Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'kill.1p' command

$ man kill.1p

KILL(1P) POSIX Programmer's Manual KILL(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

kill ? terminate or signal processes

SYNOPSIS

kill -s signal_name pid...

kill -l [exit_status]

kill [-signal_name] pid...

kill [-signal_number] pid...

DESCRIPTION

The kill utility shall send a signal to the process or processes speci? fied by each pid operand.

For each pid operand, the kill utility shall perform actions equivalent to the kill() function defined in the System Interfaces volume of POSIX.1?2017 called with the following arguments:

* The value of the pid operand shall be used as the pid argument.
* The sig argument is the value specified by the -s option, -sig? nal_number option, or the -signal_name option, or by SIGTERM, if none of these options is specified.

OPTIONS

The kill utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines, except that in the last two SYNOPSIS forms, the -signal_number and -signal_name options are usually more than a single character.

The following options shall be supported:

-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 Section 2.5.2, Special Parameters and wait) 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 symbolic constant name without the SIG prefix defined in the Base Definitions volume of POSIX.1?2017) corresponding to that signal shall be written. Otherwise, the results are unspecified.

-s signal_name

Specify the signal to send, using one of the symbolic names defined in the <signal.h> header. Values of signal_name shall be recognized in a case-independent fashion, without the SIG prefix. In addition, the symbolic name 0 shall be recognized, representing the signal value zero. The corresponding signal shall be sent instead of SIGTERM.

-signal_name

Equivalent to -s signal_name.

-signal_number

Specify a non-negative decimal integer, signal_number, representing the signal to be used instead of SIGTERM, as the sig argument in the effective call to kill(). The correspondence between integer values and the sig value used is shown in the following list.

The effects of specifying any signal_number other than those listed below are undefined.

0 0

1 SIGHUP

2 SIGINT

3 SIGQUIT

6 SIGABRT

9 SIGKILL

14 SIGALRM

15 SIGTERM

If the first argument is a negative integer, it shall be in? terpreted as a -signal_number option, not as a negative pid operand specifying a process group.

OPERANDS

The following operands shall be supported:

pid One of the following:

1. A decimal integer specifying a process or process group to be signaled. The process or processes selected by positive, negative, and zero values of the pid operand shall be as described for the kill() function. If process number 0 is specified, all processes in the current process group shall be signaled. For the effects of nega? tive pid numbers, see the kill() function defined in the System Interfaces volume of POSIX.1?2017. If the first pid operand is negative, it should be preceded by "--" to keep it from being interpreted as an option.
2. A job control job ID (see the Base Definitions volume of POSIX.1?2017, Section 3.204, Job Control Job ID) that identifies a background process group to be signaled. The job control job ID notation is applicable only for invo? cations of kill in the current shell execution environ? ment; see Section 2.12, Shell Execution Environment.

exit_status

A decimal integer specifying a signal number or the exit sta?
tus of a process terminated by a signal.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of kill:

LANG Provide a default value for the internationalization vari?
ables that are unset or null. (See the Base Definitions vol?
ume of POSIX.1?2017, Section 8.2, Internationalization Vari?
ables for the precedence of internationalization variables
used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of
all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of
bytes of text data as characters (for example, single-byte as
opposed to multi-byte characters in arguments).

LC_MESSAGES
Determine the locale that should be used to affect the format
and contents of diagnostic messages written to standard er?
ror.

NLSPATH Determine the location of message catalogs for the processing
of LC_MESSAGES.

ASYNCHRONOUS EVENTS

Default.

STDOUT

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

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

"%s%c", <signal_name>, <separator>

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

the <separator> shall be either a <newline> or a <space>. For the last signal written, <separator> shall be a <newline>.

When both the -l option and exit_status operand are specified, the sym? bolic name of the corresponding signal shall be written in the follow? ing format:

"%s\n", <signal_name>

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

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

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

Process numbers can be found by using ps.

The job control job ID notation is not required to work as expected when kill is operating in its own utility execution environment. In ei? ther of the following examples:

```
nohup kill %1 &
system("kill %1");
```

the kill operates in a different environment and does not share the shell's understanding of job numbers.

EXAMPLES

Any of the commands:

```
kill -9 100 -165
```

```
kill -s kill 100 -165
kill -s KILL 100 -165
```

sends the SIGKILL signal to the process whose process ID is 100 and to all processes whose process group ID is 165, assuming the sending process has permission to send that signal to the specified processes, and that they exist.

The System Interfaces volume of POSIX.1?2017 and this volume of POSIX.1?2017 do not require specific signal numbers for any sig? nal_names. Even the -signal_number option provides symbolic (although 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 dec? imal 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
```

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
```

RATIONALE

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 implementa?
tion-defined signal numbers and was considered by the standard develop?
ers to be too difficult for scripts to parse conveniently. The speci?
fied output format is intended not only to accommodate the historical C
shell output, but also to permit an entirely vertical or entirely hori?
zontal listing on systems for which this is appropriate.

An early proposal invented the name SIGNULL as a signal_name for signal
0 (used by the System Interfaces volume of POSIX.1?2017 to test for the
existence of a process without sending it a signal). Since the sig?
nal_name 0 can be used in this case unambiguously, SIGNULL has been re?
moved.

An early proposal also required symbolic signal_names 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_names. Since neither applications portability nor
ease-of-use would be improved by requiring this extension, it is no
longer required.

To avoid an ambiguity of an initial negative number argument specifying
either a signal number or a process group, POSIX.1?2008 mandates that
it is always considered the former by implementations that support the
XSI option. It also requires that conforming applications always use
the "--" options terminator argument when specifying a process group,
unless an option is also specified.

The -s option was added in response to international interest in pro?
viding some form of kill that meets the Utility Syntax Guidelines.

The job control job ID notation is not required to work as expected
when kill is operating in its own utility execution environment. In ei?
ther of the following examples:

```
nohup kill %1 &
system("kill %1");
```

the kill operates in a different environment and does not understand
how the shell has managed its job numbers.

None.

SEE ALSO

Chapter 2, Shell Command Language, ps, wait

The Base Definitions volume of POSIX.1?2017, Section 3.204, Job Control Job ID, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines, <signal.h>

The System Interfaces volume of POSIX.1?2017, kill()

COPYRIGHT