The Ftape-FAQ

Johan De Wit, <jdewit@unicall.be> v0.1, 13 october 1997

This is a very incomplete attempt to create a FAQ for the *Ftape Floppy Tape Device Driver*. Any suggestions, remarks ... that could improve this FAQ are welcome indeed.

Contents

1 A Word From The Maintainer Of Ftape.

This is a very incomplete attempt to create a FAQ for the *Ftape Floppy Tape Device Driver*.

You might encounter references to the following addresses while reading this document:

- The maintainer of the Ftape FAQ :
  Johan De Wit <jdewit@unicall.be>
- The Ftape maintainer :
  Claus-Justus Heine <claus@momo.math.rwth-aachen.de>
- The Ftape Home Page :
  <http://www-math.math.rwth-aachen.de/ LBFM/claus/ftape/ftape-page.html>
- The Ftape HOWTO :
  <http://sunsite.unc.edu/LDP/HOWTO>
- The Ftape Mailing List :
  <linux-tape@vger.rutgers.edu>

There is surely quite a lot missing. Please feel free to improve this FAQ. The preferred way of doing this is to post to the Ftape Mailing List in case you have a question that isn't answered here.

Also, if you are already reading the list regularly and have the impression that some questions occur again and again, feel free to send that question and possibly an answer in the format indicated below to the maintainer of the *Ftape FAQ* AND to Ftape Mailing List.

If you make FAQ related postings, then please DON'T FORGET to prepend the word "*FAQ*" to the subject of your posting. Please don't add the word "FAQ" to the subject if you post something that isn't related to the FAQ.

That's all for now.

Claus-Justus Heine.

2 "Compiling and installing Ftape" related questions !

2.1 What Ftape version should I use?

Always the latest stable version which is _supposed_ to be available from
ftp://sunsite.unc.edu/pub/Linux/kernel/tapes and

http://www-math.math.rwth-aachen.de/ LBFM/claus/ftape/ftape-page.html

However, for the time being you should try **ftape-3.03b-970603** which unluckily hasn't made its way to the sunsite archives as it was intended to be a beta version only and will hopefully soon be out-dated by the new version ftape-4.0.

ftape-3.03b-970603 is available from
http://www-math.math.rwth-aachen.de/ LBFM/claus/ftape/ftape-page.html ONLY.

<answer from Claus Heine>

2.2 I'm having problems getting my XYZ drive to run under the 2.0.xx kernel with the built-in driver. How do I fix this?

The default version of *Ftape* included in the 2.0.xx kernel sources is 2.08 or 2.09 and is very out of date. Please update the *Ftape drivers* to the latest from the Ftape Home Page.

<answer from Tim Jones>

2.3 I'm running Linux/SMP and the system just freezes when trying to access the Ftape devices!

You need to add `-D__SMP__` to the `KERNEL_OPT` variable in the file `MCONFIG`.

<answer from Claus Heine>

2.4 Why does depmod complain about "undefined symbols"?

Ignore the depmod error messages. The problem is that the *Ftape modules* have to be compiled without the version checksum feature (i.e. `CONFIG_MODVERSIONS`) with 2.0.* kernels. This causes no problem, even when the modules are used with a kernel that has support for this feature; only that `depmod` wrongly complains about undefined symbols. Ignore the complaints of `depmod` and try to insert the modules despite of these complaints:

```
modprobe zftape
```

If this fails, something is wrong.

<answer from Claus Heine>

2.5 "insmod" says the kernel version is wrong

The `insmod` program can check the kernel version against the version that *Ftape* was compiled for in two ways: It can directly compare the kernel version number recorded in the *Ftape module* against the version of the running kernel, or, if both the kernel and *Ftape* is compiled with versioned symbols, compare the version of the used kernel symbols.

If you have upgraded your version of GCC to v2.7.0 or later, you must recompile the modules utilities with gcc v2.7.x.

Newer versions of `insmod` allows you to "force" insertion of a module into the kernel, even though the version string is incorrect.

<from the Ftape-Howto>

2.6 "insmod" says that kernel 1.2.0 and 1.2.0 differ

Did you remember to apply the `ksyms.c` patch to the kernel? If not, read the `README.linux-1.2` file in the source distribution.

<from the Ftape-Howto>

2.7 Trying to compile Ftape gives me the error "modversions.h: no such file or directory"

The `modversions.h` file is created when the kernel is compiled with the configuration item `CONFIG_MODVERSIONS` turned on. With this option enabled, the file will be created during the `make dep` step.

One more handy tip is that a `make mrproper` will remove `/usr/include/linux/modversions.h`. You will need to reconfig the kernel and do a `make dep` to get the file back.

<from the Ftape-Howto>

2.8 What is this versioned symbols stuff anyway?

When you say 'yes' to `CONFIG_MODVERSIONS` during '`make config`', all the symbols exported by the kernel, i.e: the symbols that the loadable modules can "see", are augmented to include a checksum across the types of the call/return parameters. This allows `insmod` to detect whether the definition of a variable or function in the kernel has changed since the time when *Ftape* was compiled.

This ensures a high degree of safety, such that you do not crash the kernel because you used an outdated module with your kernel.

If you enable `CONFIG_MODVERSIONS` in the kernel, make sure you have

```
-DMODVERSIONS -include /usr/include/linux/modversions.h
```

uncommented in the MODULE_OPT line in the *Ftape* Makefile. Conversely, if you do not have CONFIG_MODVERSIONS enabled, make sure you have it commented out.

<from the Ftape-Howto>

2.9 I seem to be getting sftape instead of zftape. When I run "ftmt status" command, I get output that the Ftape docs says corresponds to sftape (/dev/qft0: Invalid argument). Why?

There are (at least) two possible sources of the problem:

- All Ftape-3.* versions prior to 3.04 install the modules into

  ```
  /lib/modules/misc
  instead of
  /lib/modules/uname -r/misc
  ```

  As `modprobe` searches in `/lib/modules/misc/` last there might be an old `ftape.o` module floating around in `/lib/modules/`*uname -r*`/misc` which `modprobe` finds first (`uname -r` stands for the kernel version). Remove the old `ftape.o` module.

- Your kernel has support for *Ftape* compiled in. Reconfigure your kernel without support for *Ftape* (`CONFIG_FTAPE`) and recompile and install it.

<answer from Claus Heins>

2.10 My Ditto DASH/FC-20/Exabyte Accelerator card works under Microsoft Windows, but I get a drive not found type of error in /var/log/messages when trying to use it under Linux.

You are probably trying to use the same IRQ and DMA settings as your on-board FDC. This does not work in versions of *Ftape* prior to 3.03b. Please update the *Ftape Drivers* to the latest from the Ftape Home Page.

<answer from Tim Jones>

2.11 Ftape DMA transfers gives ECC errors

Sadly to say there are some SVGA cards and Ethernet cards that do not decode their addresses correct. This typically happens when the *Ftape* buffers are in the range `0x1a0000` to `0x1c0000`. Somehow, the DMA write cycles get clobbered and every other byte written gets a bad value (`0xff`). These problems are reported to happen with both SVGA and Ethernet cards. We know of at least one (bad?) ATI 16bit VGA card that caused this.

The easiest solution is to put the card in an 8bit slot (it is often not enough to reconfigure the card to 8bit transfers). Moving the *Ftape* buffer away from the VGA range is only a partial solution; All DMA buffers used in Linux can have this problem! Let us make this one clear: This has nothing to do with the *Ftape* software.

<from the Ftape-Howto>

2.12 Help! I'm getting 'dmaalloc() failed' in my syslog file.

You should only see this is you are trying to `insmod` the `ftape.o` module. Try running `swapout` first. It is provided with the standalone *Ftape* source. It doesn't appear in the *Ftape* source that's provided with the kernel.

Here's an example of how you can set your rc.local file to use it.

```
# Install the Floppy Tape Driver
if [ -f /boot/modules/`uname -r`/misc/ftape.o ]; then
   echo Installing ftape for Linux `uname -r`
   swapout
   insmod /boot/modules/`uname -r`/misc/ftape.o
fi
```

Please note that you won't have this type of problem if you compile the *Ftape driver* into the kernel.

<from the Ftape-Howto>

2.13 Syslogd works overtime when running Ftape

The compile-time options `NO_TRACE` and `NO_TRACE_AT_ALL` in *Ftape* control the amount of system logging. Add whichever is appropriate to the `FTAPE_OPT` line in the Makefile and recompile.

<from the Ftape-Howto>

2.14 How do I change the trace-level?

There are three ways you can do this (in order of personal preference).

While we're at it, here are the meanings of the various trace levels.

- 0 Bugs
- 1 + Errors
- 2 + Warnings
- 3 + Information
- 4 + More information
- 5 + Program flow
- 6 + FDC/DMA info
- 7 + Data flow
- 8 + Everything else

1. Using insmod to change trace-level

   If you are using the modules mechanism to load the *Ftape* driver, you can specify the tracing level as an option to the insmod command.

   ```
   /sbin/insmod ftape.o tracing=<tracing-level>
   ```

2. Using mt to change trace-level

   The *Ftape* driver has a hack in it that allows the `fsr` option in `mt` to be used to set the tracing level. `zftape` does not have this hack.

   ```
   mt -f /dev/ftape fsr <tracing-level>
   ```

   The use of the `fsr` command in `mt` is a *hack*, and will probably disappear or change with time.

3. Recompiling to change trace-level

   The file `tracing.c` contains a line `int tracing = 3;`. Change the 3 to whatever is appropriate and recompile.

<From the Ftape-Howto>

2.15 I'm having problems with Ftape. I'm using the latest version of Ftape from the Ftape Home Page and believe that I've located a real bug. What should I do?

Check the Ftape Home Page. for an even newer version. Then check the FAQ contained in the that package if your problem is listed there. Next, try to check if the manual that comes with the *Ftape distribution* mentions your problem.

There is no need to read the entire manual, simply check the "Concept Index" if it contains a keyword that might be related to your problem, then jump to the proper location in the manual.

If you are still convinced you've found a bug, then post a general question describing the problem to the Linux-Tape Mailing List , but do not attach your entire *Ftape* error-log. If we've seen the problem before, we'll let you know where the resolution effort stands. If we haven't, the Ftape maintainer will most likely request that you send him the entire *Ftape* error-log (snipped from your system messages file).

<answer from Tim Jones>

3 "Using Ftape" related questions !

3.1 How fast is Ftape ?

You can achieve quite respectable backup and restore speeds with *Ftape*: a Colorado DJ-20 and an Adaptec 1542CF controller, has been measured at 4.25Mbyte/min sustained data transfer rate (no compression) across a 70Mbyte tar archive, while comparing the archive on the tape with data on an IDE disk. The speed of *Ftape* is mostly dependent on the data transfer rate of your FDC: The AHA1542CF has a "post-1991 82077" FDC, and it will push 1Mbit/sec at the tape drive. If you have an FDC which can only deliver 500Kbit/sec data rates, you will see half the transfer rate (well, roughly).

3.2 When I write to some of my tapes, they seem to spend a lot of time "shoe-shining," or repositioning instead of streaming. Is something wrong with my system?

There has been a few reports of "shoeshining". This is when the tape just seems to run back and forth endlessly. This has been seen on a Jumbo 250 (74407.3051@compuserve.com) and on an Iomega 250 Ditto Insider (tom@opus.cais.com). In the latter case it has been narrowed own to using an ELF Linux and running off a SCSI hard disk (connected to an Adaptec 1542cf). Please contact me if you have an update to this problem.

<from the Ftape-Howto>

Probably not. If you are backing up a large number of < 2K files, you're just going to have to live with it. In this event, the repositions are caused by file system access overhead. If you are backing up a normal system's files, this may be caused by slop or media stretching in the tape cartridge. By simply retensioning the tape, you should see this go away. Try

```
ftmt -f /dev/zqft0 reten
```

to retension the tape. If retensioning doesn't solve this, and it's only happening on certain tapes, it might be wise to replace the tapes in question.

<answer from Tim Jones>

If you use afio as your backup tool you can set it to write a very large number of buffers in one hit by using the -c flag. Make it large enough so that you supply enough data for most of a single end-to-end pass over the tape. For my system, the following streams quite nicely - stopping relatively few times per tape pass on an unloaded system:

```
find /usr/local -xdev -print | afio -o -v -f -b 10240 -c 800 /dev/qft0
```

In my case I'm writing 800 x 10240 bytes per tape write, i.e. about 8MB. haven't experimented that much with these settings - so someone might like to establish more optimal ones.

Presumably other backup utilities could be modified to use a similar technique.

<answer by Michael Hamilton>

GNU tar doesn't use buffering in this way. The commercial backup program "bru" is able to multi-buffer using shared memory; this works only when writing compressed archive with bru (regardless whether you use *Ftape's* builtin compression)

Another way to overcome the problem might be to use more dma buffers in the *Ftape kernel driver* like:

```
mt -f /dev/qft0 setdrvbuffer \$((6\mch{\ast}32786))
```

$((6*32786)) should be expanded by your shell when using a Bourne compatible one. This has a negative impact on the system's memory pool: *Ftape's* dma buffers cannot be used by any other part of the kernel nor by any other application. And kernel memory cannot be swapped out. If you decide to use this kind of multi-buffering then you should unload the driver as soon as it isn't needed any longer.

<answer by Claus Heine>

3.3 Do I have to reboot to the DOS world to format tapes?

Not if you are using the latest version of the *Ftape drivers* from the Ftape Home Page. To format a QIC-80, TR-1, TR-3, QICWide 3010 or 3020 tape, make sure that the *Ftape-format module* is loaded and use the special version of the mt utility included with *Ftape* in the contrib/gnumt directory. Use the `ftformat` option with an optional argument that is interpreted as follows:

ftmt ftformat 0

keep the existing header information, DON'T erase the tape prior formatting it.

ftmt ftformat 1

keep the existing header information and erase the tape prior formatting it.

ftmt ftformat 2

DON'T keep the existing header information, DON'T erase the tape.

ftmt ftformat 3

DON'T keep the existing header information, but erase the tape prior to formatting it.

"ftformat" without parameter is synonymous to "ftformat 1" NOTE: Do not try to format Ditto 2GB tapes.

<answer from Tim Jones>

NOTE: The formatting ioctl interface as well as the user interface have been completely rewritten for *Ftape-4.0*. Further information will be included in the manual of *Ftape-4.0*.

<answer from Claus Heine>

3.4 Ftape detects more bad sectors than DOS on QIC-3020 tapes

If you look at the difference, you will notice that *Ftape* always detects 2784 sectors more than DOS.

The number that *Ftape* reports is correct (of course `:-)`. Each correctly formatted QIC-3020 tape has 2784 sectors at fixed positions that are marked in the bad sector map. To quote from the specs:

> Tracks 5,7,9,11,13,15,17,19,21,23,25 and 27 within 4 segments of either EOT or BOT are prone to increased error rates due to hole imprints. Therefore, these regions shall be mapped as bad at format time and entered in the bad sector map by indicating that all sectors within the identified segments are bad.

This gives 12 tracks ∗ 2 ∗ 4 segments ∗ 29 sectors == 2784 sectors.

So *Ftape* choose to report the real number of sectors that cannot be used on the tape, while DOS gives a more optimistic number giving a better indication of tape quality. (*Ftape's* behavior might change in the future to detect correct formatting and display the separate numbers. It has rather low priority though).

QIC-3010 are alike QIC-3020 tapes regarding this.

<from the Ftape-Howto>

3.5 Is it ok that I'm not hearing the tape move when I do a fsf or a bsf with mt?

Yes. The driver merely updates an internal counter when those commands are issues. The tape should move to the proper location on the next read or write access to the tape drive.

<from the Ftape-Howto>

3.6 Why does my XYZ backup program complain about "Invalid argument" errors?

`zftape` requires the data to be written in multiples of a fixed minimal block size. This is a very usual behavior for a tape device. There are three ways to get rid of those errors:

- set *Ftape's* block size to the block size used by the backup program. The example below works for "afio":

  ```
  mt -f /dev/qft0 setblk 5120
  ```

- If you don't want to use *Ftape's* built in compression you can also use

  ```
  mt -f /dev/qft0 setblk 0
  ```

  to switch *Ftape* to variable block size mode and be able to write the data in arbitrary portions to the tape (BUT: the builtin compression doesn't work with this setting). When you intend to use "KBackup" then this is the only way to make it work together with *Ftape* (it _may_ work, don't know if it does)

- tell your backup program about *Ftape's* default block size of 10k (which is also the default of GNU tar). For "afio" you can use the following command line switch:

  ```
  afio -b 10k ...
  ```

You may want to read the section "Tape blocks" of the manual (use its "Concept index" to directly jump to that section)

<answer by Claus Heine>

3.7 Why do I get "/dev/qft0: No such device" errors?

I assume that the following is the problem: The *Ftape* module is loaded OK into the kernel:

```
/usr/src/ftape-3.03b-970603# lsmod
 Module         Pages    Used by
 ftape             22           0
```

but then this happens:

```
\$ ftmt -f /dev/qft0 status
ftmt: /dev/qft0: No such device
```

Solution You need to load the *zftape.o* module as well. With Ftape-3.∗ the *ftape.o* module doesn't implement the VFS interface. This is done by *zftape.o*.

<answer from Claus Heine>

3.8 I get "device busy" when I make multiple backups on a tape using some script.

The "device busy" messages can only occur while the *Ftape devices* are still held open by some program. As soon as the close() system call has completed the busy flag is cleared. May be "bru" or some other program has still forked off a child that dies delayed?

Yes, this will reproduce the problem, it seems:

```
tar -cvvzf /dev/nqft0 --block-compress ; mt rewind
```

You can skip the "–block-compress" if using the most recent version of GNU tar.

However, this is not a bug of *Ftape*. It seems that the parent tar process exits before its child has closed the tape device. I know, however, from hacking the tar code ages ago, that tar properly waits for its parent to die.

However, the busy message simply means that the "busy" variable is still held at 1 (zftape/zftape-init.c). And this simply means that there still is a process hanging around that holds the tape device open.

I think I have it (only for the case of tar 'cause I have the source code.

If on uses tar with compression, then it forks a child which will become the compressor bei execing "gzip" or whatever. Before the call to execlp() the child will fork off a grand child of its parent tar. That grandchild will do the actual tape I/O.

```
tar - fork() - write to child tar
        |
      child tar - fork() - gzip (will pipe to grand child tar)
                    |
                  grand child tar - open archive.
```

Now, parent tar only waits for its child to die. gzip surely doesn't wait for the grand child as the gzip is a result of an execlp().

What I don't know is whether the grand child should be implicitly waited for by the parent tar, or if the wait() function also waits for grand childs.

But this seems to be the problem: the parent tar already has exited while its grandchild still is busy closing the archive. One hardly will notice this problem if the close() happens fast (i.e. regular files, block devices, also other tape devices?), but it isn't a bug in *Ftape*, but either in the backup programs or in the kernel or maybe libc exit code.

Don't know if the considerations above also apply to bru. If there is no grandchild and the parent process properly waits for its childs then there shouldn't be a problem.

<answer from Claus Heine>

3.9 How do I "...." with tar?

These are really `tar` questions: Please read the `man` page and the `info` page. If you have not got it either, try

```
tar  --help 2>\&1 | less
```

If your version of `tar` is v1.11.1 or earlier, consider upgrading to v1.11.8 - This version can call `GNU zip` directly (i.e.: it supports the `-z` option) and has an elaborate help included. Also, it compiles right out of the box on Linux.

<from the Ftape-Howto>

3.10 What block-size should I use with tar ?

When using compression, and in all general, it can be a benefit to specify to `tar`, that it should block the output into chunks. Since *Ftape* cuts things into 29Kbyte blocks, saying '`-b58`' should be optimum.

"Why 29Kbyte?", I hear you cry. Well, the QIC-80 standard specifies that all data should be protected by an Error Correcting Code (ECC) code. The code specified in the QIC-80 standard is known as a Reed-Solomon (R-S) code. The R-S code takes 29 data bytes and generates 3 parity bytes. To increase the performance of the ECC code, the parity bytes are generated across 29 1Kbyte sectors. Thus, *Ftape* takes 29Kbytes of data, adds 3Kbytes of ECC parity, and writes 32Kbytes to the tape at a time. For this reason, *Ftape* will always read and write 32K byte blocks to be able to detect (and correct) data errors.

If you are curious, and wish to know more, look in the `ecc.c` and `ecc.h` files, for an explanation of the code and a reference to a textbook on Reed-Solomon codes.

<from the Ftape-Howto>

3.11 Where can I find the tar/mt/cpio/dd binaries - sources - manpages?

All of these tools have been developed by the GNU project, and the source (and man page) can be fetched from just-about any ftp site in the world (including `ftp.funet.fi`, `tsx-11.mit.edu`, and `sunsite.unc.edu`). In any case they can be fetched from the official GNU home site: `prep.ai.mit.edu [18.71.0.38]:/pub/gnu`. The latest versions (as of September 12 1996) are:

```
cpio:   2.4.2 (cpio-2.4.2.tar.gz)
dd:     3.13 (fileutils-3.13.tar.gz)
mt:     2.4.2 (cpio-2.4.2.tar.gz)
tar:    1.11.8 (tar-1.11.8.tar.gz)
gzip:   1.2.4 (gzip-1.2.4.tar.gz)
```

They all compile out of the box on Linux `v1.0.4` / `libc v4.5.19` / `gcc v2.5.8`.

<from the Ftape-Howto>

3.12 If I use tapers compression, is it a bad idea to use the compression with zftape, or would it be better to not use tapers compression, and let zftape do it?

It is not bad as such to compress data twice (which would be the case when using tapers compression together with *zftape's* compression) but it doesn't make any sense. You won't gain much further compression, but only waste CPU cycles.

Tapers compression should be quite safe, as taper compresses single files; in contrast to *tar -czf ...* which makes the entire data stream a large compressed block of data, which is really a bad thing with serious backups as a single bad byte at the beginning of the archive can make the entire archive unusable, well, it will be at least quite difficult to recover.

<Answer from Claus Heine>

3.13 How does zftape compression compare to say gzip -9?

gzip -9 is better (i.e. one gains higher compression). *zftape's* compression is comparable with the Un*x *compress* program, but should be faster, and is faster than *gzip*.

<Answer from Claus Heine>

3.14 I don't trust compression, but hear that the sftape interface is going away. What should I do?

Use the *zftape interface*, but don't load the *zft-compressor module*. The device then becomes `/dev/qft0`.

<answer from Tim Jones>

3.15 Ftape says "This tape has no 'Linux raw format"

You get this complaint if you haven't *erased* your freshly formatted tape. This is because *Ftape* expect a "magic header" on the tape, to be able that it is allowed to interpret the header segment in its own way (eg: file marks). To remove the problem, say

```
mt -f /dev/nftape erase
```

<from the Ftape-Howto>

3.16 Can I exchange tapes with someone using DOS?

No. The DOS software conforms to the QIC-80 specs about the layout of the DOS filesystem, and it should(?) be a small problem to write a program that can read/write the DOS format. In fact, I'd bet that creating a nice user interface would be a bigger problem.

<From the Ftape-Howto>

3.17 How does 'mt eom' work when you've started overwriting a tape in the middle?

(EOM is "End Of recorded Media", the position right after all data already recorded to the tape)

One cannot use tape "files" like files on an ordinary file system.

In principle, a tape doesn't allow anything but appending new data at EOM. However, if one positiones just in the middle of the already recorded data AND starts writing, then the driver first deletes all following files (thus moving the EOM to the actual position) and then starts writing.

Thus, the new EOM after finishing the write process, is then after the newly recorded data.

One of the consequences of the above is, of course, that writing to the tape in the middle of the already recorded area, is destructive in the sense, that it not only overwrites the "file" the tape is positioned at, but also deletes all following files.

<from the Ftape-Howto> <Answer from Claus Heine>

3.18 When I made backups before using taper, under the 2.0.29 ftape my drive didn't support fsf, under the new zftape it does, why would this be, and what exactly is fsf ?

It probably didn't work before because you didn't use a

```
mt -f /dev/rft0 erase
```

before writing data to the cartridge. THIS ISN'T necessary any more.

But, hey, what does *mt fsf*? Tape drives don't store files in the sense that you can use

```
cp somefile /dev/my_what_ever_tape
```

or be able to mount the tape drive like you could mount a harddisk. You can't do nothing with a tape drive but write data to it in a sequential manner.

As this is quite inconvenient, somebody invented something which is known under the name *file mark* or *eof mark* (eof == End Of File). Those marks don't separate files that have been backed up to the tape device, but only separate blocks of data (whatever data that might be).

Normally, the kernel tape device drivers take care of writing file marks when the tape device is closed, i.e.

```
tar -cf /dev/nqft0 /bin
tar -cf /dev/nqft0 /etc
mt -f /dev/nqft0 rewind
```

would result in a backup of all files under */bin* and */etc*. When the first *tar* finishes, the kernel driver will take care of writing a file mark to the tape at the the current tape position, and when the second *tar* process has finished, another file mark is written to the tape cartridge at that position.

Now, the sense of those file marks is, that it is possible to skip between different archives on the tape more quickly than would be possible with reading the data back.

The commands to do that are:

mt fsf

fast skip to the next file mark towards EOT (End Of Tape)

mt bsf

fast skip to the next file marks towards BOT (Begin Of Tape)

Thus, to extract the second archive in the example above, one doesn't need to read the first archive back, but can proceed as follows:

```
mt -f /dev/nqft0 rewind
mt -f /dev/nqft0 fsf
tar -xvf /dev/nqft0
```

<Answer from Claus Heine>

3.19 What exactly is the difference between ftape, and zftape?

When *Ftape* was young there were two versions of the floppy tape driver, one of them was called *zftape* because of its built-in user-transparent on-the-fly compression. Whether such a thing is a feature or a bug ('cause this needn't be done in kernel space) is another question. However, the ioctl interface and file mark handling provided by *zftape* was much better and had less bugs. And *zftape* allows to use floppy tape cartridges with different OS. Well, you can't exchange data, but *zftape* won't overwrite volumes created by your Windoze program, and vice versa.

Nowadays, *Ftape* is name of the entire floppy tape driver package AND *ftape.o* is the file-name of the kernel module that implements the low-level hardware support. *zftape* has ceased to exist as a separate package, but the new *Ftape* versions (since ftape-3.00) contain a *zftape.o* module that needs to be loaded on top of *ftape.o* (i.e. you need to load BOTH modules to be able to access your floppy tape drive) and implements the file system interface and the advanced (?) features of the previous verions *zftape*.

<Answer from Claus Heine>

3.20 What is the difference between a rewinding, and non rewinding drive?

Well, the rewinding tape devices rewind the tape to BOT (Begin Of Tape) when the device is closed, i.e.

```
tar -cvf /dev/qft0 /bin
```

will rewind the tape cartridge when the tar job has finished. In contrast,

```
tar -cvf /dev/nqft0 /bin
```

will NOT rewind the tape cartridge and leave the tape R/W head at its current position.

Rewinding devices should be used when performing a single backup, non-rewinding devices can be useful when doing multiple backups as one doesn't need to space to EOM (End Of recorded Media) before appending another archive.

Non-rewinding devices MUST be used when sending any of the tape motion command to the tape drive, such as

```
mt -f /dev/nqft0 fsf
```

, because when the *mt* process finishes then the tape device is closed which would result in rewinding the cartridge with the rewinding devices.

<Answer from Claus Heine>

3.21 Can someone tell me how to use mt to rewind my TR-3 drive one record using zftape record, so I can verify it?

Well, it depends. If the tape is still positioned inside the volume just written, "mt bsf 1" (or equivalently "mt bsf") will backspace just to the beginning of that volume (this is how "tar –verify" works). If the tape is already positioned AFTER the filemark that marks the end of the last written volume, then you need to issue "mt bsf 2"

The logic behind this is as follows: "MTBSF #count" backspaces over #count file marks, stops, and then positions on the `EOT` side of the last skipped file mark. This means, an "mt bsf 2" will position right at the beginning of the previous volume.

<answer form Claus Heine>

3.22 By non-rewinding, they mean that it doesn't automatically rewind, correct? It doesn't mean that under no circumstances it iwill rewind, right? I tried using /dev/zqft0, and it instantly rewinds the tape.

You are right: auto-rewind means, the tape is rewound when the tape device is closed, non-rewinding means, the tape isn't automatically rewound when the tape device is closed (but you can, of course, use the tape motion commands BSF/FSF etc. to position the tape head at every position you like).

<answer form Claus Heine>

3.23 What is the difference between what mt considers a record and what it considers a file?

A record is the minimal amount of bytes that will be accepted by the tape in one read/write operation (except in "variable block size mode" where it just should be the amount of data actually written in a single write operation??).

For *zftape* every read and write access has to be a multiple of a fixed block size (fixed, but tunable with `MTSETBLK`). This block size is a "tape record" (as mentioned in the GNU mt man page and defaults to 10kb for *zftape*.

A "file" (in the sense of the mt man page) is a, well, misleading terminus. What is meant is an area of the tape between two file marks. This is not a file like a file on the file system, in the sense that it could have a name, file access modes, could be moved or copied with cp, mv, rm etc.

Instead, It simply is the area of the tape that was recorded in one backup session, its end is marked by a tape file mark, and its beginning is delimited by either `BOT` or the file mark of the previous tape "file". That tape "files" are the things that can be skipped with the `MTBSF/FSF` commands.

<answer form Claus Heine>

3.24 This script implements a simple contents listing for the zftape package using the "MTIOCVOLINFO" ioctl.

Here is as little perl/bash script that lists the contents of a cartridge using the *zftape* specific "volinfo" ioctl. Hope this shows how to handle this kind of stuff.

What it basically does is the following:

1. Rewind the cartridge
2. Issue the volinfo command:

```
claus@thales:~\$ mt volinfo
file number         = 1
block size          = 10240
physical space used =  522.0 kilobytes
real size of volume =  520.0 kilobytes
```

Parse the ouput and place the values in appropriate variables

3. Skip to the next volume with "mt fsf"
4. Exit if this gives an error (EOD), otherwise "goto 2)"

bf/The Perl Script/

```
#!/usr/bin/perl
#
#     Copyright (C) 1997 Claus-Justus Heine
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   This script implements a simple contents listing for the zftape
#   package using the MTIOCVOLINFO ioctl.
#

$version = <<EOT;
listtape-1.0 -- a perl script to list the contents of a floppy tape cartridge
under Linux using the zftape driver

RCS \$Revision: 1.2 \$
RCS \$Date: 1997/09/11 00:16:28 \$
EOT

$tapedev = "/dev/tape";
$usage = <<EOT;
Usage: listtape [options ...]

Mandatory or optional arguments to long options are mandatory or optional
for short options too.
```

```
-f, --file=FILE       Tape device to use. Default is  "/dev/tape".
-h, --help            Print this help.
-?                    Same as '-h'.
--usage           Same as '-h'.
-V, --version         Print version information.

Author: Claus-Justus Heine <claus\@momo.math.rwth-aachen.de>
EOT

while ($ARGV[0] =~ /^-/) {
$_ = shift;
if (/--file/) {$_ = shift; $tapedev = $_; next;}
if (/-f/) {$_ = shift; $tapedev = $_; next;}
if (/--help/) { print $usage; exit 0; }
if (/-h/) { print $usage; exit 0; }
if (/--usage/) { print $usage; exit 0; }
if (/-\?/) { print $usage; exit 0; }
if (/--version/) { print $version; exit 0; }
if (/-V/) { print $version; exit 0; }
die $usage;
}

\&open_tape($tapedev, "status");
while(<FTMT>)
{
$online = 1 if (/.*online.*/);
}

if (! $online) { die "No cartridge present.\n"; }

\&mtop($tapedev, "rewind");

printf "%11s%12s%20s%20s\n",
"file number", "block size", "volume size", "tape space";

while (1)
{
\&open_tape($tapedev, "volinfo");
while (<FTMT>) {
if (/^file number\s*=\s*([0-9]*)$/) { $filenumber = $1; }
if (/^block size\s*=\s*([0-9]*)$/) { $blocksize = $1; }
if (/^physical space used\s*=\s*([[0-9]*.*)/) { $rawsize = $1; }
if (/^real size of volume\s*=\s*([[0-9]*.*)/) { $size = $1; }
```

```
}
close(FTMT);
if (\&mtop($tapedev, "fsf 1") != 0) {
\&mtop($tapedev,"rewind");
print "\nRemaining space: $rawsize\n";
print "Tape block size: $blocksize\n";
exit 0;
}
printf "%6d          %5d  %20s%20s\n",
    $filenumber, $blocksize, $size, $rawsize;
}

sub mtop
{
local ($tape, $operation) = @_;
local ($exitval);
system "ftmt -f $tape $operation > /dev/null 2>\&1";
}

sub open_tape
{
local ($tape, $operation) = @_;
local ($command);

$command = "ftmt -f " . $tape . " " . $operation . " |";
open(FTMT, $command) || die "Couldn't open $command -- $!\n";
}
```

The Basch Script

```
#! /bin/bash
#
#     Copyright (C) 1997 Claus-Justus Heine
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
```

```
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   This script implements a simple contents listing for the zftape
#   package using the MTIOCVOLINFO ioctl.
#

#
# insert better option parsing here
#
TAPEDEV=${1-/dev/tape}

if ! echo $TAPEDEV | grep "/dev/n"
then
TAPEDEV=/dev/n$(basename $TAPEDEV)
fi

if ! [ -c $TAPEDEV ]
then
echo $TAPEDEV is not a character device!  1>\&2
exit 1
fi

if ! mt -f $TAPEDEV rewind
then
echo Could not rewind $TAPEDEV - no cartridge present?  1>\&2
exit 1
fi

echo -e "\nContents of $TAPEDEV:\n"

printf "%11s%12s%20s%20s\n" "file number" "block size" "volume size" "tape space"

trap "rm -f /tmp/$0.$$" exit

while true
do
if ! foo=$(mt -f $TAPEDEV volinfo |cut -f 2 -d =)
then
echo $TAPEDEV doesn\'t seem to be a floppy tape device 1>\&2
exit 1
fi
#
```

```
# "echo foo | read foo" will not work as the "read foo" is executed in
# another shell.
#
echo $foo > /tmp/$0.$$
read file blksz used usedunit size sizeunit < /tmp/$0.$$
if ! mt -f $TAPEDEV fsf 1 > /dev/null 2>\&1
then
echo -e "\nRemaining space: $used $usedunit"
echo -e "Tape block size: $blksz"
if ! mt -f $TAPEDEV rewind
then
    echo Rewind of $TAPEDEV failed 1>\&2
    exit 1
fi
exit 0
fi
printf "%6d          %5d  %20s%20s\n"\
$file $blksz "$size $sizeunit" "$used $usedunit"
done
```

<answer from Claus Heine>

4 "Tape and Drivers" related questions !

4.1 What are good makers of Travan tapes?

I was the UNIX Product Manager at Archive Corp (Prior to the Conner/Seagate mess) and we performed extensive tests of tape media for compatibility certification, including retentivity, flaking and length consistancy. Based on the results of the tests, we selected the best of these certified manufacturers' products to private label as our own media. Here is the order in which we selected vendors up through 1995 (when I lost contact with the ATI group):

QIC

1. 3M (now known as Imation)
2. QMaxell/Sony (tie)
3. (BTW - Iomega uses Sony private-labelled media)

4MM

1. Fuji
2. Maxell/Sony (tie - is this a trend?)

8MM

1. Fuji/Exabyte - which we believed to be OEM'd Fuji (tie - so much for trend!)
2. Sony
3. Maxell

DLT

1. Maxell
2. Sony

Otherwise, we had entries in our search from other vendors which were generally a private-labelled version of one of the major labels above. The exceptions were Verbatim and DIC. Both of these manufacturer's media had fall-out rates and length discrepancies that were so high that we would not certify them and even warned customers about them indicating that we could not offer any sort of guarantee that they would get a good backup using the media from these manufacturers.

In addition, since coming to EST, I've found that Verbatim media is still not worth the money saved in purchasing it. We have 11 of their TR-Extra and QIC-Extra (QICXL) tapes that were useless after fewer than 20 passes each.

While this is my personal opinion, it is based on over 9 years of experience with this very question, I strongly recommend Imation/3M media for QIC/Travan user, Fuji media 4MM users, Exabyte/Fuji for 8MM and DEC labelled media for DLT users.

<answer from Tim Jones>

4.2 Where can I obtain the QIC standards?

If you wish to help developing *Ftape*, or add some utility (e.g. a tape formatting program), you will need that appropriate QIC standards. The standard(s) to get is: QIC-80, -117, -3010, and 3020. QIC-117 describes how commands are sent to the tape drive (including timing etc), so you would probably never need it. QIC-80/3010/3020 describes higher level part, such as tape layout, ECC code, standard filesystem. You can get the QIC standards from the following address:

```
Quarter Inch Cartridge Drive Standards, Inc.
311 East Carrillo Street
Santa Barbara, California 93101
Phone: (805) 963-3853
Fax:   (805) 962-1541
```

Note: They are registered as 'Freeman Associates, Inc' in the phone book.

<from the Ftape-Howto>

4.3 Is the Iomega Ditto 2GB drive supported?

Yes, if you are using the latest version ftape-3.03b-970603 of the *Ftape drivers* from the Ftape Home Page.

<answer from Tim Jones>

As the Ditto 2GB is a Tr-3 tape (though it can only store 1GB instead of the 1.6GB you get with a regular Tr-3 drive) you need an FDC (FDC means: Floppy Disk Controller) that is capable of at 1Mbit/sec transfer rate. You don't need to worry about this if you have an accellerator card (i.e. the Ditto Dash controller). Otherwise try to purchase an FDC that claims to be capable of driving 2.88Mb floppies because this implies that the FDC is capable of 1Mbit transfer rate.

Ftape prints the maximum data rate of the FDC to the kernel log files like this:

```
ftape-ctl.c (ftape_init_drive) - Highest FDC supported data rate: 500 Kbps.
```

<answer from Claus Heine>

5 Miscellaneous !

5.1 How to subscribe to the Ftape Mailing List?

You can subscribe to that list by sending mail to

```
majordomo@vger.rutgers.edu
```

with the single line

```
subscribe linux-tape
```

in its body. Please store the answer you get from majordomo in a safe place because it contains instructions how to `UNSUBSCRIBE` from the mailing list.

<answer from Claus Heine>

5.2 How to un-subscribe from the Ftape Mailing List?

Send mail to

```
majordomo@vger.rutgers.edu
```

with the single line

```
unsubscribe linux-tape MY@EMAIL.ADDRESS
```

where `MY@EMAIL.ADDRESS` has to be replaced by the the email address that you used when subscribing to the list. Note that you must have received an email with instructions how to unsubscribe from the mailing list at the time you subscribed to it.

<answer from Claus Heine>