

IEEE Standards Interpretation for IEEE Std 1003.1™-2001 IEEE Standard Standard for Information Technology -- Portable Operating System Interface (POSIX®)

Copyright © 2006 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 #85

Topic: exit vs _exit **Relevant Sections:** XSH exit

The XSH page for exit() and _exit() says:

"The _Exit() [CX] and _exit() [/CX] functions shall not call functions registered with atexit() nor any registered signal handlers. Whether open streams are flushed or closed, or temporary files are removed is implementation-defined."

The statement about stream flushing is a change from the requirements for _exit() that were in POSIX.1-1996 and SUSv2, and appears to have occurred during the addition of text relating to _Exit() from C99. Hopefully it is unintentional, since a great many applications rely on _exit() not flushing stream buffers and this difference between exit() and _exit() is explained in pretty much every text book on UNIX/POSIX programming.

Also, the statements on this page about removal of temporary files conflict with the description of tmpfile() which states "The file shall be automatically deleted when all references to the file are closed." Thus the removal is not performed by the exit() function itself, although removal may be performed by the system during a call to the exit() function if there are any references to the file still open. One solution would be to remove all mention of temporary file removal from this page, thus leaving it to be specified by the tmpfile() page, but differences from the C Standard on the exit() page need to be CX shaded and how to you CX shade a deletion? Instead, the parts relating to temporary file removal should be modified to cover all methods by which the last reference can be closed, such as closure of a dup'ed file descriptor on process termination.

Action:

On line 10203 change: "Whether open streams are flushed or closed, or temporary files are removed is implementation-defined."

to: "[CX] Open streams shall not be flushed. Whether open streams are

closed (without flushing) is implementation-defined. If closing a stream results in the last reference to a file created by tmpfile() being closed, then the file shall be automatically deleted.[/CX]"

On line 10198 change: "The exit() function shall then flush all open streams with unwritten buffered data, close all open streams, and remove all files created by tmpfile()."

to: "The exit() function shall then flush all open streams with unwritten buffered data and close all open streams. [CX] If closing a stream results in the last reference to a file created by tmpfile() being closed, then the file shall be automatically deleted.[/CX]"

Append to line 10210:

"If closing a file descriptor results in the last reference to a file created by tmpfile() being closed, then the file shall be automatically deleted."

Interpretation Response

The standards states the requirements for exit(), and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

Rationale for Interpretation

None.