NAME

untimeout – cancel previous timeout function call

SYNOPSIS

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

int untimeout(int *id***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

id Identification value generated by a previous **timeout**(9F) function call.

DESCRIPTION

untimeout( ) cancels a pending **timeout**(9F) request. **untimeout( )** will not return until the pending callback is cancelled or has run. Because of this, locks acquired by the call-back routine should not be held across the call to **untimeout( )** or a deadlock may result.

RETURN VALUES

untimeout( ) returns **-1** if the *id* is not found. Otherwise, it returns an integer value greater than or equal to **0**.

CONTEXT

untimeout( ) can be called from user or interrupt context.

EXAMPLE

In the following example, the device driver has issued an IO request and is waiting for the device to respond. If the device does not respond within 5 minutes, the device driver will print out an error message to the console.

```
static void
xxtimeout_handler(caddr_t arg)
{
    struct xxstate *xsp = (struct xxstate *)arg;

    mutex_enter(&xsp->lock);
    cv_signal(&xsp->cv);
    xsp->timeout_id = 0;
    xsp->flags |= TIMED_OUT;
    mutex_exit(&xsp->lock);
}
static u_int
xxintr(caddr_t arg)
{
    struct xxstate *xsp = (struct xxstate *)arg;
     .
     .
     .
    mutex_enter(&xsp->lock);
    if (xsp->timeout_id != 0) {
        (void) untimeout(xsp->timeout_id);
        xsp->timeout_id = 0;
    }
```

```
    /* Service interrupt */

    cv_signal(&xsp->cv);
    mutex_exit(&xsp->lock);

    return(DDI_INTR_CLAIMED);
}

static void
xxcheckcond(struct xxstate *xsp)
{
     .
     .
     .
    mutex_enter(&xsp->lock);
    xsp->timeout_id = timeout(xxtimeout_handler,
      (caddr_t)xsp, (5 * drv_usectohz(1000000)));
    while (/* Waiting for interrupt  or timeout*/)
        cv_wait(&xsp->cv, &xsp->lock);

    if (xsp->flags & TIMED_OUT)
        cmn_err(CE_WARN, "Device not responding");
     .
     .
     .
    mutex_exit(&xsp->lock);
     .
     .
     .
}
```

SEE ALSO

open(9E), **cv_signal**(9F), **cv_wait_sig**(9F), **delay**(9F), **timeout**(9F)

Writing Device Drivers