

Assignment 1: Java Unsynchronized Threads File Manipulation

Complete the following activities.

1. Study the programs `OS_SimpleThread.java` and `OS_NThreadTest.java`; both are shown in the lecture handout.
2. Write two Java programs: `YOURLASTNAME_YOURFIRSTNAME_Driver.java` and `YOURLASTNAME_YOURFIRSTNAME_MyThread.java`, but substitute your last name for `YOURLASTNAME` and your first name for `YOURFIRSTNAME`.
3. `YOURLASTNAME_YOURFIRSTNAME_Driver.java` opens (no command line arguments) and reads from the file `in.txt`. The file `in.txt` will contain only two lines. The first line begins with a character 't' or 'T' followed by a positive integer number that is greater than or equal to 1. This number represents the number of threads the program `YOURLASTNAME_YOURFIRSTNAME_Driver.java` should create. The second line begins with a character 'f' or 'F' followed by the names of one or more file names. The number of file names listed is will be equivalent to the number of threads `YOURLASTNAME_YOURFIRSTNAME_Driver.java` will create. `YOURLASTNAME_YOURFIRSTNAME_Driver.java` must handle the following scenarios: (1) the file `in.txt` does not exist and (2) the file `in.txt` cannot be opened. Study the example file `in.txt` provided.
4. Once the driver program reads/processes the data from `in.txt`, it then creates several threads. Use the provided Java program `DriverTemplate.java` as a template to write `YOURLASTNAME_YOURFIRSTNAME_Driver.java`.
5. Each thread that executes `YOURLASTNAME_YOURFIRSTNAME_MyThread.java`, as created by the program `YOURLASTNAME_YOURFIRSTNAME_Driver.java`, opens (no command line arguments) one of the files specified on the second line of the file `in.txt`. The file opened by a thread is based on its order of creation; the first (id = 0) thread opens the first file, the second (id = 1) thread opens the second file, and so on. Each thread then reads a line from its file and writes the line read to an output file named `t#_out.txt`, where # represents the thread's numeric identifier. `YOURLASTNAME_YOURFIRSTNAME_MyThread.java` must handle the following scenarios: (1) the file `t#_in.txt` does not exist, (2) the file `t#_in.txt` cannot be opened, or (3) the file `t#_out.txt` cannot be opened. Use the provided Java program `MyThreadTemplate.java` as a template to write `YOURLASTNAME_YOURFIRSTNAME_MyThread.java`.

6. After all threads finish execution, the program YOURLASTNAME_YOURFIRSTNAME_Driver.java opens all output files t#_out.txt, reads a line from each file, and writes the line to an output file out.txt. To determine whether any of the children threads are alive, I recommend reading about the .isAlive() Java Thread method. Study the example files t0_out.txt and out.txt provided. YOURLASTNAME_YOURFIRSTNAME_Driver.java must handle the following scenarios: (1) t#_out.txt does not exist, (2) t#_out.txt cannot be opened, or (3) out.txt cannot be opened.
7. Study the diagram in the file CSCI4011-63100-171_A02_Java_Unsync_Threads_Diagram.pdf to understand better the unsynchronized ordering of events.
8. Submit the java files YOURLASTNAME_YOURFIRSTNAME_Driver.java and YOURLASTNAME_YOURFIRSTNAME_MyThread.java using Moodle before the specified due date and time.

If you have questions concerning this homework assignment, please ask your instructor for help. You may discuss the homework assignment with your classmates. However, the homework assignment you complete and turn in, for a grade, **must be your own work**.