NAME

wd − ATA/IDE disk controller interface

SYNOPSIS

/sys/conf/SYSTEM:

```
        NWD wd_drives         # EIDE disk drive
```

/etc/dtab:

```
        #Name Unit# Addr   Vector Br Handler(s)        # Comments
        wd    ?     171000 510    4  wdintr            # EIDE disks
```

major device number(s):

raw: 27
block: 13

minor device encoding:

bits 0007 specify partition of WD drive
bits 0010 specify WD drive

DESCRIPTION

This is a driver for ATA-2 (IDE) compatible disk drives.

Files with minor device numbers 0 through 7 refer to various portions of drive 0; minor devices 8 through 15 refer to drive 1. Only two drives are supported under one IDE interface. The standard device names begin with ‘‘wd’’ followed by the drive number and then a letter a-h for partitions 0-7 respectively. The character ? stands here for a drive number in the range 0-7.

The block files access the disk via the system’s normal buffering mechanism and may be read and written without regard to physical disk records. There is also a ‘raw’ interface which provides for direct transmission between the disk and the user’s read or write buffer. A single read or write call results in exactly one I/O operation and therefore raw I/O is considerably more efficient when many words are transmitted. The names of the raw files conventionally begin with an extra ‘r.’

In raw I/O the buffer must begin on a word (even) boundary, and counts should be a multiple of 512 bytes (a disk sector). Likewise *seek* calls should specify a multiple of 512 bytes.

DISK SUPPORT

This driver configures the drive type of each drive when it is first opened. Partition information is read from the disklabel. If there is no label or the label is corrupt then the ’a’ partition is used to span the entire drive.

The wd?a partition is normally used for the root file system, the wd?b partition as a swap area, and the wd?c partition for pack-pack copying (it maps the entire disk).

FILES

/dev/wd[0-7][a-h]
/dev/rwd[0-7][a-h]
/dev/MAKEDEV   script to create special files

SEE ALSO

hk(4), ram(4), ra(4), rk(4), rl(4), rp(4), rx(4), si(4), xp(4), dtab(5), autoconfig(8), disklabel(8)

DIAGNOSTICS

wd: insw: no DRQ count = %d Drive failed to set DRQ after *count* word transfers.

wd%d: %s LBA mode NOT supported! Drive IDENTIFY command indicated that drive does not support logical block addressing (LBA). Drive is NOT compatible with current driver.

wd: timeout waiting for BSY Drive set BSY too long.

wd: timeout waiting for DRDY Drive took too long to become ready.

wd: timeout waiting for DRQ Drive took too long to request data transfer.

wd: write error st = %o er = %o An error occured during a write operation. Status register and error register contents are reported.

wd: extra interrupt An interrupt occured when none was pending. The interrupt is cleared and the error is ignored.

wd%d%c: hard read error bn %d An unrecoverable read error occured while reading the given block.

wd%d%c: hard write error bn %d An unrecoverable write error occured while reading the given block.

BUGS

In raw I/O *read* and *write*(2) truncate file offsets to 512-byte block boundaries, and *write* scribbles on the tail of incomplete blocks. Thus, in programs that are likely to access raw devices, *read*, *write* and *lseek*(2) should always deal in 512-byte multiples.

Only IDE drives capable of logical block addressing (LBA) are supported.

Data transfers only occur in PIO mode. This is a limitation of the interface and would be corrected if the interface was capable of DMA.