Problem Set 5, Part b

Due: Thursday, November 13, 2003

Reading:

Maurice Herlihy and Nir Shavit. *Introduction to Multiprocessor Synchronization and Concurrent Data Structures.* Chapters 3, 4. (In the course cabinet.)

Distributed Algorithms, Chapter 13 (skim).

Reading for next week:

Chapter 12.

Problems:

1. In chapter two we discussed time-stamp systems.

   (a) Prove, by way of a counter example, that the sequential time-stamp system T^3 in Chapter 2, started in a valid state (with no cycles among the labels), will not work for three threads in the concurrent case. Note that it is not a problem to have two identical labels since one can break such ties using thread ids. The counter example should thus bring about a situation where in some state of the execution three labels are not totally ordered.

   (b) The sequential time-stamp system in Chapter 2 had a range of 3^n different possible label values. Design a sequential time-stamp system that requires only $n2^n$ labels. Note that in a time-stamp system, one may look at all the labels in order to choose a new label, yet once a label is chosen, it should be comparable to any other label without knowing what the other labels in the system are. Hint: think of the labels in terms of their bit representation.

2. The following are the definitions of regular and atomic registers which we will elaborate upon in today's lecture:

   A register is called *Regular* if it has the following properties:

   - Every *READ* command, which is not concurrent with a *WRITE* command, obtains the last written value.
   - Every *READ* command which is concurrent with a *WRITE* command obtains either the old or the new value.

   An *Atomic Register* is :

   - *Regular*.
   - *Linearizable* to a sequential register.

   Given these definitions, what is the weakest type of register (regular or atomic) that can be used for for implementing each of the variables `level[i]` and `victim[l]` in the Filter algorithm (shown in class)? What about the `flag[i]` and `label[i]` in the Bakery Algorithm (shown in class)? Which (if any) of the two algorithms' properties (mutual-exclusion, no-deadlock, no-lockout) would not hold if all the registers were regular? Prove all your answers.

3. (a) Implement an m-writer p-reader read/write atomic register using $(m+p)^2$ single-reader/single-writer registers of unbounded size. Give a convincing argument for why the register is atomic (no more than half a page).

   (b) Modify the implementation in (a) to only provide sequential consistency, and to be more efficient than the atomic version. (No proofs are required.)

4. Exercise 13.17