

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

Copyright © 2006 by the Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue New York, New York 10016-5997 USA All Rights Reserved.

Interpretations are issued to explain and clarify the intent of a standard and **do not** constitute an alteration to the original standard. In addition, interpretations are not intended to supply consulting information. Permission is hereby granted to download and print one copy of this document. Individuals seeking permission to reproduce and/or distribute this document in its entirety or portions of this document must contact the IEEE Standards Department for the appropriate license. Use of the information contained in this document is at your own risk.

IEEE Standards Department Copyrights and Permissions 445 Hoes Lane, Piscataway, New Jersey 08855-1331, USA

### Interpretation Request #52

**Topic:** pthread\_mutex\_setprioceiling() EDEADLK **Relevant Sections:** XSH pthread\_mutex\_setprioceiling() Page: 1096 Line: 34688

The POSIX.1 specification for pthread\_mutex\_setprioceiling() states:

“The pthread\_mutex\_setprioceiling( ) function shall either lock the mutex if it is unlocked, or block until it can successfully lock the mutex, then it shall change the mutex's priority ceiling and release the mutex.

The problem is that it is not clear what happens if mutex is already locked by the calling thread. POSIX.1 states that calling pthread\_mutex\_lock() is an error if the mutex is already locked by the calling thread, and specifies an error condition for that situation (EDEADLK). But it doesn't say anything for pthread\_mutex\_setprioceiling(); no error condition is specified for this case.

Furthermore, some ceiling-change protocols require the mutex to be locked before the pthread\_mutex\_setprioceiling() operation can be invoked, to atomically change the ceiling and some of the state protected by the mutex.

Add the following sentence at the end of the text quoted above:

“If the mutex is already locked by the calling thread, the function shall change the mutex's priority ceiling and leave the mutex locked.”

Alternatively, if there are backwards compatibility issues that preclude this solution, an alternate solution would be to define a new interface for dynamically changing the mu-

tex's ceiling while the calling thread holds the lock on the mutex.

**Interpretation Response #52**

The standards states the requirements for `fgetc()`, `fgetwc()`, `fgets()`, `fgetws()`, and `gets()`, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

**Rationale for Interpretation**

None.