

- Why should locks be held for as short a time as possible?

Locks completely stop other threads from going through them until the thread that is currently within the lock is released. So if one thread locks for a long time, another will come in after that and it will continue on until all threads have completed their processes. So we want the locks to be held for only short periods of time, so the program doesn't take a really long time to individually complete every thread.

- With a reader/writer lock, why can the single writing thread be the only thread in the critical section

Since changing values inside a lock can result in changes throughout a program, a reader/writer lock allows multiple threads within a section of code, however only if they are readers. Then other threads can still utilize the information, and won't change other stuff in the code that threads were meant to change, causing problems for other threads later. By allowing only one writer the code is only changed by one thread at a time avoiding these latent issues.

- What's a recursive mutex useful for?

Normally a mutex can't be acquired more than once in a row, so if in one function a mutex is acquired but then another function acquires the mutex this is the job for the recursive mutex.

- Looking at the book, explain in human terms the operation of Test-and-Set.

The test and set method is a simple operation to both determine the value of a pointer, the test part, and then update that value, the set, finally returning the old value.

- Looking at the book, explain in human terms how Test-and-Set can be used to create a spin lock.

To create a spin lock we can imagine a single thread, t1, to start that calls the lock function where no other thread has the lock, so the flag should be zero for t1. Then when the thread uses a while loop to check that TestAndSet's return value is equal to 1, both the lock's flag with the value of 0 and the new value 1 will be sent into the function, so our function will then replace the flag with 1 instead of 0 and still return 0. Thus t1 will not be caught spinning in the loop, which does nothing but wait. The spin lock is the simplest lock to build and simply spins using CPU cycles till the lock becomes available.