NAME

proc_signal, proc_ref, proc_unref – send a signal to a process

SYNOPSIS

#include <sys/ddi.h>
#include <sys/sunddi.h>
#include <sys/signal.h>

void *proc_ref(void);

void proc_unref(void ****pref*);**

int proc_signal(void ****pref*, int** ***sig*****);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

pref A handle for the process to be signalled.

sig Signal number to be sent to the process.

DESCRIPTION

This set of routines allows a driver to send a signal to a process. The routine **proc_ref( )** is used to retrieve an unambiguous reference to the process for signalling purposes. The return value can be used as a unique handle on the process, even if the process dies. Because system resources are committed to a process reference, **proc_unref( )** should be used to remove it as soon as it is no longer needed.

proc_signal( ) is used to send signal *sig* to the referenced process. The following set of signals may be sent to a process from a driver:

SIGHUP	The device has been disconnected
SIGINT	The interrupt character has been received
SIGQUIT	The quit character has been received
SIGPOLL	A pollable event has occurred.
SIGKILL	Kill the process (cannot be caught or ignored)
SIGWINCH	Window size change.
SIGURG	Urgent data are available.

See **signal**(5) for more details on the meaning of these signals.

If the process has exited at the time the signal was sent, **proc_signal( )** returns an error code; the caller should remove the reference on the process by calling **proc_unref( )**.

The driver writer *must* ensure that for each call made to **proc_ref( )**, there is exactly *one* corresponding call to **proc_unref( )**.

RETURN VALUES

proc_ref( )

pref An opaque handle used to refer to the current process.

proc_signal( )

0 The process existed before the signal was sent.

–1 The process no longer exists; no signal was sent.

CONTEXT

proc_unref( ) and **proc_signal( )** can be called from user or interrupt context. **proc_ref( )** should only be called from user context.

SEE ALSO

signal(5), **putnextctl1**(9F)

Writing Device Drivers