CSE 490h/CSE M552
Project 2
Due: 5pm, January 28, 2011

In this assignment, you are to add write-back cache coherence to your distributed storage system. The cache coherence algorithm should provide single copy (linearizable) semantics among a small number of clients communicating with a single server node. Files can be cached and modified locally at the client, so that repeated reads or writes by the same client to the same file should involve no server communication. When another client tries to read or write a modified file, appropriate messages need to be sent and delays inserted to preserve single copy semantics.

Failures should be handled as follows: failures can cause updates to be lost, but provided that some node has the latest copy of the file, your system should be able to continue after the failed node has rebooted, to continue to process read and write requests. This means being able to reconstruct the necessary state lost at the client or server. The subsequent assignment will provide support for safely handling the case where the latest update is lost.

The turn in instructions are the same as the previous assignment, with the following addition:

1) After writing the code for this assignment, but before testing your code, please draw a state machine diagram for the client and server, considering only a single memory location at a time. This state machine diagram will probably be similar to the one drawn in lecture, but may differ depending on the specifics of your implementation.

2) Run Synoptic on your code, and paste its output into the turning document. We'd like both the Synoptic run before you debugged your program, and afterward.

3) Please explain any difference between these three pictures, if any. Did Synoptic help you locate a bug, did it confirm your understanding of the behavior of your program, or something else?