CSE 551
Problem Set #1

Due: 4:30pm, Thursday, June 4, 2009

1. Write pseudo-code in Java, Python, C++ or C, to implement reader-writer locks (covered in class), with strict FIFO queueing (readers can enter if no earlier writer is waiting; writers can enter if no earlier reader or writer is waiting). All thread and synchronization operations should be via explicit library calls (fork, join, lock, unlock, wait, signal), and assuming Mesa semantics for condition variables. Please note: The code does not need to compile or run; we are interested in the algorithm, not the syntax. Please add enough comments so that the TA can understand your intent.

2. Write pseudo-code for a highly concurrent, multithreaded file buffer cache. Assume that the file system is to run on a system with dozens of CPUs, and so must be able to do many file operations in parallel. For simplicity, file operations are in terms of complete block reads and writes, readblock(x) and writeblock(x, block). On cache misses, the file system reads/writes blocks to disk, using diskread(x) and diskwrite(x, block); disk operations are synchronous, in that the calling thread blocks until the disk operation completes.

3. Add to your file buffer cache the ability to group operations atomically, by implementing the operations, xstart(x), xdone(x, commit/abort), and xrecover(), the latter called after a reboot.

4. A distributed checkpoint can be used to record a (causally) consistent state of a distributed computation. An example is for distributed garbage collection; purely local garbage collection will miss cycles of references that cross machine boundaries. Explain how you can use Lamport clocks to take a consistent checkpoint.

5. Explain how transient loops can occur with BGP.

6. The abstract of the following paper is misleading; explain why. Choffnes and Bustamante. Taming the Torrent: A Practical Approach to Reducing Cross-ISP traffic in P2P Systems. ACM SIGCOMM 2008.