

## 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 #33

**Topic:** Thread Cancellation, Async-Cancel Safety **Relevant Clauses:** 18.1.4

`pthread_cancel` is specified to be async-cancel safe, along with the `pthread_setcancelstate` and `pthread_setcanceltype` functions. I see no reasonable justification for allowing cancellation while async cancellation is enabled, nor does the rationale attempt to give any justification.

I believe that `pthread_cancel` should be removed from the list of async-cancel safe functions. Async cancellation is an unusual “aberration”, and I see no reason to allow use of any POSIX standard functions with async cancellation type except the functions required to disable cancellation.

### Interpretation Response

The standard is clear that `pthread_cancel` shall be async-cancel safe, and conforming implementations must conform to this.

### Rationale for Interpretation

Requiring `pthread_cancel()` to be async-cancel safe potentially makes it easier for helper threads running with asynchronous cancellation enabled to themselves cancel other threads. This imposes no undue burden on implementations since, if necessary, the `pthread_cancel()` implementation can always internally set the cancellation state to defer cancellation while `pthread_cancel()` is being executed and then restore the cancellation state after the cancel request has been executed.