

## **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 #70**

**Topic:** `rmdir` **Relevant Sections:** 5.5.2.2

5.5.2.2 states “The directory shall be removed only if it is an empty directory.” Reading the standard it seems ambiguous how a conforming implementation might behave if the “empty directory” did not contain the implementation’s nominal collection of dot and dot-dot entries. 2.4 seems to suggest that a directory must contain dot and dot-dot (or at least pretend as though those entries existed). It seems that an implementation may then return `ENOTDIR` if `rmdir()` is called with an “empty directory” that was missing either dot or dot-dot since that directory does not conform to the abstract notion of that implementations “directory”. In short, here are the questions -- Does 5.5.2.2 permit `rmdir()` to fail when path refers to an empty directory and the empty directory is missing either the dot or dot-dot entries? If the implementation considers this entry to not be a valid directory, may it permit the dot and/or dot-dot entries to be removed with `unlink()` prior to removal of the directory itself with `unlink()`? If the implementation returns `EPERM` when `unlink()` is used to remove directories, and the implementation returns `ENOTDIR` (or perhaps `EINVAL`) when `rmdir()` is invoked with such a malformed directory, how does one remove the directory using only the mechanisms specified in 1003.1?

### **Interpretation Response**

`rmdir()` may fail when an empty directory is considered by the implementation to be malformed. The standard clearly states (5.5.1.2) that `unlink()` may only remove directories if the process has appropriate privileges and the implementation supports `unlink` on directories. The standard does not speak to the issue of recovering from such a malformed directory and such an issue is beyond the scope of POSIX.1.

### **Rationale for Interpretation**

The standard clearly states that implementations may return error values for reasons

other than those described in the standard if those conditions can be treated identically to the error conditions described, or may return additional error values.