The Mutual Exclusion Problem Has Been Solved

Leslie Lamport
Digital Equipment Corporation

14 September 1990

> A common assumption underlying mutual exclusion algorithms in shared memory systems is that:
>
> b. A memory reference to an individual word is mutually exclusive.

So reads question 28 and its suggested answer in the recent Self-Assessment Procedure on concurrency [8]. A mutual exclusion algorithm is needed to ensure that two concurrently executing processes do not access a shared resource at the same time. In a shared memory system, each word of memory is a shared resource. Assumption b states that two concurrently executing processes do not access this shared resource at the same time, which means that the memory system must implement its own mutual exclusion algorithm. Although a multiprocess algorithm that makes this assumption may be quite useful, it is not a solution to the mutual exclusion problem—the problem of achieving mutual exclusion in an asynchronous multiprocessor. (We would not say that an algorithm solves the $P = NP$ problem if it assumes a primitive operation that computes an NP-complete function in polynomial time.)

Is the Self-Assessment Procedure's suggested answer correct? That depends on how we interpret the word "common". The Random House Dictionary [4] gives two relevant definitions:

1. belonging equally to, or shared alike by, two or more or all in question: common property; common interests.

5. of frequent occurrence; usual; familiar: a common event; a common mistake.

In other words, the "common assumption" is either a universal one underlying all mutual exclusion algorithms (definition 1), or merely a popular one underlying many of them (definition 5).

The suggested answer is certainly correct with the second definition. Most mutual exclusion algorithms do assume mutually exclusive access to memory—a reasonable assumption for a practical algorithm, since it is satisfied by most multiprocessor memory systems. But what if definition 1 is meant? In that case, the answer asserts that the mutual exclusion problem has not been solved, since all mutual exclusion algorithms assume that the memory system implements mutual exclusion.

To find out which meaning was intended, we must turn to the references given with the answer. They all imply that mutually exclusive access to memory is required for a solution [1, page 8], [2, page 88], [3, page 276], [7, page 40]. Therefore, the first meaning of "common" was intended. The answer to question 28 of the Self-Assessment Procedure asserts that the mutual exclusion problem has not been solved.

This assertion is incorrect. If mutual exclusion had to depend on lower-level mutual exclusion, how could any shared memory provide mutually exclusive access in an asynchronous multiprocessor? The mutual exclusion problem was solved, without assuming lower-level mutual exclusion, by the bakery algorithm [5]. Further solutions, and a more thorough discussion of the problem, can be found in [6]. One might hope that by now, sixteen years after the bakery algorithm's publication, it would be common knowledge that this classic concurrency problem has been solved. Apparently, it is not.

It is insufficiently considered that men more often require to be reminded than informed.

Samuel Johnson

References

[1] M. Ben-Ari. *Principles of Concurrent Programming.* Prentice/Hall International, Englewood Cliffs, New Jersey, 1982.

[2] Per Brinch Hansen. *Operating System Principles.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[3] Raphael A. Finkel. *An Operating Systems Vade Mecum.* Prentice Hall, Englewood Cliffs, New Jersey, second edition, 1988.

[4] Stuart Berg Flexner, editor. *The Random House Dictionary of the English Language.* Random House, New York, second edition, 1987.

[5] Leslie Lamport. A new solution of Dijkstra's concurrent programming problem. *Communications of the ACM*, 17(8):453–455, August 1974.

[6] Leslie Lamport. The mutual exclusion problem—parts i and ii. *Journal of the ACM*, 33(2):313–348, January 1985.

[7] M. Raynal. *Algorithms for Mutual Exclusion.* MIT Press, Cambridge, Massachusetts, 1986.

[8] Brian A. Rudolph. Self-assessment procedure xxi. *Communications of the ACM*, 33(5):563–575, May 1990.