

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 #38

Topic: errors and fstat with file off_t **Relevant Sections:** 5.3.1.4 **Classification:** No change

Section 5.3.1.4 open() Errors - Assume off_t can take the values between -7 and 7. Assume also that an Ada binding has created a file 8 bytes big. Shall open() work or fail? If fail, with what error code? If work, what does fstat() return? (Paul Wanish IBM)

Interpretation Response

IEEE Std 1003.1-1990 does not specify error conditions for open() or for fstat() for the case where the file size cannot be represented in an object of type off_t. A conforming implementation could define such error conditions.

Unless such an error condition is defined by the implementation, open() must succeed in this example. The value in the st_size field of the structure returned by fstat() on the file descriptor returned by this open() is unspecified for the case where the actual value is larger than can be stored in an object of type off_t.

A call to fstat() must fail if it is impossible to represent the file size in a variable of type off_t. When this occurs, errno must be set to a non-zero value that is not one of the values specified for fstat() by POSIX.1.

Rationale for Interpretation

An implementation of fstat() might express the failure to set the st_size field by setting the values of the fields other than st_size in the structure pointed to by the buf argument to the proper values. fstat() could return -1 with errno set to a value defined by the implementation, and documented to mean that the file size exceeds the size that can be stored in type off_t. Such an implementation of fstat() could even store the true

file size in an implementation-defined member of the `stat` structure, of some appropriate type.

It is suggested that a future revision of IEEE Std 1003.1 specify the behavior of `stat()` and `fstat()` when used on files whose sizes cannot be represented in variables of type `off_t`. This condition can arise when file systems are mounted from a remote POSIX.1 system on which `off_t` is a larger type than on the local system.