

NAME

tm – TM11/TU10 magtape interface

DESCRIPTION

The files *mt0*, ..., *mt7* refer to the Digital Equipment Corporation TM11/TU10 magnetic tape control and transports at 800bpi. The files *mt0*, ..., *mt3* are designated normal-rewind on close, and the files *mt4*, ..., *mt7* are no-rewind on close. When opened for reading or writing, the tape is assumed to be positioned as desired. When a file is closed, a double end-of-file (double tape mark) is written if the file was opened for writing. If the file was normal-rewind, the tape is rewound. If it is no-rewind and the file was open for writing, the tape is positioned before the second EOF just written. If the file was no-rewind and opened read-only, the tape is positioned after the EOF following the data just read. Once opened, reading is restricted to between the position when opened and the next EOF or the last write. The EOF is returned as a zero-length read. By judiciously choosing *mt* files, it is possible to read and write multi-file tapes.

A standard tape consists of several 512 byte records terminated by an EOF. To the extent possible, the system makes it possible, if inefficient, to treat the tape like any other file. Seeks have their usual meaning and it is possible to read or write a byte at a time (although very inadvisable).

The *mt* files discussed above are useful when it is desired to access the tape in a way compatible with ordinary files. When foreign tapes are to be dealt with, and especially when long records are to be read or written, the “raw” interface is appropriate. The associated files are named *rmt0*, ..., *rmt7*. Each *read* or *write* call reads or writes the next record on the tape. In the write case the record has the same length as the buffer given. During a read, the record size is passed back as the number of bytes read, up to the buffer size specified. In raw tape I/O, the buffer must begin on a word boundary and the count must be even. Seeks are ignored. An EOF is returned as a zero-length read, with the tape positioned after the EOF, so that the next read will return the next record.

FILES

/dev/mt?, /dev/rmt?

SEE ALSO

tp(I)

BUGS

If any non-data error is encountered, it refuses to do anything more until closed. The driver is limited to four transports.