NAME

sd, st, sg − SCSI hard disk / tape / generic

DESCRIPTION

The **sd***, **st***, **sg*** family of devices refer to the SCSI hard disk, tape and generic driver using the Adaptec 154x series of controllers. This manual page only describes the differences between the sd and hd devices, read **hd**(4) first.

The devices numbers of the SCSI devices are statically mapped onto the SCSI targets 0 to 7. This is done like the hd devices with **sd[0−4]** referring to target 0, **sd[5−9]** to target 1, etc. The logical unit number is always 0, because devices with more than one logical unit are virtually extinct. The mapping may be changed from the boot environment however (see **boot**(8)).

Tapes start at minor device 64, with **nrst0** at minor 64, **rst0** at 65, **nrst1** at 66, etc. The mapping is again static to target (minor − 64) / 2. The **rst** devices rewind the tape on close, the **nrst** devices do not. See **mt**(1), and **mtio**(4) for a description of the commands that may be sent to the tape, either from the command prompt or from a program.

Through the eight raw generic devices **rsg[0−7]** starting at minor 120, one can send SCSI commands to any SCSI device from user mode. (Minix-vmd only.)

The driver returns a drive geometry of 64 heads by 32 sectors per track for small disks with the DIOCGETP ioctl. For large disks 255x63 is returned. The size in sectors is usually larger than the largest cylinder number indicates, because the disk is not likely to exactly match that faked geometry. Note that DOS may not be able to access those last few sectors.

Disk like devices.

Removable disks (floppies), CD-ROM’s and WORM disks may also be accessed through the **sd** devices. One is not allowed to write a WORM disk however, because it is likely to be taken from an alien operating system, so it seems safer to not allow Minix to stomp over it. One usually needs special O.S. support to keep one from writing to the same block twice.

The **DIOCEJECT** ioctl ejects CD-ROMs, floppies, etc. (See **eject**(1).) A fixed disk spins down if it supports the stop command.

SCSI Tapes

There are two types of SCSI tapes drives supported by the driver: fixed or variable block size tape drives. Examples of the first kind are cartridge tapes, with a fixed 512 bytes block size. An Exabyte tape drive has a variable block size, with a minimum of 1 byte and a maximum of 245760 bytes (see the documentation of such devices.) The maximum is truncated to 32767 bytes for Minix-86 and 61440 bytes for Minix-vmd, because the driver can’t move more bytes in a single request.

A read or write to a fixed block size tape must be a precise multiple of the block size, any other count gives results in an I/O error. A read from a variable block sized tape must be large enough to accept the block that is read, otherwise an I/O error will be returned. A write can be any size above the minimum, creating a block of that size. If the write count is larger than the maximum block size then more blocks are written until the count becomes zero. The last block must be larger than the minimum of course. (This minimum is often as small as 1 byte, as for the Exabyte.)

The **mt blksize** command may be used to select a fixed block size for a variable block sized tape. This will speed up I/O considerably for small block sizes. (Some systems can only use fixed mode and will write an Exabyte tape with 1024 byte blocks, which read very slow in variable mode.)

A tape is a sequence of blocks and filemarks. A tape may be opened and blocks may be read from it upto a filemark, after that all further reads return 0. After the tape is closed and reopened one can read the blocks following the filemark if using a non-rewinding device. This makes the tape look like a sequence of files.

If a tape has been written to or opened in write-only mode, then a filemark is written if the tape is closed or if a space command is issued. No extra filemark is written if the drive is instructed to write filemarks.

Raw Generic Devices

Under Minix-vmd one can use the generic SCSI devices to program a SCSI device entirely from user mode. The disk and tape devices probe for devices when opened, start disks and load tapes, but the generic devices do nothing of this. Given an open file descriptor to any SCSI character device (not just the generic devices) one can use the following ioctl:

```
ioctl(fd, SCIOCCMD, &scsicmd)
```

The structure whose address is passed as the third argument is defined in <sys/scsi.h> as follows:

```
struct scsicmd {
        void     *cmd;
        size_t   cmdlen;
        void     *buf;
        size_t   buflen;
        void     *sense;
        size_t   senselen;
        int      dir;
};
```

Cmd and **cmdlen** hold the address and length of an object holding a Group 0 or Group 1 SCSI command. The next two fields describe a buffer of at most 8 kilobytes used in the data in or out phase. **Dir** is 0 if data is to be read from the device, 1 if data is written to the device. If the ioctl succeeds then 0 is returned, otherwise -1 with **errno** set to **EIO** and the request sense info returned in the buffer described by the sense and senselen fields. If the sense key is zero on error then a host adapter error occurred, this means that the device is most likely turned off or not present.

FILES

/dev/sd[0−9], /dev/sd[1−46−9][a−d]	Usual disk devices.
/dev/rst4, /dev/nrst4	Usual tape device.
/dev/rsg[0−7]	Raw generic devices.

SEE ALSO

hd(4), **mt**(1), **eject**(1), **mtio**(4), **dd**(1).

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)