IEEE Standards Interpretation for IEEE Std 1003.1™-1990 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)

Interpretation Request #76
Topic: read() **Relevant Sections:** 6.4.1.2

I would like a clarification on section 6.4.1 in IEEE Std 1003.1-1990, the System Interface Standard, describing the read() function. In the description in section 6.4.1.2, it is unclear exactly in what state the caller's read buffer is left after the function returns, in two specific situations:

1. If the read() function returns a value less than "nbyte" but greater than or equal to zero, indicating that the end-of-file was reached, must POSIX implementations guarantee that the remaining bytes in the buffer, from just after the last valid byte transferred to "nbyte"-1, be left unmodified by the read() function? Or is the remainder of the buffer left with undefined contents?

2. Similarly, if the read() function returns an error other than EINTR, must POSIX implementations guarantee that the caller's read buffer is left unmodified, or are the contents undefined? (It seems fairly clear from paragraph 7 that at least for EINTR the contents must be considered undefined, since the OS may return -1 anyway after data has been transferred.)

In the (unlikely I would think) case that this issue hasn't already been dealt with and resolved, my suggested correction would be to state explicitly that after a partial read, the contents of the remainder of the buffer is undefined, and after an error, the contents of the entire read buffer is undefined. In systems that implement read() using some IPC mechanism that may have alignment or size granularity restrictions, it is much more difficult (and in some cases impossible) to implement read() by receiving data from the IPC system directly into the caller's buffer without copying it, if the implementation must guarantee that no more bytes in the read buffer are modified than are actually success-

fully transferred.

Interpretation Response
For both of these issues, 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.

Rationale for Interpretation
Once the buffer has been passed to the system, the state of the buffer is undefined. Forwarded to Interpretations group: Apr 10 1996 Forwarded for review: Oct 22 1996 Finalized: Nov 24 1996