

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 #19

Topic: static data - how volatile? **Relevant Sections:** 4.2.4.3

A number of interfaces (getlogin, getenv, ctermid, ttyname, getgid [sic, should be getgrgid], getgrnam, getpwuid, getpwnam) may be implemented to “point to static data [which], therefore, may be overwritten by each call.” A naive interpretation would be that this applies to subsequent calls to the same interface. Another interpretation is that the data returned is completely volatile, and must be copied immediately after it’s returned. Some existing implementations share the same static storage between different interfaces, so calling one interface can change the data pointed to by a pointer previously returned by a different interface. Do such implementations conform to the Standard?

Interpretation Response

No, such implementations do not conform. However, it would be conforming for functions described within the same subclause to use the same static area. getgrgid() and getgruid() may use the same area, and getpwuid() and getpwnam() may use the same area, but getgrgid() and getpwuid() may not use the same area.

Rationale for Interpretation

Each of the sections in question contains text similar to that in subclause 9.2.1.3 (Returns for getgrgid() and getgrnam()):

The return values may point to static data that is overwritten by each call.

This is clearly referring to the functions described in 9.2.1, and does not mean that the data can be overwritten by each call to any function. The fact that getenv() and getgrgid() (for example) each return a pointer to a static area that may be overwritten by a subsequent call to the same function cannot be interpreted to mean that they may over-

write each others' data.

The only ambiguity is in 9.2.1 and 9.2.2 (the `get gr*` and `getpw*` functions) where two functions are described together in one subclause. Does this mean that the data may be overwritten by the next call to either function, or the just next call to the same function? Must each function have its own static area? Since the standard does not clearly specify, it is conforming for the two to share a static area. (This is what was intended, since it is existing practice.)

Editorial note (not part of the interpretation)

Consider changing the existing wording to use similar wording to that in ANSI C, making it clear that the application shall not modify the returned data, and specify which functions may share the static area.