Tapes and U(NIX)

The various tape formats available today can be quite confusing to understand. This article attempts to clear up some of the confusion, and help you deal with these tape formats.

By Roland Schneider

Any of us who have ever typed the command "`rm -r *`", or seen the lights flicker and the computer system die, know the momentary feeling of panic that comes with the realization of just how transient those little magnetized spots on the system's disks really are. When you consider that each of the 10 billion bits of data on a modern hard disk occupies less than 0.00002 square millimeters, and they *all* have to be right, every time, you feel a little uneasy.

Of course, that's why we make backups and archives. Problem is, though, the bits on the tape are just as fragile as those on the disk, and since most people don't use tape commands very often, it's easy to make a mistake. A general understanding of how tape drives work helps avoid *some* of the possible mistakes, and also helps when exchanging tapes with other computer systems.

The common tape formats in use on UNIX systems are, from oldest to newest: 1/2" reels at 800, 1600, or 6250 bpi (bytes per inch), 1/4" cartridge with 4, 9, 15, or 18 tracks, 8mm videotape (Exabyte) at low or high density, and 4mm DAT. Apart from simply being grossly incompatible, each tape format has its own set of operational subtleties.

1/2", 9 track reel tapes

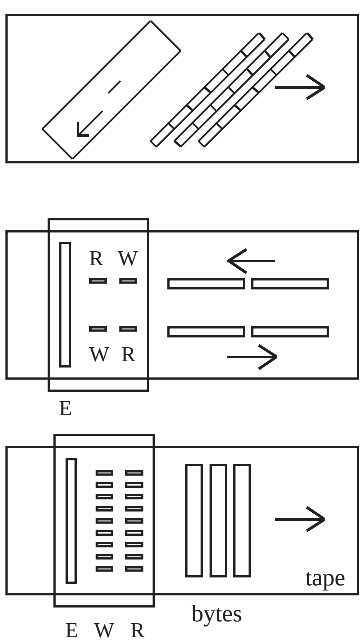
This used to be the most common type of tape drive found in larger computer installations. The tapes are bulky and the drives themselves are larger (and more expensive) than most modern computers. However, the UNIX tape utilities were written with this type of drive in mind, and many are still in use, so they bear some examination.

Data is recorded in 9 tracks simultaneously across the tape – eight data bits plus parity. During a tape write operation, the tape first passes over the erase head, which demagnetizes the tape, then across the write heads, which magnetize the tape in one of two directions, depending if a "1" or a "0" is being written, and then across the read heads, which read the data back to make sure it was successfully written.

In all but the lowest performance 1/2" drives, the tape is kept under tension by running a loop from each reel into a vacuum column. Because only the mass of the tape itself has to be accelerated, the tape can start, stop, and reverse very abruptly – much more abruptly than the tape reels themselves could.

A single UNIX `write()` system call will write a single block of data (sometimes called a record, just to confuse things) onto the tape, followed by an inter-record gap. The tar "b" option sets the size of this block in 512 byte increments. The upper limit for the blocksize depends on the drive hardware, but it is frequently 10K bytes, corresponding to a tar blocksize of 20. Big blocksizes usually give a higher data transfer rate (fewer starts and stops) and allow more data to be put on a tape. (fewer inter-record gaps)

When a tape device opened for writing is closed, two *tape marks* are written. The first indicates the end of the file, the second the end of the tape. If another file is later written on the same tape, the second tape mark is erased, leaving



Format: 8mm 2.2Gb Helical Scan (Exabyte)
Tracks: 1 long track, in diagonal stripes across tape
Head: on rotating, diagonal drum

Format: QIC 150 (150Mb 1/4" cartridge)
Tracks: 18, recorded one at a time in alternating directions
Head: 2 read, 2 write (one for each direction), head moves for other tracks, *all erased at once*

Format: 6250 BPI (150Mb 1/2" reel)
Tracks: 9, recorded simultaneously
Head: 9 read, 9 write (one for each bit + parity)

E = Erase, W = Write, R = Read

only the end-of-file indication, and two marks will be written at the end of the new file.

A single `read()` system call will read one tape block. If the size of the read exceeds the size of the block, `read()` returns a byte count indicating how many bytes were actually read. Many implementations of tar use this technique to automatically figure out the correct block size when reading a tape.

Because the erase head is just ahead of the write head, it is possible for a 1/2" tape drive to *append* to the last file on a tape by erasing both the tape marks at the end of a file and writing more blocks of data. The tar “`u`” and “`r`” options use this feature. It is actually possible to start writing *anywhere* on a tape, although this effectively destroys all existing data following the current tape position.

1/4" cartridge (QIC) tape

With advances in recording technology, particularly the quality of the tape coatings, it became possible to store more bits on less tape. The 1/4" cartridge is now one of the most popular tape formats. The storage capacity of cartridges and 1/2" reel tapes is about the same, but the cartridges are smaller and more convenient to handle, and the drives are much simpler and cheaper.

A cartridge tape drive writes data onto tape serially, one bit at a time, instead of 9 bits at a time like a 1/2" drive. As a result, the bytes are recorded lengthwise on the tape instead of across it. Why do it this way? Well, for one, it makes the tape head much simpler. Instead of having 9 read and 9 write heads crammed beside one another, 1/4" drives have only one read and one write head for each direction. This makes the drive cheaper, and allows greater information density on the tape. Incidentally, both floppy and hard disks also write data serially.

Depending on the vintage of the drive, there can be 4, 9, 15, or 18 tracks of data on the tape. (Newer drives can usually read but not write older formats) When one track has been written, the tape reverses and the tape heads are repositioned so that the next track can be written, in the opposite direction. The read and write heads are arranged so that the read head always follows the write head, allowing the data to be verified as it is written. There is only *one* erase head, and it covers the entire width of the tape, so *all* the tape tracks are erased simultaneously when track 0 is being written. Because it is impossible to selectively erase one track, it's also impossible to erase the end-of-file mark and extend a tape file. That's why the tar “u” and “r” options don't work for cartridge tapes. It also means that if you accidentally write something at the beginning of a 1/4" tape when you meant to add a new tape file at the end, you effectively destroy the whole tape, not just the data you wrote over. The *only* places you are allowed to write are at the beginning of tape or after the last file on the tape.

Quarter inch tape drives deal with tape blocks quite differently than 1/2" drives. Each tape block is 512 bytes long, and has extra error-checking and synchronization information attached to it. (That's what the QIC-?? standards specify) As a result, all `write()` system calls *must* be in multiples of 512 bytes.

There is no equivalent to the inter-record gap found on 1/2" tapes. When a 1/4" drives wants to read or write a block, it first backs up a bit so that it can get the tape up to speed before it has to start reading or writing. This is in contrast to the extremely sudden starts and stops a 1/2" tape with a vacuum column is capable of. All this starting, stopping, and backing up does two things: it wastes a *lot* of time, and it messes up the tension with which the tape is wound on the reels. The “`mt retention`” command can fix the latter.

The best solution to the start/stop problem is to feed the drive data quickly enough so that it can write continuously, without ever stopping. This is called *streaming*, and drives that work best this way are frequently called *streaming tape drives*, or simply *streamers.* If it isn't possible to stream the tape, large `write()`'s should be performed so that as much data as possible is written at once. The same applies in reverse when reading. Recommended tar block sizes are 126 or 200, although larger sizes are allowed.

8mm Helical Scan (Exabyte)

Eight millimeter tape drives have only been available for a few years, but their incredibly high information density has made them very popular with system administrators with large disks to back up. Not only is the media cheap, ($15 for 4.5 Gb) but being able to dump several entire file systems onto a single tape, unattended, is an attractive alternative to swapping tapes every twenty-five minutes for twelve hours. Even for a medium-sized system, the cost of an 8mm drive can be justified in terms of media savings alone.

An 8mm tape drive is basically an 8mm VCR where the video electronics have been replaced with a digital computer interface. As with all VCRs, the head is mounted on a rotating drum and moves diagonally across the tape. Data is recorded serially, as with a 1/4" tape, but there is only one very long track split into short diagonal bands across the tape. Sophisticated error correction coding is used to ensure reliability.

Like 1/4" drives, 8mm drives waste a lot of time if they have to start and stop, so large block sizes are usually preferable. Note, though, that the UNIX “dump” utility uses multiple processes to overlap disk reads and tape writes to keep the tape busy, so the default block size should be used for dumps onto 8mm.

In most other respects, 8mm tapes are more like 1/2" tapes than 1/4" cartridges. They can write variable-sized blocks, up to 65535 bytes, and can start writing at the beginning of the tape or after the end of any file.

Tape Head Alignment

The different tape standards specify exactly where on the tape each track is supposed to be. A track on an 18 track 1/4" cartridge tape is less than 0.35 mm wide. It is therefore essential that the tape heads are positioned in exactly the right spot for each track. Tape drives have an internal adjustment to set the alignment.

Misaligned heads usually show up as read errors when trying to load someone else's tape. However, the problem can be much more insidious than that: you will be able to read your own tapes, because the alignment was the same for both reading and writing. But what happens when your drive breaks down and is replaced or realigned? Suddenly, all the tapes written by the misaligned drive are unreadable, and valuable archived data is lost. Of course, with a lot of effort and a competent serviceman, the new drive can be misaligned like the old one and the tapes read, but this is, at best, a tricky procedure. The moral: get your tape head alignment checked if you are even a little suspicious.

Conclusions

Of course, all this only deals with the process of getting the data on and off the tape. It says nothing about attaching any meaning to that data. For transfers between UNIX systems, that isn't usually a problem, since it is just a matter of using the same utility for reading and writing. Exchanging data with other systems is more interesting, but that's a topic for a future article.