IEEE Standards Interpretations for IEEE Std 1003.1™-2001 IEEE Standard for Information Technology - Portable Operating System Interface (POSIX®)

Interpretation Request #28

Topic: pthread_cond_wait deadlock situations **Relevant Sections:** XSH pthread_cond_wait

It is unclear what the expected behavior is in case of a recursive-type mutex which would be locked several times. In this case, there could be several interpretations: -> The internal counter of the mutex is decreased once, as if pthread_mutex_unlock had been called. In this case, the mutex is not released as requested line 33271. -> The internal counter of the mutex is decreased to zero, as if pthread_mutex_unlock had been called enougth times to release the mutex. -> It is illegal to call this function with a recursive mutex locked more than once.

According to which is the correct behavior (see below), here are the actions to take: -> line 33271 should be like: "These function atomically release mutex (the effects being the same as if pthread_mutex_unlock had been called) and cause..." -> In this case a new paragraph should be inserted after the line 33276: "In case of a mutex of type PTHREAD_MUTEX_RECURSIVE, if the mutex has been locked more than once by the calling thread, the mutex is released as if pthread_mutex_unlock was called enought times. When the thread will leave the pthread_cond_wait or pthread_cond_timedwait function, the mutex shall be re-acquired as if pthread_mutex_lock was called the same number of times as pthread_mutex_unlock was called." -> this should be added after line 33270: "In case of a mutex of type PTHREAD_MUTEX_RECURSIVE, if the mutex is locked several times, the function shall return an error."

Interpretation Response

The standard does not speak to this issue, and as such no conformance distinction can be made between alternative implementations based on this. This permits implementations to choose how to handle situations where an application might deadlock, some

implementations might choose to detect and report that deadlock while others wouldn't.

Rationale for Interpretation
None.