Assignment 3

Due date

- Due by 11.59 PM on Thursday, March 27th.

Submit your code as per the provided instructions. A signup sheet will be provided to you during class to setup an appointment with the TA to provide a demo of your project.

Updates

- Tue Mar 25 18:08:51 EDT 2014: Grading guidelines have been posted

Assignment Goal

Apply the design principles and patterns you have learned so far to develop and test code for the given problem.

Team Work

- You are required to work in teams of two students on this project. CS442 students should change their project partner. You CANNOT collaborate or discuss the design, implementation, or debugging ideas with any other team.
- Both team members should be familiar with all aspects of the code. During the demo, the TA will randomly pick one of the team members to demo and answer questions. Failure to show familiarity with any part of the code will result in deduction of 50% of the assignment grade for both team members.

Programming Language

You are required to program this project in Java.

Compilation Method

- You are required to use ANT for the following:
  - Compiling the code
  - running the code
  - Generating a tarball for submission
  - Generating javadocs
- Your code should compile and run on *bingsuns* or the *debian-pods* in the Computer Science lab in the Engineering Building.

Policy on sharing of code

- EVERY line of code that you submit in this assignment should be written by your team or be part of the code template provided for this assignment. Do NOT show your code to any other group. Our code-comparison software can very easily detect similarities.

- Post to the listserv if you have any questions about the requirements. Do NOT post your code to the listserv asking for help with debugging. However, it is okay to post design/concept questions on programming in Java/C/C++.

Project Description

dWindows TaskManager

Develop a text based dashboard for a TaskManager. Check out the TaskManager on any Windows based OS for an example. There are three tabs on the dashboard for this assignment.

- *Processes*. It has the following information: ProcessName: value, UserName: value, CPU: value, Memory: value, Description: value
- *Performance*: It has the following information; Current Memory usage: value, current CPU usage: value. Total Physical memory: value, Total Cached: value.
- *Users*: It has the following information: userName: value, status: value. The options for status are: active, inactive, disconnected.

- For the information required to display in the dashboard, read synthetic data from a local file, one line at a time, process it, and update the display accordingly.
- Explain the format of the data in your README.txt file. You can choose any text-based format that you find easy to parse and process. The synthetic data should include data in different lines. Processing of each line should affect at least one tab of the dashboard.
- The format in each line may be different. For example, it is ok to have the first word of a line indicate which tab the data is for.
- Note that the tab for Total Memory and Total CPU should be updated whenever it is read from the input file.
- The Total Physical memory and Total cached memory, displayed in the *performance tab* are constant. Decide, and mention in the README.txt, how it is stored in the input file.
- Data from the local file should be read by a single thread. So, do NOT design for a multi-threaded application.
- The dashboard should show up-to-date information for all the information that a user wants to keep track of. It should write to stdout as it gets new information after reading a line of input. *However, the update of one tab (or information in a tab) need not automatically update other tabs.*
- The dashboard should also store all the information it receives. So. separate out the display module from the storage module.

FAQ

- How would the processes be impacted by a change from "Active" to "Disconnected"?. Answer: No change.
- If a user logs off, is it acceptable to terminate all of their > processes in addition to removing them from the list of users? Answer: It is acceptable, but not required.
- Should the information to be displayed stored? Answer: separate data structure(s) should be designed to store the information. The "display" should just display. So, if I add a GUI requirement, just the display module will change and nothing else.
- Are we to display all the Processes information, Performance information, and Users information every time

something is read from the file, and regardless as to whether they change as a result? For example, a user logs in, but this only affects the Users tab. Should we still display all three? Answer: No, just display the new information on stdout corresponding to the entry in the input file. Do not display everything.

- Use Logger in the following manner:
  - 0: No output should be printed. Only error messages should be printed (for example, message from a catch caluse before exiting).
  - 1: Only ouput from the dashboard should be printed
  - 2: Design on your own and mention in the README what is printed at this debug granularity
  - 3: Design on your own and mention in the README what is printed at this debug granularity
  - 4: Design on your own and mention in the README what is printed at this debug granularity

Design Requirements

- Same as Assignment-1

Code Organization

- Your directory structure should be the following:

```
-firstName_lastName_firstName_lastName
  ---taskManager
    ----- build.xml
    ----- README.txt
    ----- src
          ---taskManager
           ----------driver
           ----------util
           ----------other packages that you need
```

Code Templates

- None provided for this assignment.

Submission

- Same as Assignment-1. All team members should upload to blackboard.

Late Submissions

- The policy for late submissions is that you will lose 10% of the grade for each day that your submission is delayed.

Grading Guidelines

Grading Guidelines.

mgovinda at cs dot binghamton dot edu
Back to csx42: Design Patterns