

## **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 #71**

**Topic:** read, thread blocked on O\_NONBLOCK **Relevant Sections:** XSH read Page: 1190 Line: 37207

If a thread is blocked while attempting to read from a file descriptor that was opened with the O\_NONBLOCK bit clear, can that thread be unblocked by another thread setting the O\_NONBLOCK bit via `fcntl()`? Is an implementation that unblocks a previously blocked thread under these conditions conforming?

Issue an interpretation: "The standard clearly states that if, on entry to read, the thread finds O\_NONBLOCK is not set (i.e. is clear), the thread shall block until a very specific event happens (data becomes available). It does not include the event that some other thread changed the O\_NONBLOCK flag, and so the blocked thread should not wake up. Conforming implementations must conform to this."

### **Interpretation Response #71**

The standard does not speak to this issue, and as such no conformance distinction can be made between alternative implementations based on this.

### **Rationale for Interpretation**

None.