IEEE P1003.2 Draft 11.2 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

12623 ## 4.61 `tee` — Duplicate standard input

12624 ### 4.61.1 Synopsis

12625 `tee [-ai][`*file* `...]`

12626 ### 4.61.2 Description

12627 The `tee` utility shall copy standard input to standard output, making a copy in
12628 zero or more files. The `tee` utility shall not buffer output.

12629 The options determine if the specified files are overwritten or appended to.

12630 ### 4.61.3 Options

12631 The `tee` utility shall conform to the utility argument syntax guidelines described
12632 in 2.10.2.

12633 The following options shall be supported by the implementation:

12634 `-a` Append the output to the files rather than overwriting them.

12635 `-i` Ignore the SIGINT signal.

12636 ### 4.61.4 Operands

12637 The following operands shall be supported by the implementation:

12638 *file* A pathname of an output file. Implementations shall support pro-
12639 cessing of at least 13 *file* operands.

12640 ### 4.61.5 External Influences

12641 #### 4.61.5.1 Standard Input

12642 The standard input can be of any type.

12643 #### 4.61.5.2 Input Files

12644 None.

12645 #### 4.61.5.3 Environment Variables

12646 The following environment variables shall affect the execution of `tee`:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

12647 **LANG** This variable shall determine the locale to use for the
12648 locale categories when both **LC_ALL** and the correspond-
12649 ing environment variable (beginning with **LC_**) do not
12650 specify a locale. See 2.6.

12651 **LC_ALL** This variable shall determine the locale to be used to over-
12652 ride any values for locale categories specified by the set-
12653 tings of **LANG** or any environment variables beginning
12654 with **LC_**.

12655 **LC_CTYPE** This variable shall determine the locale for the interpreta-
12656 tion of sequences of bytes of text data as characters (e.g.,
12657 single- versus multibyte characters in arguments).

12658 **LC_MESSAGES** This variable shall determine the language in which mes-
12659 sages should be written.

12660 #### 4.61.5.4 Asynchronous Events

12661 Default, except that if the `-i` option was specified, SIGINT shall be ignored.

12662 ### 4.61.6 External Effects

12663 #### 4.61.6.1 Standard Output

12664 The standard output shall be a copy of the standard input.

12665 #### 4.61.6.2 Standard Error

12666 Used only for diagnostic messages.

12667 #### 4.61.6.3 Output Files

12668 If any *file* operands are specified, the standard input shall be copied to each
12669 named file.

12670 ### 4.61.7 Extended Description

12671 None.

12672 ### 4.61.8 Exit Status

12673 0 The standard input was successfully copied to all output files.

12674 >0 An error occurred.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

12675 ### 4.61.9 Consequences of Errors

12676 If a write to any successfully opened *file* operand fails, writes to other successfully
12677 opened *file* operands and standard output shall continue, but the exit status shall
12678 be nonzero. Otherwise, the default actions specified in 2.11.9 shall apply.

12679 ### 4.61.10 Rationale. *(This subclause is not a part of P1003.2)*

12680 **Examples, Usage**

12681 The `tee` utility is usually used in a pipeline, to make a copy of the output of some
12682 utility.

12683 The *file* operand is technically optional, but `tee` is no more useful than `cat` when
12684 none is specified.

12685 **History of Decisions Made**

12686 The buffering requirement means that `tee` is not allowed to use C Standard {7}
12687 fully-buffered or line-buffered writes, not that `tee` has to do one-byte reads fol-
12688 lowed by one-byte writes.

12689 It should be noted that early versions of BSD silently ignore any invalid options,
12690 and accept a single – as an alternative to `-i`. They also print the message

12691 `"tee: cannot access %s\n"`, <*pathname*>

12692 if unable to open a file.

12693 Historical implementations ignore write errors. This is explicitly not permitted
12694 by this standard.

12695 Some historical implementations use O_APPEND when providing append mode;
12696 others just *lseek* () to the end of file after opening the file without O_APPEND.
12697 This standard requires functionality equivalent to using O_APPEND ; see 2.9.1.4.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

12698 # 4.62 `test` — Evaluate expression

12699 ## 4.62.1 Synopsis

12700 `test [`*expression* `]`

12701 `[ [`*expression* `] ]`

12702 ## 4.62.2 Description

12703 The `test` utility shall evaluate the *expression* and indicate the result of the 1
12704 evaluation by its exit status. An exit status of zero indicates that the expression 1
12705 evaluated as true and an exit status of 1 indicates that the expression evaluated 1
12706 as false. 1

12707 In the second form of the utility, which uses `[ ]`, rather than `test`, the square
12708 brackets shall be separate arguments.

12709 ## 4.62.3 Options

12710 The `test` utility shall not recognize the `--` argument in the manner specified by
12711 utility syntax guideline 10 in 2.10.2.

12712 Implementations shall not support any options. 1

12713 ## 4.62.4 Operands

12714 All operators and elements of primaries shall be presented as separate arguments 2
12715 to the `test` utility.

12716 The following primaries can be used to construct *expression* :

12717 `–b` *file* True if *file* exists and is a block special file.

12718 `–c` *file* True if *file* exists and is a character special file.

12719 `–d` *file* True if *file* exists and is a directory.

12720 `–e` *file* True if *file* exists.

12721 `–f` *file* True if *file* exists and is a regular file.

12722 `–g` *file* True if *file* exists and its set group ID flag is set.

12723 `–n` *string* True if the length of *string* is nonzero.

12724 `–p` *file* True if *file* is a named pipe (FIFO).

12725 `–r` *file* True if *file* exists and is readable.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.