

## IEEE Standards Interpretations for IEEE Std 1003.1™-2001 IEEE 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 #2

**Topic:** interaction of file descriptors and I/O streams **Relevant Sections:** 2.5.1

Error This could be filed as an error in this section or an omission in `fflush()`. The problem is that in section 2.5.1 of XSH, which deals with the interaction of file descriptors and I/O streams, it is written: If the stream is open with a mode that allows reading and the underlying open file description refers to a device that is capable of seeking, the application shall either perform an `fflush( )`, or the stream shall be closed. The man page for `fflush()` explicitly does not mention what happens if `fflush()` is applied to an input stream or an update stream with the last operation being input. This wording comes straight from ISO C. The problem is that ISO C does not have to deal with file descriptors. POSIX does. In the early POSIX versions the text in section 2.5.1 was added to make clear how the interaction works. But the `fflush()` documentation was never updated.

There are two possible options: - the original intent was to define `fflush()` on input streams and it was just forgotten to update the `fflush()` documentation. In this case the page should be updated with the following in line 11607 (CX shaded): If `/stream/` points to an input stream or an update stream in which the most recent operation was not input, `fflush()` shall cause all buffered input to be discarded and on seekable files the file offset for the file descriptor be reset to the first unprocessed byte. - if `fflush()` is not meant or cannot be changed some other mechanism for synchronizing input streams must be found. As far as I can see no existing function fits the bill so I cannot make any concrete proposal.

### Interpretation Response

The standards states the requirements for `fflush()` and interaction between file descriptors in section 2.5.1 . However, concerns have been raised about this which are being

referred to the sponsor.

### **Rationale for Interpretation**

The standard developers noted that action in a future revision should be to restore the wording to that of the original 1003.1-1988 standard in `fflush()` and `fclose()`.

### **Notes to the Editor (not part of this interpretation)**

For a future revision: Add to `fflush()` as a new para at the end of DESCRIPTION: [CX shading on] For a stream open for reading, if the file is not already at EOF, and the file is one capable of seeking, the file offset of the underlying open file description shall be adjusted so that the next operation on the open file description deals with the byte after the last one read from, or written to, the stream being flushed. [CX shading off] Add to `fclose()` as a new third paragraph in DESCRIPTION [CX shading on] If the file is not already at EOF, and the file is one capable of seeking, the file offset of the underlying open file description shall be adjusted so that the next operation on the open file description deals with the byte after the last one read from or written to the stream being closed. [CX shading off]