NAME

kill – send a signal to a process or a group of processes

SYNOPSIS

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

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

MT-LEVEL

Async-Signal-Safe

DESCRIPTION

kill( ) sends a signal to a process or a group of processes. The process or group of processes to which the signal is to be sent is specified by *pid*. The signal that is to be sent is specified by *sig* and is either one from the list given in **signal** (see **signal**(5)), or 0. If *sig* is 0 (the null signal), error checking is performed but no signal is actually sent. This can be used to check the validity of *pid*.

The real or effective user ID of the sending process must match the real or saved (from **exec**(2)) user ID of the receiving process unless the effective user ID of the sending process is super-user, (see **intro**(2)), or *sig* is **SIGCONT** and the sending process has the same session ID as the receiving process.

If *pid* is greater than 0, *sig* will be sent to the process whose process ID is equal to *pid*.

If *pid* is negative but not **(pid_t)–1**, *sig* will be sent to all processes whose process group ID is equal to the absolute value of *pid* and for which the process has permission to send a signal.

If *pid* is **0**, *sig* will be sent to all processes excluding special processes (see **intro**(2)) whose process group ID is equal to the process group ID of the sender.

If *pid* is **(pid_t)–1** and the effective user ID of the sender is not super-user, *sig* will be sent to all processes excluding special processes whose real user ID is equal to the effective user ID of the sender.

If *pid* is **(pid_t)–1** and the effective user ID of the sender is super-user, *sig* will be sent to all processes excluding special processes.

RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned and **errno** is set to indicate the error.

ERRORS

kill( ) will fail and no signal will be sent if one or more of the following are true:

EINVAL	*sig* is not a valid signal number.
EPERM	*sig* is **SIGKILL** and *pid* is **(pid_t)1** (that is, the calling process does not have permission to send the signal to any of the processes specified by *pid*).
EPERM	The effective user of the calling process does not match the real or saved user and is not super-user, and the calling process is not sending **SIGCONT** to a process that shares the same session ID.
ESRCH	No process or process group can be found corresponding to that specified by *pid*.

SEE ALSO **kill**(1), **intro**(2), **exec**(2), **getpid**(2), **getsid**(2), **setpgrp**(2), **sigaction**(2), **sigsend**(2), **signal**(3C), **signal**(5)

NOTES **sigsend**(2) is a more versatile way to send signals to processes.