

## IEEE Standards Interpretations for IEEE Std 1003.1™-2001 IEEE 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 #30

**Topic:** sigpause and use in threaded applications **Relevant Sections:** XSH sigpause

There are some problems with how sigpause() is specified with respect to threads. The main issue is that the sighold() page (on which sigpause() is described) says "Use of any of these functions is unspecified in a multi-threaded process", and yet sigpause() is included in the list in 2.9.5.2 Cancellation Points of functions for which a cancellation point shall occur. If the use of sigpause() in a multi-threaded process is unspecified, there is no point in requiring that it contains a cancellation point. Under the DESCRIPTION heading on the sighold() page it says, "The sigpause() function shall remove sig from the calling process' signal mask and suspend the calling process until a signal is received". Since it uses "process" here instead of "thread", this is consistent with the use of sigpause() in multi-threaded processes being unspecified. However, later under the RETURN VALUE heading it uses "thread" instead of "process": "The sigpause() function shall suspend execution of the thread until a signal is received".

Either update the sighold() page appropriately so that the behaviour of sigpause() in multi-threaded processes is specified, or change "thread" to "process" under the RETURN VALUE heading and move sigpause() from the "shall" list to the "may" list in 2.9.5.2 Cancellation Points.

### Interpretation Response

The standard is unclear on this issue, and no conformance distinction can be made between alternative implementations in their use of sigpause in thread applications based on this. This is being referred to the sponsor.

### Rationale for Interpretation

None.

**Notes to the Editor (not part of this interpretation)**

In the next revision we should either deprecate these functions or define them to be thread-safe. This rdvk should be copied to SD/5. Background notes on sigpause and related functions and their modern analogs: Each of these historic functions has a direct analog in the modern POSIX functions which are required to be per-thread and thread-safe (aside from sigprocmask, which is replaced by pthread\_sigmask). They thus could be required to be thread-safe too. sigset is a simple wrapper for sigaction. sighold is sigprocmask/pthread\_sigmask with SIG\_BLOCK. sigignore is sigaction with SIG\_IGN. sigpause is sigsuspend. sigrelse is sigprocmask/pthread\_sigmask with SIG\_UNBLOCK.