CS 455-555: Distributed Systems

Homework 2 (50 points)

Due Date –On Canvas–

Objective

Modify the single-threaded time server so that it limits the number of connections to be no more than once in an interval of **five** seconds. Use the provided client to test that it indeed limits the connection rate.

Setup

We will use our individual repository for hw and exams (this should have been done setup in the Homework 1). Switch to the class resources repository `CS455-resources` and please do a `git pull -rebase` to pull the files for the homework. They will be in the folder named `projects/hw2` in the `CS455-resources` repository.

Copy and commit the starter files for hw2 from the `projects/hw2` folder in the `CS455-resources` repository to the `hw2` folder in your hw-exams individual repository.

The server code is in the class `TimeServer.java` and this is the only class you will need to modify. The client code is in the class `TimeClient.java`, which you can use to test the server.

Implementation

Use a `Timer` from the `java.util` package. Check out its docs, in particular, for the `scheduleAtFixedrate` method. You will need to use the `TimerTask` from the `TimeServer` class so it can accept callbacks from the `Timer` thread.

The idea is that the server will accept one connection and then go into a *wait* until the timer goes off and *notifies* it to accept again. In order to implement this synchronization, use an object as a lock. See the code snippet below for an illustration on how to use a blank object as a lock.

```
Object lock = new Object(); // use this object's monitor as a lock
synchronized(lock) {
    lock.wait();
}
synchronized (lock) {
    lock.notify();
}
```

Your solution should not use `Thread.sleep()` anywhere!

Sample output

The following output from the server and clients shows what this may look like. Note the timestamps on the client output. **Are they what you would expect? Why or why not?**

```
[amit@localhost hw2(master)]$ java TimeServer
Received connect from 127.0.0.1: 57566
Timer expired!
Received connect from 127.0.0.1: 57568
Timer expired!
Received connect from 127.0.0.1: 57570
Timer expired!
Received connect from 127.0.0.1: 57572
Timer expired!
Received connect from 127.0.0.1: 57574
Timer expired!
Timer expired!
```

```
[amit@localhost hw2(master)]$ java TimeClient  localhost 5005
Time on host localhost is Tue Feb 23 22:08:18 MST 2016
[amit@localhost hw2(master)]$ java TimeClient  localhost 5005
Time on host localhost is Tue Feb 23 22:08:20 MST 2016
[amit@localhost hw2(master)]$ java TimeClient  localhost 5005
Time on host localhost is Tue Feb 23 22:08:25 MST 2016
[amit@localhost hw2(master)]$ java TimeClient  localhost 5005
Time on host localhost is Tue Feb 23 22:08:30 MST 2016
[amit@localhost hw2(master)]$ java TimeClient  localhost 5005
Time on host localhost is Tue Feb 23 22:08:35 MST 2016
[amit@localhost hw2(master)]$
```

Required Files

- Make sure to include a `README.md` file that follows the template provided. Not using the template will result in -10 points penalty.
- In your `README.md` file, the reflection section should answer the question asked above in the sample output section.

Submitting the Homework

Homework is individual so we will use your individual github classroom repository. To obtain credit, submit your homework through classroom github as described below.

Change to the directory called hw2 in your classroom individual repo and place all required files in that folder. Then do the following steps (on your master branch):

- git add [The appropriate files]
- git commit -m "Homework hw2 complete"
- git push origin master