2007 Paper 8 Question 1

Advanced Systems Topics

(a) Mutexes are usually built from *atomic* processor instructions. What does it mean for a processor instruction to be atomic, and how can this property be implemented in a cache-coherent multiprocessor system? [4 marks]

(b) Provide pseudocode for a simple multi-reader spinlock, including all four operations supported by this type of lock. Describe any atomic operations that your pseudocode uses. [8 marks]

(c) Why might the simple multi-reader spinlock scale poorly? Sketch a more scalable design assuming that read-only critical sections are vastly more frequent than critical sections that modify shared state. [4 marks]

(d) Is it possible to implement mutual exclusion in a multiprocessor system that provides only atomic load and store instructions? If so, why do modern processors provide read–modify–write instructions? [4 marks]