NAME

da - SCSI Direct Access device driver

SYNOPSIS

device da

DESCRIPTION

The **da** driver provides support for all SCSI devices of the direct access class that are attached to the system through a supported SCSI Host Adapter. The direct access class includes disk, magneto-optical, and solid-state devices.

A SCSI Host adapter must also be separately configured into the system before a SCSI direct access device can be configured.

CACHE EFFECTS

Many direct access devices are equipped with read and/or write caches. Parameters affecting the device's cache are stored in mode page 8, the caching control page. Mode pages can be examined and modified via the camcontrol(8) utility.

The read cache is used to store data from device-initiated read ahead operations as well as frequently used data. The read cache is transparent to the user and can be enabled without any adverse effect. Most devices with a read cache come from the factory with it enabled. The read cache can be disabled by setting the RCD (Read Cache Disable) bit in the caching control mode page.

The write cache can greatly decrease the latency of write operations and allows the device to reorganize writes to increase efficiency and performance. This performance gain comes at a price. Should the device lose power while its cache contains uncommitted write operations, these writes will be lost. The effect of a loss of write transactions on a file system is non-deterministic and can cause corruption. Most devices age write transactions to limit vulnerability to a few transactions recently reported as complete, but it is none-the-less recommended that systems with write cache enabled devices reside on an Uninterruptible Power Supply (UPS). The **da** device driver ensures that the cache and media are synchronized upon final close of the device or an unexpected shutdown (panic) event. This ensures that it is safe to disconnect power once the operating system has reported that it has halted. The write cache can be enabled by setting the WCE (Write Cache Enable) bit in the caching control mode page.

TAGGED QUEUING

The **da** device driver will take full advantage of the SCSI feature known as tagged queueing. Tagged queueing allows the device to process multiple transactions concurrently, often re-ordering them to reduce the number and length of seeks. To ensure that transactions to distant portions of the media, which may be deferred indefinitely by servicing requests nearer the current head position, are completed in a timely fashion, an ordered tagged transaction is sent every 15 seconds during continuous device operation.

BAD BLOCK RECOVERY

Direct Access devices have the capability of mapping out portions of defective media. Media recovery parameters are located in mode page 1, the Read-Write Error Recovery mode page. The most important media remapping features are 'Auto Write Reallocation' and 'Auto Read Reallocation' which can be enabled via the AWRE and ARRE bits, respectively, of the Read-Write Error Recovery page. Many devices do not ship from the factory with these feature enabled. Mode pages can be examined and modified via the camcontrol(8) utility.

KERNEL CONFIGURATION

It is only necessary to explicitly configure one **da** device; data structures are dynamically allocated as disks are found on the SCSI bus.

SYSCTL VARIABLES

The following variables are available as both sysctl(8) variables and loader(8) tunables:

kern.cam.da.default_timeout

This variable determines how long the **da** driver will wait before timing out an outstanding command. The units for this value are seconds, and the default is currently 60 seconds.

kern.cam.da.disable_wp_protection

Disable detection of write-protected disks. Default is disabled. (detection of write-protected disks is enabled).

kern.cam.da.enable_biospeedup

Enable BIO_SPEEDUP processing. Default is enabled.

kern.cam.da.enable_uma_ccbs

Use UMA for CCBs. Default is enabled.

kern.cam.da.poll_period

Media polling period in seconds. Default is 3 seconds.

kern.cam.da.retry_count

This variable determines how many times the **da** driver will retry a READ or WRITE command. This does not affect the number of retries used during probe time or for the **da** driver dump routine. This value currently defaults to 4.

kern.cam.da.send_ordered

Send Ordered Tags. On shutdown, step through all the **da** peripheral drivers, and if the device is still open, sync the disk to physical media. Default is enabled.

kern.cam.sort_io_queue

kern.cam.da.X.sort_io_queue

These variables determine whether request queue should be sorted trying to optimize head seeks. Set to 1 to enable sorting, 0 to disable, -1 to leave it as-is. The default is sorting enabled for HDDs and disabled for SSDs.

kern.cam.da.X.delete_method

This variable specifies method to handle BIO_DELETE requests:

ATA_TRIM	ATA TRIM via ATA COMMAND PASS THROUGH command,
UNMAP	UNMAP command,
WS16	WRITE SAME(16) command with UNMAP flag,
WS10	WRITE SAME(10) command with UNMAP flag,
ZERO	WRITE SAME(10) command without UNMAP flag,
DISABLE	disable BIO_DELETE support.

kern.cam.da.X.minimum_cmd_size

This variable determines what the minimum READ/WRITE CDB size is for a given **da** unit. Valid minimum command size values are 6, 10, 12 and 16 bytes. The default is 6 bytes.

The **da** driver issues a CAM Path Inquiry CCB at probe time to determine whether the protocol the device in question speaks (e.g. ATAPI) typically does not allow 6 byte commands. If it does not, the **da** driver will default to using at least 10 byte CDBs. If a 6 byte READ or WRITE fails with an ILLEGAL REQUEST error, the **da** driver will then increase the default CDB size for the device to 10 bytes and retry the command. CDB size is always chosen as the smallest READ/WRITE CDB that will satisfy the specified minimum command size, and the LBA and length of the READ or WRITE in question. (e.g., a write to an LBA larger than 2^32 will require a 16 byte CDB.)

NOTES

If a device becomes invalidated (media is removed, device becomes unresponsive) the disklabel and information held within the kernel about the device will be invalidated. To avoid corruption of a newly inserted piece of media or a replacement device, all accesses to the device will be discarded until the last file descriptor referencing the old device is closed. During this period, all new open attempts will be rejected.

FILES

*/dev/da** SCSI disk device nodes

DIAGNOSTICS

None.

SEE ALSO

ada(4), cam(4), geom(4), nda(4), gpart(8)

HISTORY

The **da** driver was written for the CAM SCSI subsystem by Justin T. Gibbs. Many ideas were gleaned from the **sd** device driver written and ported from Mach 2.5 by Julian Elischer.