

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

This interpretation does not necessitate any modification to assertions in IEEE Std 2003.1-1992. This is being referred to the sponsor for clarifying wording in the next amendment.

Interpretation Request #39

Topic: F_SETLKW and seek() **Relevant Sections:** not specified **Classification:** The unaddressed issue

Advisory locking is imprecise on what byte of a file is locked when using F_SETLKW and either SEEK_CUR or SEEK_END. Once blocked, the address could be that on entry, or established after the process is no longer blocked. From e-mail, discussions, it appears the address is calculated before the decision to block, and does not change.

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. This is being referred to the sponsor for clarifying wording in the next amendment.

Rationale for Interpretation

The standard doesn't specify the behavior in this case. It is clear, however, that the request must use one of the file size or seek pointers that was in effect while the fcntl() was being serviced. The email discussions about how existing implementations work is not relevant.