

NAME

psignal, **kern_psignal**, **pgsignal**, **tdsignal** - post signal to a thread, process, or process group

SYNOPSIS

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

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

void

```
kern_psignal(struct proc *p, int signum);
```

void

```
pgsignal(struct pgrp *pgrp, int signum, int checkctty);
```

void

```
tdsignal(struct thread *td, int signum);
```

DESCRIPTION

These functions post a signal to a thread or one or more processes. The argument *signum* common to all three functions should be in the range [1-NSIG].

The **kern_psignal**() function posts signal number *signum* to the process represented by the process structure *p*. The **kern_psignal**() function used to be called **psignal**() but was renamed in order to eliminate a name collision with the libc function of that name and facilitate code reuse. With a few exceptions noted below, the target process signal disposition is updated and is marked as runnable, so further handling of the signal is done in the context of the target process after a context switch. Note that **kern_psignal**() does not by itself cause a context switch to happen.

The target process is not marked as runnable in the following cases:

- ⊕ The target process is sleeping uninterruptibly. The signal will be noticed when the process returns from the system call or trap.
- ⊕ The target process is currently ignoring the signal.
- ⊕ If a stop signal is sent to a sleeping process that takes the default action (see `sigaction(2)`), the process is stopped without awakening it.
- ⊕ SIGCONT restarts a stopped process (or puts them back to sleep) regardless of the signal action (e.g., blocked or ignored).

If the target process is being traced **kern_psignal()** behaves as if the target process were taking the default action for *signal*. This allows the tracing process to be notified of the signal.

The **pgsignal()** function posts signal number *signal* to each member of the process group described by *pg*. If *checkctty* is non-zero, the signal will be posted only to processes that have a controlling terminal. **pgsignal()** is implemented by walking along the process list headed by the field *pg_members* of the process group structure pointed at by *pg* and calling **kern_psignal()** as appropriate. If *pg* is NULL no action is taken.

The **tdsignal()** function posts signal number *signal* to the thread represented by the thread structure *td*.

SEE ALSO

sigaction(2), signal(9), tsleep(9)

HISTORY

The **psignal()** function was renamed to **kern_psignal()** in FreeBSD 9.0.