

## IEEE Standards Interpretation for IEEE Std 1003.1™-1990 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)

Copyright © 2001 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 #87

**Topic:** struct sched\_param and sched\_setscheduler **Relevant Sections:** 13.1, 13.3.3.1

IEEE Std 1003.1-1996 contains a structure , struct sched\_param with one defined member sched\_priority. Consider the following code extract: #define \_POSIX\_C\_SOURCE 199506L void mycode() { struct sched\_param param1; int j; j = MINPRIO; /\* MINPRIO is a valid priority \*/ /\* Also assume below that pid and policy are valid below \*/ param1.sched\_priority = j; if (sched\_setscheduler(pid, policy, &param1) == -1) { /\* branch 1 \*/ } else /\* branch 2 \*/ }

Q1. Is an implementation conforming, that returns -1 on a call to sched\_setscheduler() (branch 1) due to the fact that it has implementation defined members of the struct sched\_param that are not initialized? Section 13.1 has a section of text that states: "Implementations may add extensions as permitted by 1.3.1.1, item(2). Adding extensions to this structure, which may change the behavior of the application with respect to this standard when those fields are uninitialized, also requires that the extension be enabled as required by 1.3.1.1"

Q2. Is this a requirement on the application or the implementation?

Q3. Is it possible to write a strictly conforming POSIX.1 application using the sched\_param structure?

Q1. Yes since a conforming implementation may have extended the sched\_param structure

Q2. The application writer is required to consult the system documentation and may have to add extra initialization code for the sched\_param structure.

Q3. Its not possible to write a strictly conforming POSIX.1 application that uses the sched\_param structure.

**Interpretation Response**

Q1: The standard is clear that this is not conforming. From 1.3.1.1, page 3 of 1003.1b-1993: The conformance document shall define an environment in which an application can be run with the behavior specified by the standard. In no case shall such an environment require modification of a Strictly Conforming POSIX.1 Application.

Q2: This is a requirement on the implementation.

Q3: Yes. The implementation must not allow semantics related to implementation-specific members of struct sched\_param to affect the behavior of a Strictly Conforming POSIX.1 Application.

**Rationale for Interpretation**

The standard way to provide a choice between a strictly conforming execution environment and one with extensions is to have the implementation ignore the extensions unless an implementation-defined feature test macro is #defined when the program is compiled. An implementation that returns -1 unless implementation-added structure members are initialized and that documents this as an error condition as specified in subclause 2.4 (page 28, lines 675-684 of POSIX.1b-1993) is not in conformance with the Standard. Such an implementation is also broken with respect to 1.3.1.1, as quoted in the Interpretation above, and broken with respect to the application developer's reasonable expectations. Forwarded to Interpretations group: 14 Apr 1998 Proposed Interpretation: 17 Jul 1998 Finalised: February 17 1999