

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)

Copyright © 1996 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 #30

Topic: Thread Scheduling Functions, Description **Relevant Clauses:** 13.5.1.2

The standard currently requires that the `pthread_attr_setschedparam` and `pthread_attr_getschedparam` functions be present if `{_POSIX_THREADS}` is defined, while all other thread scheduling attribute functions are present if `{_POSIX_THREAD_PRIORITY_SCHEDULING}` is defined. It is pointless to have `pthread_attr_setschedparam` without also having `pthread_attr_setpolicy` and `pthread_attr_setinheritsched`, for example. The rationale for this section (B.13.1) makes clear the original intent of the working group -- "In this standard, the basic thread scheduling functions are defined under the `{_POSIX_THREADS}` option so that they are required of all threads implementations. However, there are no specific scheduling policies required by this option to allow for conforming thread implementations that are not targeted to realtime applications." Although it is understood that rationale (however clearly following the intent of the working group) is not normative or binding, it is not reasonable that all thread implementations support "scheduling parameters" but not scheduling policy, or the other mechanism that gives scheduling parameters meaning. The standard must be repaired to, at least, restore consistency. While the best policy would be to follow the intent of the working group, and make all of the scheduling-related functions available if `{_POSIX_THREADS}`, it would be acceptable to make them all available only if `{_POSIX_THREAD_PRIORITY_SCHEDULING}`. REF: pages 300-301, section 13.5.1.2, lines 509, 552 page 303, section 13.5.2.2, line 599 page 548, section B.13.1, lines 7897-7901

Interpretation Response

The standard is clear that on p. 300, line 509, `_POSIX_THREAD_PRIORITY_SCHEDULING` is defined then the functions: `pthread_attr_setscope()`, `pthread_attr_getscope()`,

`pthread_attr_setinheritsched()`, `pthread_attr_getinheritsched()`, `pthread_attr_setschedpolicy()`, `pthread_attr_getschedpolicy()` shall be provided. It is also clear on page 301, line 552, that if `_POSIX_THREADS` is defined then the functions: `pthread_attr_setschedparam()` and `pthread_attr_getschedparam()` shall be provided. Conforming implementations must conform to these.

Rationale for Interpretation

None.