**Submission Listener Delay:**

Code Block:

//Validates the Filename is correct

**if** (*validName*(fileName)) {

Thread.*sleep*(5000);

System.***out***.println("New File Detected: " + fileName);

System.***out***.println("Beginning new Thread...");

**new** AlgorithmRunner(submissionDirectory.getAbsolutePath(), fileName).start();

}//End if

The block of code above is what occurs once the listener detects a new submission. The program however is too fast and will execute this block of code WHILE the files are still be transferred to the location. This results in the thread being called when some/most of the files haven’t even arrived yet. To combat this I call Thread.sleep(5000) to have a 5 second delay. In the event that the assignments have significant file-sizes the transmission may take longer, this would result in the bug returning.

If we suspect the bug has returned, an easy test is to increase the amount the thread sleeps for significantly. If after that problems stop, then this is likely the problem.

A potential long-term fix would be to instead have the listener create a new listener on the submission-folder which would then track how long it has been since the last file was uploaded to the folder. Once that timer gets to some point, begin the thread. This should be fairly easy to change, however will have additional overhead. In addition this won’t solve the bug still, it will only be a BIGGER bandaid to it. If a single large file is being uploaded, the bug would likely return.

**Server Oddities:**

This less of a bug, and more of a couple caveats to running the code itself.

1. Once the server is started, in order to have it close itself a folder with the name “stop” must be uploaded. There is probably other more conventional ways to do it. This was just easy.
2. To start the server you do as follows:
   1. Navigate to the public\_html
   2. Navigate to 4F00\_MOPP
   3. Run: <java -cp ./bin SystemStarter>

Realistically, we can do <java -cp ./4F00\_MOPP/bin SystemStarter> instead from public\_html, which is more intuitive. However it would require a slight modification of the following:

File parentDirectory = **new** File(System.*getProperty*("user.dir")).getParentFile();

File submissionDirectory = **new** File(parentDirectory, "process");

We would need to add <+ ”4F00\_MOPP”> to the parentDirectory variable. This is omitted as it allows the code to be run both on Sandcastle as well as on my personal computer for when I’m debugging it. This is left as is purely for my convenience.