IEEE P1003.2 Draft 11.2 – September 1991

initial negative number argument specifies a signal number (rather than a process group) is the existing behavior, and was retained. Therefore, to send the default signal to a process group (say 123), an application should use a command similar to one of the following:

```
kill -TERM -123
kill -- -123
```

The `−s` option was added in response to international interest in providing some form of `kill` that meets the Utility Syntax Guidelines.

Some implementations provide `kill` only as a shell built-in utility and use that status to support the extension of killing background asynchronous lists (those started with &), by the use of job identifiers. For example,

```
kill %1
```

would `kill` the first asynchronous list in the background. This standard does not require (but permits) such an extension, because other related job-control features are not provided by the shell, and because these facilities are not ordinarily usable in portable shell applications. This notation is expected to be introduced by the UPE.

4.33 `ln` — Link files

4.33.1 Synopsis

`ln [−f]` *source_file target_file*

`ln [−f]` *source_file* . . . *target_dir*

4.33.2 Description

In the first synopsis form, the `ln` utility shall create a new directory entry (link) for the file specified by the *source_file* operand, at the *destination* path specified by the *target_file* operand. This first synopsis form shall be assumed when the final operand does not name an existing directory; if more than two operands are specified and the final is not an existing directory, an error shall result.

In the second synopsis form, the `ln` utility shall create a new directory entry for each file specified by a *source_file* operand, at a *destination* path in the existing directory named by *target_dir*.

If the last operand specifies an existing file of a type not specified by POSIX.1 {8}, the behavior is implementation defined.

The corresponding destination path for each *source_file* shall be the concatenation of the target directory pathname, a slash character, and the last pathname component of the *source_file*. The second synopsis form shall be assumed when the final operand names an existing directory.

For each *source_file* :

(1) If the *destination* path exists:

(a) If the `-f` option is not specified, `ln` shall write a diagnostic message to standard error, do nothing more with the current *source_file,* and go on to any remaining *source_files.*

(b) Actions shall be performed equivalent to the POSIX.1 {8} *unlink* () function, called using *destination* as the *path* argument. If this fails for any reason, `ln` shall write a diagnostic message to standard error, do nothing more with the current *source_file,* and go on to any remaining *source_files.*

(2) Actions shall be performed equivalent to the POSIX.1 {8} *link* () function using *source_file* as the *path1* argument, and the *destination* path as the *path2* argument.

4.33.3 Options

The `ln` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following option shall be supported by the implementation:

`-f` Force existing *destination* pathnames to be removed to allow the link.

4.33.4 Operands

The following operands shall be supported by the implementation:

source_file A pathname of a file to be linked. This can be a regular or special file; whether a directory can be linked is implementation defined.

target_file The pathname of the new directory entry to be created.

target_dir A pathname of an existing directory in which the new directory entries are to be created.

4.33.5 External Influences

4.33.5.1 Standard Input

None.

4.33.5.2 Input Files

None.

4.33.5.3 Environment Variables

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

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.

4.33.5.4 Asynchronous Events

Default.

4.33.6 External Effects

4.33.6.1 Standard Output

None.

4.33.6.2 Standard Error

Used only for diagnostic messages.

4.33.6.3 Output Files

None.

4.33.7 Extended Description

None.

4.33.8 Exit Status

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

0 All the specified files were linked successfully.

>0 An error occurred.

4.33.9 Consequences of Errors

Default.

4.33.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

None.

History of Decisions Made

Some historic versions of `ln` (including the one specified by the *SVID*) unlink the destination file, if it exists, by default. If the mode does not permit writing, these versions will prompt for confirmation before attempting the unlink. In these versions the `-f` option causes `ln` to not attempt to prompt for confirmation.

This allows `ln` to succeed in creating links when the target file already exists, even if the file itself is not writable (although the directory must be). Previous versions of this draft specified this functionality.

This draft does not allow the `ln` utility to unlink existing destination paths by default for the following reasons:

— The `ln` utility has traditionally been used to provide locking for shell applications, a usage that is incompatible with `ln` unlinking the destination path by default. There was no corresponding technical advantage to adding this functionality.

— This functionality gave `ln` the ability to destroy the link structure of files, which changes the historical behavior of `ln`.

— This functionality is easily replicated with a combination of `rm` and `ln`.

— It is not historical practice in many systems; BSD and BSD-derived systems do not support this behavior. Unfortunately, whichever behavior is selected can cause scripts written expecting the other behavior to fail.

— It is preferable that `ln` perform in the same manner as the *link*() function, which does not permit the target to already exist.

This standard retains the `−f` option to provide support for shell scripts depending on the *SVID* semantics. It seems likely that shell scripts would not be written to handle prompting by `ln`, and would therefore have specified the `−f` option.

It should also be noted that `−f` is an undocumented feature of many historical versions of the `ln` utility, allowing linking to directories. These versions will require modification.

Previous drafts of this standard also required an `−i` option, which behaved like the `−i` options in `cp` and `mv`, prompting for confirmation before unlinking existing files. This was not historical practice for the `ln` utility and has been deleted from this version.

Although symbolic links are not part of the standard, the `−s` option should be used only for the traditional purpose of creating symbolic links.

4.34 `locale` — Get locale-specific information

4.34.1 Synopsis

`locale [ −a | −m ]`

`locale [−ck]` *name* . . .

4.34.2 Description

The `locale` utility shall write information about the current locale environment, or all public locales, to the standard output. For the purposes of this clause, a *public locale* is one provided by the implementation that is accessible to the application.

When `locale` is invoked without any arguments, it shall summarize the current locale environment for each locale category as determined by the settings of the environment variables defined in 2.5.

When invoked with operands, it shall write values that have been assigned to the keywords in the locale categories, as follows:

— Specifying a keyword name shall select the named keyword and the category containing that keyword.

— Specifying a category name shall select the named category and all keywords in that category.