

## **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.

These are interpretations of IEEE Std 1003.1c-1995.

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 #6**

**Topic:** 6.7.1.1 **Relevant Clauses:** `_POSIX_PRIORITIZED_IO`

The first sentence of the fifth paragraph of 6.7.1.1 says "If `{_POSIX_PRIORITIZED_IO}` and `{_POSIX_PRIORITY_SCHEDULING}` are defined, then asynchronous I/O is queued in priority order, with the priority of each asynchronous operation based on the current scheduling priority of the calling process." The statement is ambiguous when the calling process is multi-threaded. If a multi-threaded process initiates async I/O requests from threads of various priorities, what is the "priority of the requesting process" for each such request -- the priority of the requesting thread, or something else? This is ill-defined, as a multi-threaded process may have several priorities.

Assuming that the interpretation answers "the priority of the requesting thread" to the above question, I suggest that the word "process" in the sentence be changed to "thread". Another possibility is that it is the priority of the "initial" thread started when the process was created, but this creates problems for async I/O requests from other threads subsequently created with a higher priority, since they cannot utilize their higher priority to advantage.

### **Interpretation Response**

The standard is clear: the priority is that of the process as set by `sched_setparam` or other mechanisms. Even in the presence of threads this remains a "well-defined" term and value. A conforming implementation shall use that value in the calculation of the I/O priority.

**Rationale for Interpretation**

None.