IEEE Standards Interpretations for IEEE Std 1003.1c™-1995 IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(R)) - System Application Program Interface (API) Amendment 2: Threads Extension (C Language)

Interpretation Request #34
Topic: pthread_mutex_init, pthread_cond_init **Relevant Clauses:** 11.3.2.2,11.4.2.2

The POSIX synchronization functions in section 11 were intended to allow use of efficient hardware-based synchronization mechanisms, and thus, it is essential that calling code adhere to certain rules. One important rule is that code cannot COPY synchronization objects, as the copy may not behave the same as the original. The description of Semaphore initialization specifically states that "Only sem itself may be used for performning synchronization.

The result of referring to copies of sem in calls to sem_wait, sem_trywait, sem_post, and sem_destroy, is undefined." I believe that the intent of the working group would be best served if a similar restriction had been placed upon code using the pthread_mutex_t and pthread_cond_t synchronization objects, but the required wording is missing from the standard. Code that currently does not violate the standard may thus fail to port to implementations of the standard that the working group did not intend to exclude.

POSIX 1003.1n should amend the standard to require that conforming applications shall not copy mutex and condition variable synchronization objects, just as is currently required of code using POSIX semaphores. REF: page 242, section 11.2.1.2, Semaphore Functions Description REF: page 255, section 11.3.2.2, Mutex initialization REF: page 261, section 11.4.2.2, Condition variable initialization

Interpretation Response
The standard is clear that it does not restrict the copying of mutex objects and conforming implementations must allow copies. However, the interpretations committee believes

that this is not the intent of the working and balloting groups. Rather, the intent was to be the same as for semaphores with the effect of using copies being "undefined".

Rationale for Interpretation

Implementations may wish to use hardware support to accelerate the performance of the mutex. For this to work, only the 'initial' copy would be the hardware and thus shared.