

## 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. 345 East 47th Street New York, New York 10017 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 #21

**Topic:** timers and SIGEV\_THREAD **Relevant Clauses:** 14.2.2.2 (P1003.1b)

In P1003.1c D10, corresponding section in IEEE Std 1003.1b-1993 is 14.2.2.2 The behavior for timer\_create() specifies what happens when a sigevent structure of type SIGEV\_NONE or type SIGEV\_SIGNAL is passed to the function. POSIX.1c does not specify what the behavior of this function is if the sigevent structure is for SIGEV\_THREAD. The most complicated part is when sigevent specifies SIGEV\_THREAD but the timer is a reloading timer so that it continually expires. What is supposed to happen with timers and SIGEV\_THREAD?

A thread is to be created when the timer expires? What happens with reloading timers which continually expire? Does only one thread get created and from then on an overrun count is incremented? Since these threads are detached, you have no way of knowing this thread terminates in order to stop incrementing the count and create a new thread on the next timer expiration.

### Interpretation Response

This is a duplicate. See Interpretation #3, part 13.

### Rationale for Interpretation

None.