc/hosts on vulcan2 and vulcan3 in the example above). **Undergraduate students need not support the same logical filename assignment for the same file on two different hosts.**

Access to the files on a specific host must be provided through the Java RMI program developed in Homework-2. You must start the server on the host where the files will be accessed and use the client program to obtain the *filesize*. For other operations such as download or replication call the corresponding server methods from the client program (Note: The replicate operation (Graduate Students Only) involves first copying the file from *ServerA* to the *WebServer* and then copying the file from *WebServer* to *ServerB*).

Working Environment:
Use the CIS machines that are assigned to you to start the webserver, rmiregistry, and server program on the assigned ports. There is no need to be physically in the lab to use these machines, you can connect to these machines remotely using an SSH Client. When you are not in the department you have to setup a SSH tunnel for HTTP access to the webserver. Follow the directions given at: http://www.cis.uab.edu/cs633/SSHTunnelSetup.html to setup the tunnel (you have change the hostname titanic.hpcl.cis.uab.edu to the hostname assigned to you and also the port # 6060 to your assigned port for the webserver).

Here is the list of steps that you need to perform in order to get started with this homework:
1. Install Apache Tomcat Servlet Container
2. Setup the Java RMI Calculator server
3. Install the example web application with the Java RMI Calculator example
4. Test the Calculator example
5. Test PostgreSQL connectivity and execute simple JDBC example

Feedback Questions (answer to these questions has no impact on your grade):
Was this homework too difficult, or too easy?
Was the assignment fun or challenging?
Was there something that was unclear?
Was the homework too long for the given amount of time?
What did you learn from this homework?

Submission Instructions:
List ALL the references you used in this homework as well as test cases used to test your programs. This includes any classes that you used that you did not write and any help you received from any other sources. Use appropriate class name and include comments to indicate various operations performed by the program. Your program and other documents must have the following header information within comments:

```
/*
  Name:
  BlazerId:
  Homework #:
*/
```

Create a tar/zip file of the homework3 directory using the following filename format: blazerId-cs431-hw3.tar or blazerId-cs431-hw3.zip (graduate students use 631 or 731 instead of 431). The tar/zip file must include the source code, txt/word/pdf/ps document for the typed solution to short

answer questions and the feedback questions, instructions for executing the programs. Login to WebCT and go to the assignments section (Homework-3) and upload a single tar/zip file (using the browse button under attachments). There is no need to turn in any printed solutions for this homework, WebCT submission is sufficient.

Late Submissions:
Submissions must be made on the due date before the beginning of the class. Late submissions will lose 10% for every 24-hour period, up to a maximum of 50% (weekends and holidays count as one 24-hour period). Any submissions made after one-week will receive a score of 0 for this homework.

Resources:
1. Java Developer's Almanac - Code Samples:
   http://java.sun.com/developer/codesamples/examplets/
2. Java RMI Tutorial:
   http://java.sun.com/developer/onlineTraining/rmi/RMI.html
   http://java.sun.com/j2se/1.5.0/docs/guide/rmi/hello/hello-world.html
3. JSP Tutorial: http://www.jsptut.com/
4. Java Database Connectivity (JDBC) Basics:
   http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html