

NAME

kill - send signal to a process

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <sys/types.h>
```

```
#include <signal.h>
```

int

```
kill(pid_t pid, int sig);
```

DESCRIPTION

The **kill()** system call sends the signal given by *sig* to *pid*, a process or a group of processes. The *sig* argument may be one of the signals specified in `sigaction(2)` or it may be 0, in which case error checking is performed but no signal is actually sent. This can be used to check the validity of *pid*.

For a process to have permission to send a signal to a process designated by *pid*, the user must be the super-user, or the real or saved user ID of the receiving process must match the real or effective user ID of the sending process. A single exception is the signal SIGCONT, which may always be sent to any process with the same session ID as the sender. In addition, if the `security.bsd.conservative_signals` sysctl(9) is set to 1, the user is not a super-user, and the receiver is set-uid, then only job control and terminal control signals may be sent (in particular, only SIGKILL, SIGINT, SIGTERM, SIGALRM, SIGSTOP, SIGTTIN, SIGTTOU, SIGTSTP, SIGHUP, SIGUSR1, SIGUSR2).

If *pid* is greater than zero:

The *sig* signal is sent to the process whose ID is equal to *pid*.

If *pid* is zero:

The *sig* signal is sent to all processes whose group ID is equal to the process group ID of the sender, and for which the process has permission; this is a variant of `killpg(2)`.

If *pid* is -1:

If the user has super-user privileges, the signal is sent to all processes excluding system processes (with P_SYSTEM flag set), process with ID 1 (usually `init(8)`), and the process sending the signal. If the user is not the super user, the signal is sent to all processes which the caller has permissions to, excluding the process sending the signal. No error is returned if any process could be signaled.

If the process number is negative but not -1, the signal is sent to all processes whose process group ID is equal to the absolute value of the process number. This is a variant of `killpg(2)`.

RETURN VALUES

The `kill()` function returns the value 0 if successful; otherwise the value -1 is returned and the global variable `errno` is set to indicate the error.

ERRORS

The `kill()` system call will fail and no signal will be sent if:

- | | |
|----------|---|
| [EINVAL] | The <i>sig</i> argument is not a valid signal number. |
| [ESRCH] | No process or process group can be found corresponding to that specified by <i>pid</i> . |
| [EPERM] | The sending process does not have permission to send <i>sig</i> to any receiving process. |

SEE ALSO

`getpgrp(2)`, `getpid(2)`, `killpg(2)`, `sigaction(2)`, `sigqueue(2)`, `raise(3)`, `init(8)`

STANDARDS

The `kill()` system call is expected to conform to IEEE Std 1003.1-1990 ("POSIX.1").

HISTORY

A version of the `kill()` function appeared in Version 3 AT&T UNIX. The signal number was added to the `kill()` function in Version 4 AT&T UNIX.