Processes versus Threads

Traditionally, the term *process* was used to mean a part of an algorithm or program that executes a sequence of operations. Among today's programmers, what used to be called a process is now called a thread, and a process is a collection of one or more threads together with a region of shared memory that only those threads can access.

Shared memory is a concept that is meaningful only for programs written in certain programming languages. It is meaningless for specifications, including specifications of algorithms. Is the variable b of the one-bit clock specification part of shared memory? This is a meaningless question. It is like asking: Is the variable b purple?

Since we are concerned with specification and algorithms, not with programs written in certain programming languages, I take *process* to have its traditional meaning.

CLOSE