NAME

delay – delay execution for a specified number of clock ticks

SYNOPSIS

#include <sys/ddi.h>

void delay(long *ticks***);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

ticks　　The number of clock cycles to delay.

DESCRIPTION

delay( ) provides a mechanism for a driver to delay its execution for a given period of time. Since the speed of the clock varies among systems, drivers should base their time values on microseconds and use **drv_usectohz**(9F) to convert microseconds into clock ticks.

delay( ) uses **timeout**(9F) to schedule an internal function to be called after the specified amount of time has elapsed. **delay( )** then waits until the function is called.

delay( ) does not busy-wait. If busy-waiting is required, use **drv_usecwait**(9F).

CONTEXT

delay( ) can be called from user context only.

EXAMPLES

Before a driver I/O routine allocates buffers and stores any user data in them, it checks the status of the device (line 12). If the device needs manual intervention (such as, needing to be refilled with paper), a message is displayed on the system console (line 14). The driver waits an allotted time (line 17) before repeating the procedure.

```
 1 struct  device {                      /* layout of physical device registers */
 2       int     control;                /* physical device control word */
 3       int     status;                 /* physical device status word */
 4       short   xmit_char;              /* transmit character to device */
 5 };
 6
 7
    . . .
 9                                       /* get device registers  */
10     register struct device *rp = ...
11
12     while (rp->status & NOPAPER) { /* while printer is out of paper */
13                                        /* display message and ring bell */
                                          /* on system console */
14        cmn_err(CE_WARN, "^xx_write: NO PAPER in printer %d\007",
15                (getminor(dev) & 0xf));
16        /* wait one minute and try again */
17        delay(60 * drv_usectohz(1000000));
18     }
```

SEE ALSO

biodone(9F), **biowait**(9F), **drv_hztousec**(9F), **drv_usectohz**(9F), **drv_usecwait**(9F), **timeout**(9F), **untimeout**(9F)

Writing Device Drivers