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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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]

Many early implementations of `diff` require seekable files. Since POSIX.1 {8} supports named pipes, the working group decided that such a restriction was unreasonable. Note also that the allowed file name – almost always refers to a pipe.

No directory search order is being specified in 4.17.6.1.1. The historical ordering is, in fact, not optimal, in that it prints out all of the differences at the current level, including the statements about all common subdirectories before recursing into those subdirectories.

The message

> `"diff %s %s %s\n"`, <*diff_options*> , <*filename1*> , <*filename2*>

does not vary by locale because it is the representation of a command, not an English sentence.

4.18 `dirname` — Return directory portion of pathname

4.18.1 Synopsis

`dirname` *string*

4.18.2 Description

The *string* operand shall be treated as a pathname, as defined in 2.2.2.102. The string *string* shall be converted to the name of the directory containing the filename corresponding to the last pathname component in *string* , performing actions equivalent to the following steps in order:

(1) If *string* is `//`, skip steps (2) through (5).

(2) If *string* consists entirely of slash characters, *string* shall be set to a single slash character. In this case, skip steps (3) through (8).

(3) If there are any trailing slash characters in *string* , they shall be removed.

(4) If there are no slash characters remaining in *string* , *string* shall be set to a single period character. In this case, skip steps (5) through (8).

(5) If there are any trailing nonslash characters in *string* , they shall be removed.

(6) If the remaining *string* is `//`, it is implementation defined whether steps (7) and (8) are skipped or processed.

(7) If there are any trailing slash characters in *string* , they shall be removed.

(8) If the remaining *string* is empty, *string* shall be set to a single slash character.

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

The resulting string shall be written to standard output.

4.18.3 Options

None.

4.18.4 Operands

The following operand shall be supported by the implementation:

string A string.

4.18.5 External Influences

4.18.5.1 Standard Input

None.

4.18.5.2 Input Files

None.

4.18.5.3 Environment Variables

The following environment variables shall affect the execution of `dirname`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments).

LC_MESSAGES This variable shall determine the language in which messages should be written.

4.18.5.4 Asynchronous Events

Default.

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

4827 ### 4.18.6 External Effects

4828 #### 4.18.6.1 Standard Output

4829 The `dirname` utility shall write a line to the standard output in the following for-
4830 mat:

4831 `"%s\n"`, *<resulting string>*

4832 #### 4.18.6.2 Standard Error

4833 Used only for diagnostic messages.

4834 #### 4.18.6.3 Output Files

4835 None.

4836 ### 4.18.7 Extended Description

4837 None.

4838 ### 4.18.8 Exit Status

4839 The `dirname` utility shall exit with one of the following values:

4840 0 Successful completion.

4841 >0 An error occurred.

4842 ### 4.18.9 Consequences of Errors

4843 Default.

4844 ### 4.18.10 Rationale. *(This subclause is not a part of P1003.2)*

4845 **Examples, Usage**

4846 The `dirname` utility originated in System III. It has evolved through the
4847 System V releases to a version that matches the requirements specified in this
4848 description in System V Release 3.

4849 4.3 BSD and earlier versions did not include `dirname`.

4850 Table 4-6 indicates the results required for some invocations of `dirname`.

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

Table 4-6 – `dirname` Examples

Command	Results
`dirname /`	`/`
`dirname //`	`/` or `//`
`dirname /a/b/`	`/a`
`dirname //a//b//`	`//a`
`dirname`	*unspecified*
`dirname a`	`. ($? = 0)`
`dirname ""`	`. ($? = 0)`
`dirname /a`	`/`
`dirname /a/b`	`/a`
`dirname a/b`	`a`

History of Decisions Made

The behaviors of `basename` and `dirname` in this standard have been coordinated so that when *string* is a valid pathname

```
$(basename "string")
```

would be a valid filename for the file in the directory

```
$(dirname "string")
```

This would not work for the versions of these utilities in earlier drafts due to the way processing of trailing slashes was specified. Consideration was given to leaving processing unspecified if there were trailing slashes, but this cannot be done; the POSIX.1 {8} definition of pathname allows trailing slashes. The `basename` and `dirname` utilities have to specify consistent handling for all valid pathnames.

Since the definition of *pathname* in 2.2.2.102 specifies implementation-defined behavior for pathnames starting with two slash characters, Draft 11 has been changed to specify similar implementation-defined behavior for the `basename` and `dirname` utilities. On implementations where the pathname `//` is always treated the same as the pathname `/`, the functionality required by Draft 10 meets all of the Draft 11 requirements.

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

4.19 `echo` — Write arguments to standard output

4.19.1 Synopsis

```
echo [string ...]
```

4.19.2 Description

The `echo` utility shall write its arguments to standard output, followed by a `<newline>` character. If there are no arguments, only the `<newline>` character shall be written.

4.19.3 Options

The `echo` utility shall not recognize the `--` argument in the manner specified by utility syntax guideline 10 in 2.10.2; `--` shall be recognized as a string operand.

Implementations need not support any options.

4.19.4 Operands

The following operands shall be supported by the implementation:

string A string to be written to standard output. If the first operand is `"-n"` or if any of the operands contain a backslash (\) character, the results are implementation defined.

4.19.5 External Influences

4.19.5.1 Standard Input

None.

4.19.5.2 Input Files

None.

4.19.5.3 Environment Variables

The following environment variables shall affect the execution of `echo`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

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