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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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

with the −v option of grep.

4.32 kill — Terminate or signal processes

4.32.1 Synopsis

kill −s *signal_name* *pid* ...

kill −l [*exit_status*]

Obsolescent Versions :

kill [−*signal_name*] *pid* ...

kill [−*signal_number*] *pid* ...

4.32.2 Description

The kill utility shall send a signal to the process(es) specified by each *pid* operand.

For each *pid* operand, the kill utility shall perform actions equivalent to the POSIX .1 {8}*kill* () function called with the following arguments:

(1) The value of the *pid* operand shall be used as the *pid* argument.

(2) The *sig* argument is the value specified by the −s option, −*signal_number* option, or the −*signal_name* option, or by SIGTERM , if none of these options is specified.

4.32.3 Options

The kill utility shall conform to the utility argument syntax guidelines described in 2.10.2, except that in the obsolescent form, the −*signal_number* and −*signal_name* options are usually more than a single character.

The following options shall be supported by the implementation:

−l (The letter ell.) Write all values of *signal_name* supported by the implementation, if no operand is given. If an *exit_status* operand is given and it is a value of the ? shell special parameter (see 3.5.2 and wait in 4.70) corresponding to a process that was terminated by a signal, the *signal_name* corresponding to the signal that terminated the process shall be written. If an *exit_status* operand is given and it is the unsigned decimal integer value of a signal number, the *signal_name* (the POSIX .1 {8}-defined symbolic constant name without the SIG prefix) corresponding to that signal shall be written. Otherwise, the results are unspecified.

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

7327 −s *signal_name*

7328 Specify the signal to send, using one of the symbolic names
7329 defined for Required Signals or Job Control Signals in POSIX.1 {8}
7330 3.3.1.1. Values of *signal_name* shall be recognized in a case-
7331 independent fashion, without the SIG prefix. In addition, the
7332 symbolic name 0 shall be recognized, representing the signal
7333 value zero. The corresponding signal shall be sent instead of
7334 SIGTERM.

7335 −*signal_name*

7336 (Obsolescent.) Equivalent to −s *signal_name.*

7337 −*signal_number*

7338 (Obsolescent.) Specify a nonnegative decimal integer,
7339 *signal_number,* representing the signal to be used instead of
7340 SIGTERM , as the *sig* argument in the effective call to *kill* (). The
7341 correspondence between integer values and the *sig* value used is
7342 shown in the following table.

7343

signal_number	***sig* Value**
7344 0	0
7345 1	SIGHUP
7346 2	SIGINT
7347 3	SIGQUIT
7348 6	SIGABRT
7349 9	SIGKILL
7350 14	SIGALRM
7351 15	SIGTERM

7352 The effects of specifying any *signal_number* other than those
7353 listed in the table are undefined.

7354 In the obsolescent versions, if the first argument is a negative integer, it shall be
7355 interpreted as a −*signal_number* option, not as a negative *pid* operand specifying
7356 a process group.

7357 **4.32.4 Operands**

7358 The following operands shall be supported by the implementation:

7359 *pid* A decimal integer specifying a process or process group to be sig-
7360 naled. The process(es) selected by positive, negative, and zero
7361 values of the *pid* operand shall be as described for POSIX.1 {8}
7362 *kill* () function. If the first *pid* operand is negative, it should be
7363 preceded by -- to keep it from being interpreted as an option.

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

7364 *exit_status* A decimal integer specifying a signal number or the exit status of
7365 a process terminated by a signal.

7366 ### 4.32.5 External Influences

7367 #### 4.32.5.1 Standard Input

7368 None.

7369 #### 4.32.5.2 Input Files

7370 None.

7371 #### 4.32.5.3 Environment Variables

7372 The following environment variables shall affect the execution of `kill`:

7373 **LANG** This variable shall determine the locale to use for the
7374 locale categories when both **LC_ALL** and the correspond-
7375 ing environment variable (beginning with **LC_**) do not
7376 specify a locale. See 2.6.

7377 **LC_ALL** This variable shall determine the locale to be used to over-
7378 ride any values for locale categories specified by the set-
7379 tings of **LANG** or any environment variables beginning
7380 with **LC_**.

7381 **LC_CTYPE** This variable shall determine the locale for the interpreta-
7382 tion of sequences of bytes of text data as characters (e.g.,
7383 single- versus multibyte characters in arguments).

7384 **LC_MESSAGES** This variable shall determine the language in which mes-
7385 sages should be written.

7386 #### 4.32.5.4 Asynchronous Events

7387 Default.

7388 ### 4.32.6 External Effects

7389 #### 4.32.6.1 Standard Output

7390 When the `-l` option is not specified, the standard output shall not be used.

7391 When the `-l` option is specified, the symbolic name of each signal shall be written
7392 in the following format:

7393 `"%s%c"`, *<signal_name>* , *<separator>*

7394 where the *<signal_name>* is in uppercase, without the `SIG` prefix, and the

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

7395 <*separator*> shall be either a <newline> or a <space>. For the last signal writ-
7396 ten, <*separator*> shall be a <newline>.

7397 When both the −l option and *exit_status* operand are specified, the symbolic name
7398 of the corresponding signal shall be written in the following format:

7399 "%s\n", <*signal_name*>

7400 #### 4.32.6.2 Standard Error

7401 Used only for diagnostic messages.

7402 #### 4.32.6.3 Output Files

7403 None.

7404 ### 4.32.7 Extended Description

7405 None.

7406 ### 4.32.8 Exit Status

7407 The kill utility shall exit with one of the following values:

7408 0 At least one matching process was found for each *pid* operand, and the
7409 specified signal was successfully processed for at least one matching
7410 process.

7411 >0 An error occurred.

7412 ### 4.32.9 Consequences of Errors

7413 Default.

7414 ### 4.32.10 Rationale. *(This subclause is not a part of P1003.2)*

7415 **Examples, Usage**

7416 Any of the commands

```
7417      kill −9 100 -165
7418      kill −s kill 100 -165
7419      kill −s KILL 100 -165
```

7420 sends the SIGKILL signal to the process whose process ID is 100 and to all
7421 processes whose process group ID is 165, assuming the sending process has per-
7422 mission to send that signal to the specified processes, and that they exist.

7423 POSIX .1 {8} and POSIX. 2 do not require specific signal numbers for any
7424 *signal_names* . Even the −*signal_number* option provides symbolic (although

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

numeric) names for signals. If a process is terminated by a signal, its exit status indicates the signal that killed it, but the exact values are not specified. The `kill −l` option, however, can be used to map decimal signal numbers and exit status values into the name of a signal. The following example reports the status of a terminated job:

```
job
stat=$?
if [ $stat -eq 0 ]
then
        echo job completed successfully.
elif [ $stat -gt 128 ]
then
        echo job terminated by signal SIG$(kill -l $stat).
else
        echo job terminated with error code $stat.
fi
```

History of Decisions Made

The signal name extension was based on a desire to avoid limiting the `kill` utility to implementation-dependent values.

The `−l` option originated from the C-shell, and is also implemented in the KornShell. The C-shell output can consist of multiple output lines, because the signal names do not always fit on a single line on some terminal screens. The KornShell output also included the implementation-specific signal numbers, and was felt by the working group to be too difficult for scripts to parse conveniently. The specified output format is intended not only to accommodate the historical C-shell output, but also to permit an entirely vertical or entirely horizontal listing on systems for which this is appropriate.

An earlier draft invented the name SIGNULL as a *signal_name* for signal 0 (used by POSIX.1 {8} to test for the existence of a process without sending it a signal). Since the *signal_name* `"0"` can be used in this case unambiguously, SIGNULL has been removed.

An earlier draft also required symbolic *signal_name* s to be recognized with or without the `SIG` prefix. Historical versions of `kill` have not written the `SIG` prefix for the `−l` option and have not recognized the `SIG` prefix on *signal_name* s. Since neither application portability nor ease of use would be improved by requiring this extension, it is no longer required.

POSIX.2 contains no utility that browses for process IDs. Values for *pid* are available via the `!` and `$` parameters of the shell command language (see 3.5.2).

The use of numeric signal values was the subject of a long debate in the Working Group. During balloting, it was determined that their use should be declared obsolescent, but retained to provide backward compatibility to existing applications.

Existing implementations of `kill` permit negative *pid* operands representing process groups, but this was often unclearly documented. The assumption that an

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

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.

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