NAME

biodone – release buffer after buffer I/O transfer and notify blocked threads

SYNOPSIS

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

void biodone(struct buf ***bp*);**

INTERFACE LEVEL

Architecture independent level 1 (DDI/DKI).

ARGUMENTS

bp Pointer to a **buf**(9S) structure.

DESCRIPTION

biodone( ) notifies blocked processes waiting for the I/O to complete, sets the **B_DONE** flag in the **b_flags** field of the **buf**(9S) structure, and releases the buffer if the I/O is asynchronous. **biodone( )** is called by either the driver interrupt or **strategy**(9E) routines when a buffer I/O request is complete.

biodone( ) provides the capability to call a completion routine if *bp* describes a kernel buffer. The address of the routine is specified in the **b_iodone** field of the **buf**(9S) structure. If such a routine is specified, **biodone( )** calls it and returns without performing any other actions. Otherwise, it performs the steps above.

CONTEXT

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

EXAMPLES

Generally, the first validation test performed by any block device **strategy**(9E) routine is a check for an end-of-file (EOF) condition. The **strategy**(9E) routine is responsible for determining an EOF condition when the device is accessed directly. If a **read**(2) request is made for one block beyond the limits of the device (line 10), it will report an EOF condition. Otherwise, if the request is outside the limits of the device, the routine will report an error condition. In either case, report the I/O operation as complete (line 27).

```
 1 #define RAMDNBLK    1000                /* Number of blocks in RAM disk */
 2 #define RAMDBSIZ    512                 /* Number of bytes per block */
 3 char ramdblks[RAMDNBLK][RAMDBSIZ];  /* Array containing RAM disk */
 4
 5 static int
 6 ramdstrategy(struct buf *bp)
 7 {
 8      daddr_t blkno = bp->b_blkno;        /* get block number */
 9
10      if ((blkno < 0) || (blkno >= RAMDNBLK)) {
11           /*
12            * If requested block is outside RAM disk
13            * limits, test for EOF which could result
14            * from a direct (physio) request.
15            */
16           if ((blkno == RAMDNBLK) && (bp->b_flags & B_READ)) {
17                /*
```

```
          * If read is for block beyond RAM disk
          * limits, mark EOF condition.
          */
         bp->b_resid = bp->b_bcount;        /* compute return value */

       } else {                             /* I/O attempt is beyond */
         bp->b_error = ENXIO;               /*    limits of RAM disk */
         bp->b_flags |= B_ERROR;            /* return error */
       }
       biodone(bp);                         /* mark I/O complete (B_DONE) */
        /*
         * Wake any processes awaiting this I/O
         * or release buffer for asynchronous
         * (B_ASYNC) request.
         */
       return (0);
    }
  ...
```

SEE ALSO

read(2), **strategy**(9E), **biowait**(9F), **ddi_add_intr**(9F), **delay**(9F), **timeout**(9F), **untimeout**(9F), **buf**(9S)

Writing Device Drivers

NOTES

Drivers that use the **b_iodone** field of the **buf**(9S) structure to specify a substitute completion routine should save the value of **b_iodone** before changing it, and then restore the old value before calling **biodone( )** to release the buffer.