NAME

sigsend, sigsendset – send a signal to a process or a group of processes

SYNOPSIS

#include <signal.h>

int sigsend(idtype_t *idtype***, id_t** *id***, int** *sig***);**

int sigsendset(procset_t **psp***, int** *sig***);**

DESCRIPTION

sigsend( ) sends a signal to the process or group of processes specified by *id* and *idtype*. The signal to be sent is specified by *sig* and is either zero or one of the values listed in **signal**(5). If *sig* is zero (the null signal), error checking is performed but no signal is actually sent. This value can be used to check the validity of *id* and *idtype*.

The real or effective user ID of the sending process must match the real or effective user ID of the receiving process, unless the effective user ID of the sending process is super-user, or *sig* is **SIGCONT** and the sending process has the same session ID as the receiving process.

If *idtype* is **P_PID**, *sig* is sent to the process with process ID *id*.

If *idtype* is **P_PGID**, *sig* is sent to any process with process group ID *id*.

If *idtype* is **P_SID**, *sig* is sent to any process with session ID *id*.

If *idtype* is **P_UID**, *sig* is sent to any process with effective user ID *id*.

If *idtype* is **P_GID**, *sig* is sent to any process with effective group ID *id*.

If *idtype* is **P_CID**, *sig* is sent to any process with scheduler class ID *id* (see **priocntl**(2)).

If *idtype* is **P_ALL**, *sig* is sent to all processes and *id* is ignored.

If *id* is **P_MYID**, the value of *id* is taken from the calling process.

The process with a process ID of 0 is always excluded. The process with a process ID of 1 is excluded unless *idtype* is equal to **P_PID**.

sigsendset( ) provides an alternate interface for sending signals to sets of processes. This function sends signals to the set of processes specified by *psp*. *psp* is a pointer to a structure of type **procset_t**, defined in **<sys/procset.h>**, which includes the following members:

```
idop_t      p_op;
idtype_t    p_lidtype;
id_t        p_lid;
idtype_t    p_ridtype;
id_t        p_rid;
```

p_lidtype and **p_lid** specify the ID type and ID of one ("left") set of processes; **p_ridtype** and **p_rid** specify the ID type and ID of a second ("right") set of processes. ID types and IDs are specified just as for the *idtype* and *id* arguments to **sigsend( )**. **p_op** specifies the operation to be performed on the two sets of processes to get the set of processes the function is to apply to. The valid values for **p_op** and the processes they specify are:

POP_DIFF set difference: processes in left set and not in right set

POP_AND set intersection: processes in both left and right sets

POP_OR set union: processes in either left or right set or both

POP_XOR set exclusive-or: processes in left or right set but not in both

RETURN VALUES

On success, **sigsend( )** returns zero. On failure, it returns –1 and sets **errno** to indicate the error.

ERRORS

sigsend( ) and **sigsendset( )** fail if one or more of the following are true:

EINVAL *sig* is not a valid signal number.

EINVAL *idtype* is not a valid idtype field.

EINVAL *sig* is **SIGKILL**, *idtype* is **P_PID** and *id* is 1 (proc1).

EPERM The effective user of the calling process is not super-user, and its real or effective user ID does not match the real or effective user ID of the receiving process, and the calling process is not sending **SIGCONT** to a process that shares the same session.

ESRCH No process can be found corresponding to that specified by *id* and *idtype*.

In addition, **sigsendset( )** fails if:

EFAULT **psp** points to an illegal address.

SEE ALSO

kill(1), **getpid**(2), **kill**(2), **priocntl**(2), **signal**(3C), **signal**(5)