

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 #102

Topic: dd seek extend **Relevant Sections:** XCU dd

Should the command

```
dd if=/dev/null bs=1024 of=outfile seek=1
```

extend outfile so that it has size 1024 if it was previously smaller?

Commands like this are sometimes used to create a file with a specified size, but this technique is not currently portable.

The description of of=file says:

“If seek=expr is specified, but conv=notrunc is not, the effect of the copy shall be to preserve the blocks in the output file over which dd seeks, but no other portion of the output file shall be preserved. (If the size of the seek plus the size of the input file is less than the previous size of the output file, the output file shall be shortened by the copy.)”

It is not clear how “preserve the blocks [...] over which dd seeks” applies to the case where the seek offset is beyond EOF.

Some implementations extend the file and some don't. This is possibly just a side effect of how they implement the “shall be shortened” requirement, as there are two obvious methods of implementing it. One is to call ftruncate() (or some equivalent) to set the size to the seek offset before starting the copy (thus the writes done during the copy extend the file). The other is to overwrite existing data during the copy and then call ftrun-

cate() afterwards to shorten the file if some old data remains past the point where the copying finished. When the input file is empty, the former method leads to the output file size being changed whereas the latter does not.

Note that the issue only applies to seekable files. For non-seekable files the standard requires dd to fill the space between EOF and the specified offset with null bytes. (See the description of seek=n.)

The suggested change below would clarify that the output file is extended, to ensure the behaviour for seekable files is consistent with that for non-seekable files. An alternative would be to state explicitly that either behaviour is allowed (for seekable files).

Insert before the closing parenthesis on line 11770:

“If the input file is empty and either the size of the seek is greater than the previous size of the output file or the output file did not previously exist, the size of the output file shall be set to the size of the seek.”

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.

Rationale for Interpretation

None.