

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 #1

Topic: Get User Name **Relevant Clauses:** 4.2.4, 4.7.2

Regarding to 4.2.4 (Get User Name) and 4.7.2 (Determine Terminal Device Name), IEEE Std 1003.1c-1995 is ambiguous when `getlogin_r()` and `ttyname_r()` cannot find the login or tty name, requiring that an error code is to be returned but not identifying a specific error code. Since the standard fails to specify a return code, applications cannot expect to portably handle this condition. Is this an oversight in the specification? I suggest that `ESRCH` should be the error returned if these functions cannot find a name.

Interpretation Response

The standard does not speak to this issue and as such no conformance distinction can be made between alternative implementations based on this. The standard is silent on what error shall be returned, and the interpretations committee believes that the silence is intentional. Strictly conforming implementations are not constrained in what they may return, and applications shall be ready to deal with varying behaviors.

Rationale for Interpretation

None.