Linux Ftape HOWTO

Kai Harrekilde-Petersen, <khp@dolphinics.no> v1.8, 22 May 1996 for ftape-2.08

This HOWTO discuss the essentials of the do's and dont's for the `ftape` driver under Linux. The `ftape` driver interfaces to QIC-40, QIC-80, QIC-3010 and QIC-3020 compatible drives only. The QIC-3010 and QIC-3020 standards are also known as 'Travan' (TR-2 and TR-3). These drives connects via the floppy disk controller (FDC). It **does not** cover SCSI or QIC-02 tape drives. DAT tape drives usually (always?) connect to a SCSI controller. This is but one of the Linux HOWTO documents. You can get an index of the HOWTOs from *the Linux HOWTO index* <http://sunsite.unc.edu/mdw/HOWTO>, while the real HOWTO's can be fetched (using `ftp`) from `sunsite.unc.edu:pub/Linux/doc/HOWTO` (this is the "official" place) or via the World Wide Web from *the Linux Documentation Project home page* <http://sunsite.unc.edu/mdw/linux.html>.

Contents

1 Legalese

2 News flash

This isn't exactly news, but anyway: The ascii and HTML versions have some of the section headers permuted. This is a feature of the SGML system that the LDP project uses. I can't help it. It even seems that the HTML version like to repeat some of my section headers once in a while - luckily, the contents should be the same. In the GNU info version, you cannot select the section "(Un)supported drives", because info thinks that "(un)" refers to a file.

version 1.8 (May 22, 1996)

- Copyright policy changed to GNU GPL v2
- The maintainer's email address has changed.
- Updated to ftape-2.08
- `ftape` is now a part of the kernel distribution.

version 1.7.1 (February 13, 1996)

- Updated to ftape-2.06b

version 1.7 (January 28, 1996)

- Updated to ftape-2.06 and modules-1.3.57

version 1.6.2 (January 23, 1996)

- Connor TST3200R drive added
- Updated 2Mbps fdc information.

version 1.6.1 (January 16, 1996)

- minor corrections

version 1.6 (January 10, 1996)

- New maintainer of `ftape`
- updated to v2.05
- added new drives

It has been quite some months since I last updated the HOWTO. This is due to several reasons, but mostly because I have moved from Denmark to Norway due to a job change (this meant that my Linux box was stored away for nearly two months. ugh). Please note the new email address: `khp@dolphinics.no`. Until June 1st, I will still get the mail sent to my old address, but after that day, email to the old account will no longer reach me.

3 The preliminaries

Note that I (the howto-maintainer) no longer use `ftape` myself, so I cannot give much up-to-date advice on what hardware to buy. If you have a problem, try posting on `comp.os.linux.help`, or to the tape discussion group `linux.dev.tape`. This is a Usenet group that mirrors the traffic on the mailing list `linux-tape@vger.rutger.edu` (see 4.3 below). I encourage reading and posting on the newsgroup, since the `vger` machine is overburdened with the load of the Linux mailing lists.

You should try to post a summary of your problems and its solution(s), after you've got it working, even if you only got it partially working. Please also send me (<`khp@dolphinics.no`>) a copy of your solution so that I can add it to the HOWTO.

I read my mail daily, I try to respond to everyone, but I cannot guarantee that I will respond immediately. I usually read the newsgroups (`linux.dev.tape` and the `.kernel` list), but this is subject to the workload at my for-pay job.

If you recieve this as part of a printed distribution or on a CD-ROM, please check out *the Linux Documentation home page* `<http://sunsite.unc.edu/mdw/linux.html>` or ftp to `<ftp://sunsite.unc.edu:/pub/Linux/doc/HOWTO>` to see if there exists a more recent version. This could potentially save you a lot of trouble.

3.1 What is `ftape`

`ftape` is a driver program that controls various low-cost tape drives that connect to the floppy controller.

`ftape` is not a backup program as such; it is a device driver, which allows you to use the tape drive (just like the SoundBlaster 16 driver let you use your sound card) through the device files `/dev/[n]rft[0-3]`.

`ftape` is written by Bas Laarhoven <`bas@vimec.nl`>, with "a little help from his friends" to sort out the ECC (Error Correcting Code) stuff. `ftape` is copyrighted by Bas under the GNU General Public License, which basically says: "go ahead and share this with the world, just don't disallow other people from copying it further".

ftape is quite stable, and has been that for some time now. Most changes is to support the new QIC-3010 and QIC-3020 (aka Tranvan) tape formats and new tape drives. It is reliable enough for critical backups (but always remember to check your backups, so you won't get a nasty surprise some day).

`ftape` supports drives that conform to the QIC-117 and one of the QIC-80, QIC-40, QIC-3010, and QIC-3020 standards.

`ftape` support neither QIC-02, IDE (ATAPI), nor SCSI tape drives. SCSI drives are accessed as `/dev/[n]st[0-7]` and are supported by the kernel through the SCSI drivers. If you look for help on SCSI tape drives, you should read the `SCSI-howto`. ATAPI tape drives are supported by the kernel since 1.3.46. See section 5.1 and 5.3 for a list of supported and unsupported drives.

4 Getting and installing `ftape`

4.1 Getting `ftape`

You can get `ftape` from many ftp sites in the world. The two main sites are:

- `ftp://ftp.mcc.ac.uk/pub/linux/misc/ftape/`
- `ftp://sunsite.unc.edu/pub/Linux/kernel/tapes/`

`sunsite` is the official place to look for `ftape`.

Dr. Owen LeBlanc has been kind enough to provide me with access to Manchester's public ftp server, so I can drop ftape code directly into the `.../ftape` directory.

Due to my job and email change, the ftp area I have on `www.pip.dknet.dk` will go away my June 1st, 1996. I will not upload new versions to `pip` anymore for that reason.

I will eventually include an installation guide in this section. For the moment being, you'll have to do with the document that is included in the `ftape` distribution.

4.2 What you need to install `ftape`

There are three source distributions that you must have to get `ftape` running:

- `ftape` v2.08
- `modules` v1.3.57 or later
- Linux kernel v1.2.13 or v1.3.x (preferably 1.3.74 or later)

Even if you have a v1.2 kernel, you should use the modules-1.3.57 package, not the modules-1.2.8 package (Bjørn Ekwall, maintainer of the modules package, encourages this).

4.2.1 The `modules` utilities

The modules packages can be found on `tsx-11.mit.edu` and `sunsite.unc.edu`. You must compile and install it before you compile the kernel.

What's new in `modules-1.3.57`? As of kernel version 1.3.57, "kernel daemon" (kerneld) support is included in the standard kernel. `kerneld` can automatically insert and remove the neccessary modules to your kernel, at runtime. To be able to utilize this, you must have the `modules-1.3.57` package, which contains the `kerneld` daemon.

WARNING! If you use `ftape`'s non-rewinding devices (`/dev/nrft[0-3]`) you should *never ever* use `kerneld` to automatically insert/remove `ftape`. Failing to observe this may cause you to loose the data on the tape.

Installing the `modules` utilitles Extract the files somewhere convienient. Then, go into the `modules-1.3.57` (or `-1.3.69`) directory and say `make all; make install`. Remember that you must be `root` to be able to install the files and man pages.

If you wish to enable certain users (e.g. yourself, as non-root) to insert/remove modules without having to log in as root, you can create a new group (I call it `trust`), change the group of `insmod` to `trust` and the permisions to `4710` (i.e: suid root, the group have execute rights, but 'other' have no rights). This is, ofcourse, a security, since any user who is member of the `trust` group will be able to insert anything into the kernel and by careful coding give herself `root` rights (you have been warned). However, if you do trust the peploe in the `trust` group, this is a convenience. I have just myself in the `trust` group, and uses this trick because I prefer working as a normal user, but need to insert/remove the `ftape.o` module while working on it.

4.2.2 The Linux kernel

Since Linux version 1.2 has been out for some time I assume that everyone has switched over to it (or v1.3.x for that matter). If you have not already switched over, I assume you have a very good reason for not doing so, and that you can cope with the differences in installation etc, that it will make for you. If you wish to know what version the latest kernel is, simply try `finger @linux.cs.helsinki.fi`.

The kernel can be fetched from a large number of sites all over the world, including these:

```
sunsite.unc.edu      [152.2.22.81]:    /pub/Linux/kernel/tapes/
tsx-11.mit.edu       [18.172.1.2]:     /pub/linux/sources/system/
ftp.funet.fi         [128.214.248.6]:  /pub/OS/Linux/PEOPLE/Linus/
ftp.cs.helsinki.fi   [128.214.48.123]: /pub/Software/Linux/Kernel/
```

You will find a number of subdirectories, including two named `v1.2` and `v1.3`. These contain (you guessed it!) `v1.2` and `v1.3` of the kernel.

The `v1.3` is the development kernel branch, while the `v1.2` series are updated rather seldomly, and only after some serious bug has been found in it, and a field-tested fix has been provided.

For a minimum-hazzle kernel, you should go for the `v1.2` series, but if you are more adventerous (or the `v1.2` kernels lack your favorite CDROM driver), you can get a `v1.3` kernel.

Be warned, though, that the `v1.3` kernels are of unknown stability, and you should be well fared in patching, configuring, and compiling the kernel.

Quick guide to compiling the kernel The 'standard' (recommended) place to install the kernel source is `/usr/src/linux`.

Go to `/usr/src` and type `tar xzf linux-1.2.13.tar.gz` (or whatever kernel version you intend to compile). This will create the `linux` directory (and a lot of subdirectories), which holds the kernel source code.

```
# cd /usr/src
# tar xzf linux-1.2.13.tar.gz
```

Go into the `linux` directory and configure the kernel.

```
# cd linux
# make config
```

The configuration script will ask you about which drivers you wish to include, and what functionality (networking, filesystems, etc) that you want. If you use a kernel version later than v1.3.34, there will be NO ftape option. This is because ftape has been taught to allocate its dma buffers at insmod time. Hence, the need for 'kernel support' vanished. If you have a pre-2.04 ftape distribution, you can get a patch from ftp.mcc.ac.uk which will implement the dynamic buffers.

Note that you can safely insert a ftape module into a kernel which is configured for 'ftape support'. The only drawback is that you will lose around 112Kbyte of precious dma'able memory.

If you use a v1.2 kernel, you must make a short detour around the `kernel` directory, to apply a patch to the `kernel/ksyms.c` file at this point. The patch can be found as `ftape-2.08/linux-1.2/ksyms.patch`. Change to the directory and apply the patch.

```
# cd kernel
# patch < ../../ftape-2.08/linux-1.2/ksyms.patch
# cd ..
```

(this assumes that you have the `ftape` source lying in `/usr/src/ftape-2.08`).

After you have configured the kernel, you must generate the dependency files and then compile the kernel.

```
# make dep
# make zImage
```

If everything goes well, you should find the compiled kernel in `arch/i386/boot/zImage`. If you have selected any drivers to be compiled as modules, you must also do a '`make modules`'.

Now, you can compile ftape for you new kernel (there is no need to reboot your first).

```
# cd /usr/src/ftape-2.08
```

There are a few options that you can tinker with in the Makefile. Most importantly, if you still use gcc v2.4.5 you *must* add '`-DGCC_2_4_5_BUG`' to the '`FTAPE_OPT`' variable. Also, if you have one of Colorado's FC-10 or FC-20 controllers you must tell ftape this.

4.3 Following the development of the ftape driver

If you want to follow the development of the `ftape` driver, you should read the Usenet newsgroup `linux.dev.tape`. This is really gatewayed from the mailing list `linux-tape@vger.rutgers.edu`, but since `vger` is brought to it's knees due to the load of the various Linux mailing lists, I recommend everyone to read the newsgroup instead.

If you are unable to read news, you can subscribe to the TAPE mailing list by sending a mail saying '`subscribe linux-tape`' (*in the body*) to `majordomo@vger.rutgers.edu`. When you subscribe, you will be sent a greeting mail, which will tell you how to submit real mails and how to get off the list again.

Please note that I do not, repeat **DO NOT**, have any special powers with regard this mailing list. If you're stuck on the list, don't bother to tell me that. I can only shrug and send you my sympathy (but that won't get you off the list).

4.4 Can I format my tapes under Linux?

No! Ian Burrell has offered to work on this and has teamed up with Steven Sorbom to work on this. This is probably not just "around the corner", so please be patient.

Until formatting becomes available under Linux, you'll have to use MessyDOS (arghhh!) instead or buy preformatted tapes. However, some of the preformatted tapes are *not* checked for bad sectors!. If the `ftape` driver encounters a tape with no bad blocks, it will issue a warning. If `ftape` barfs at your preformatted tapes, try out your DOS software. If both the DOS software *and* `ftape` barfs on your tapes, a reformat will very probably cure the problem.

Note that to be able to use your newly formatted tapes under ftape, you must *erase* the tape first:

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

4.5 Which formatting programs can I use under DOS?

These are known to work:

- Colorado Memory System's software (`tape.exe`)
- Conner Backup Basics v1.1 and all Windows versions
- Norton Backup
- QICstream version 2
- Tallgrass FileSecure v1.52
- Escom Powerstream 3.0 (`qs3.exe` – QICstream v3?)

These programs are known to be more or less buggy:

- Conner Backup Basics 1.0
- Colorado Windows tape program
- CP Backup (wastes tape space, but is OK apart from that)

In fact, most software under DOS should work. The Conner Backup Basics v1.0 has a parameter off by one (someone could not read the QIC-80 specs right!), which is corrected in version 1.1. However, `ftape` detects this, and will work around it. Dennis T. Flaherty (<`dennisf@denix.elk.miles.com`>) report that Conner C250MQ owners can obtain the new v1.1, by calling Conner at 1-800-4Conner (in the US) and ask for an upgrade (for a nominal fee for the floppy). The Windows versions should work fine. Some versions of Colorado's tape program for windows, has an off-by-one error in the number of segments. `ftape` also detect and work around that bug.

Central Point Backup can be used, but it wastes precious tape space when it encounters a bad spot on the tape.

NOTE: If you are running a formatting software under DOS, which is not mentioned here, please mail the relevant info to me (<khp@dolphinics.no>), so I can update the list.

4.6 Mixing `ftape` and floppies

Since both the floppy driver and `ftape` needs the FDC (and IRQ6), they cannot run concurrently. Thus, if you have mounted a floppy and then try to access the tape drive, `ftape` will complain that it cannot grab IRQ6 and then die. This is especially a problem when designing a emergency disk for use with ftape. This solution is to either load the boot/root disk into a ramdisk and then unmount the floppy, or have two FDC's.

5 Hardware support

5.1 Supported tape drives

All drives that are both QIC-117 compatible *and* one of the QIC-40, 80, 3010, and 3020 standards should work. QIC-WIDE and Travan drives are also supported (TR-1 is just QIC-80 with 8mm tapes, while TR-2 and TR-3 is a.k.a QIC-3010 and 3020 respectively).

Currently, the list of drives that are known to work with `ftape` is:

- Alloy Retriever 250
- Archive 5580i / XL9250i
- Colorado DJ-10 / DJ-20 (aka: Jumbo 120 / Jumbo 250)
- Colorado 1400
- HP Colorado T1000
- Conner C250MQ(T), TSM420R, TSM850R, TST800R, TST3200R
- Escom / Archive (Hornet) 31250Q
- Irwin 80SX / Insight 80Mb
- Iomega 250

- Iomega Ditto Tape Insider 420, 1700, 3200
- Mountain FS8000
- Reveal TB1400
- Summit SE 150 / SE 250
- Tallgrass FS300 (needs a tiny hack to work with AHA1542B)
- Memorex tape drive backup system
- Wangtek 3040F, 3080F

You can always check out the newest list of drives that are recognised by `ftape`, by looking in the file `vendors.h` in the `ftape` distribution.

Although I do not want to endorse one drive type over another, I want to mention that the Colorado DJ-20 drive is rather noisy, when compared to, say, a Conner C250MQ drive ('tis said that the Colorado is 5-10 times as noisy as the Conner drive. I can't tell for sure, but I have a Colorado, and it *is* quite noisy).

If you have a Tallgrass FS300 and an AHA1542B, you need to increase the bus-on / bus-off time of the 1542B. Antti Virjo (<`klanvi@uta.fi`>), says that changing `CMD_BUSON_TIME` to 4 and `CMD_BUSOFF_CMD` to 12 in `linux/drivers/scsi/aha1542.c` will do the trick.

NOTE: If you have a drive that works fine, but it is not listed here, please send a mail to the HOWTO maintainer (<`khp@dolphinics.no`>).

5.2 Supported special controllers

These dedicated high-speed tape controllers are supported by `ftape`:

- Colorado FC-10, FC-20
- Mountain MACH-2
- Iomega Tape Accelerator II
- 2Mbps controllers (using the i82078-1 fdc)

Support for the FC-10 controller has been merged into the `ftape` driver in version 1.12. See the `RELEASE-NOTES` and the `Makefile` files in the `ftape` distribution. Since of version 2.03 of `ftape`, the FC-20 controller will work (but do check the Release notes!).

The support for the MACH-2 controller was added in `ftape-1.14d`.

To use the IOmega Tape Accelerator II, use `-DMACH2`, and set the right settings for I/O base, IRQ and DMA. This works (by the empirical testing of Scott Bailey <sbailey@xcc.mc.xerox.com>), with at least `ftape-2.02`.

5.2.1 Iomega Ditto Dash and other 2Mbps controllers

The Iomega Ditto Dash, and all other known 2Mbps controllers, use the Intel 82078-1 chip, which can run at 2Mbps. Support for the 82078-1 is currently under development. It is hoped that the support will be completed during january or february.

Current status is that it will work at 1Mbps, with 2Mbps support coming soon (i hope!).

5.3 Un-supported tape drives

- All drives that connect to the parallel port (eg: Colorado Trakker)
- Irwin AX250L / Accutrak 250. (not a QIC-80 drive)
- IBM Internal Tape Backup Unit (identical to the Irwin AX250L drive)
- COREtape light

Generally, ALL drives that connect to the parallel port are NOT supported. This is because these drives uses (different) proprietary interfaces, that are very much different from the QIC-117 standard.

The Irwin AX250L (and the IBM Internal Tape Backup Unit) does not work the `ftape`. This is because they only support QIC-117, but not the QIC-80 standard (they use Irwin's proprietary "servoe (Rhomat)" format). I know nothing about the Rhomat format, nor where to get any info on it. Sorry.

The COREtape light does not accept the initialisation commands, we're feeding it. This pretty much leaves the drive unusable.

5.4 Using an external tape drive with `ftape`

If you have a floppy controller which has a female DB37 connector on the bracket (and some means of delivering power to the drive), you can use it with `ftape`. OK, that sentence was not very obvious. Let's try it this way: Some FDC's (the very ancient one's), have a DB37 connector on the bracket, for connecting to external floppy drives.

If you make a suitable cable (from a quick glance on an FDC that I've got lying around, it seems to be a straight 1-to-1 cable. However, your milage may vary) from the DB37 connector (on the FDC) and to your external tape drive, you can get `ftape` to control your tape drive.

This is because that from a program's view there is no difference between the internal and the external connectors. So, from `ftape`'s point of view, they are identical.

- Pins 20-37: GROUND
- 1: +12 Volt (POWER)
- 2: +12 Volt return (GROUND)
- 3: +5 Volt return (GROUND)

- 4: +5 Volt (POWER)
- 5: 2
- 6: 8
- 7: 10
- 8: 12
- 9: 14
- 10: 16
- 11: 18
- 12: 20
- 13: 22
- 14: 24
- 15: 26
- 16: 28
- 17: 30
- 18: 32
- 19: 34

The power connector is of the "mini" type, sitting on 3.5" floppy drives. As far as I can see on the board, the idea is that you plug one of the power connectors from the PSU to this connector on the board. If you want to use just a single cable, I'd suggest that you get a 50 wire cable, and use multiple wires for the power lines (and ground, for that matter).

5.5 PCI motherboards and `ftape`

Unfortunately, some PCI motherboards cause problems when running `ftape`. Some people have experienced that `ftape` would not run in a PCI based box, but ran flawlessly in a normal ISA based 386DX machine. If you have such a problem, please read the `README.PCI` file in the `ftape` distribution.

6 Backing up and restoring data

This section describes some simple uses of `tar` and `mt`.

6.1 Writing an archive to a tape

You can use '`tar`', '`dd`', '`cpio`', and '`afio`'. You will need to use '`mt`' to get the full potential of your tapes and the `ftape` driver. For a start I'd recommend using '`tar`', as it can archive lots of directories and let you pick out seperate files from an archive. I have been told that `cpio` creates smaller archives and is more flexible than `tar`, but I haven't tried it myself. '`afio`' creates backups where each file is compressed individually and then concatenated. This will allow you to access the files "after" the point of the error. If you use `gzip`ped `tar` files, all data after the point of the error is lost! (to me, this is a pretty good reason for NOT using compression on backups).

To make a backup of your kernel source tree using `tar`, do this (assuming you have the sources in `/usr/src/linux`):

```
cd /usr/src
tar cf /dev/ftape linux
```

This will not compress the files, but gives you a smoother tape run. If you want the compression (and you've got `tar` 1.11.2), you just include the `-z` flag(*), eg: '`tar czf /dev/ftape linux`'

For further instructions on how to use `tar`, `dd` and `mt` look at the man pages and the texinfo files that comes with the respective distributions.

(*) `tar` assumes that the first argument is options, so the '`-`' is not necessary, i.e. these two commands are the same: '`tar xzf /dev/ftape`' and '`tar -xzf /dev/ftape`'

6.2 Restoring an archive

OK, let us restore the backup of the kernel source you made in section 6.1 above. To do this you simply say

```
tar xf /dev/ftape
```

If you used compression, you will have to say

```
tar xzf /dev/ftape
```

When you use compression, `gzip` will complain about trailing garbage after the very end of the archive (and this will lead to a 'broken pipe' message). This can be safely ignored.

For the other utilities, please read the man page.

6.3 Testing the archive

tar has an option (`-d`) for detecting differences between two archives. To test your backup of the kernel source say

```
tar df /dev/ftape
```

If you do not have the man page for `tar`, you are not lost (yet); tar has a builtin option list: try '`tar --help 2>&1 | more`'

6.4 Putting more than one `tar` file on a tape

To put more than one `tar` file on a tape you must have the `mt` utility. You will probably have it already, if you got one of the mainline distributions, e.g. Slackware or Debian.

`tar` generates a single Tape ARchive (that's why it is called '`tar`') and knows nothing about multiple files or positioning of a tape, it just reads or writes from/to a device. `mt` knows everyting about moving the tape back and forth, but nothing about reading the data off the tape. As you might have guessed, `tar` and `mt` in conjunction, does the trick.

By using the `nrft[0-3]` (`nftape`) device, you can use '`mt`' to position the tape the correct place ('`mt -f /dev/nftape fsf 2`' means step over two "file marks", i.e. `tar` files) and then use `tar` to read or write the relevant data.

6.5 Appending files to an archive

"Is there a way to extend an archive – put a file on the tape, then later, add more to the tape?"

No. The `tar` documentation will tell you to use '`tar -Ar`', but it does not work. This is a limitation of the current `ftape` driver.

6.6 Mount/unmounting tapes

Since a tape does not have a "filesystem" on it, you do not mount / unmount the tape. To backup, you just insert the tape and run your '`tar`' command (or whatever you use to access the tape with).

7 Creating an emergency boot floppy for `ftape`

This section was written by Claus Tøndering <`ct@login.dknet.dk`>.

Once you are the happy owner of a tape drive and several tapes full of backups, you will probably ask yourself this question: "If everything goes wrong, and I completely lose my hard disk, how do I restore my files from tape?"

What you need is an emergency floppy disk that contains enough files to enable you to boot Linux and restore your hard disk from tape.

The first thing you should do is to read "The Linux Bootdisk HOWTO" written by Graham Chapman <grahamc@zeta.org.au>. That document tells you almost everything you need to know about making an emergency floppy boot kit. The paragraphs below contain a few extra pieces of information that will make your life a bit easier when you follow Graham Chapman's procedures:

- You don't really need `/etc/init`, `/etc/inittab`, `/etc/getty`, and `/etc/rc.d/*` on your floppy disk. If Linux doesn't find `/etc/init`, it will start `/bin/sh` on your console, which is fine for restoring your system. Deleting these files gives you extra space on your floppy, which you will probably need.
- Find a small version of `/bin/sh`. They are frequently available on the boot floppies that come with a Linux distribution. This again will give you extra space. I'd suggest `ash`, which is extremely small (approx 62Kbytes), and yet very `bash` compatible.
- The `/etc/fstab` you include on your floppy disk should look something like this:

```
/dev/fd0        /               minix   defaults
none            /proc           proc    defaults
/dev/hda        /mnt            ext2    defaults
```

  Once you have booted from your floppy, give the command:

```
mount -av
```

- Make sure your floppy drive is not mounted when you access the streamer tape! Otherwise you may get the following error message:

```
Unable to grab IRQ6 for ftape driver
```

  This means that you **MUST** load the floppy into a RAMDISK.

  This has the unfortunate consequence that the programs needed to restore the files from the tape can not be located on a separate floppy disk. You have two options here:

  1. You place `tar` (or `cpio` or `afio` or whatever other backup program you use) on your root floppy disk. (This is where you'll need all the extra space created in the steps above.)
  2. Before you start restoring from tape, copy `tar` (or `cpio` or `afio` or whatever) to your hard disk and load it from there.

- Apart from your backup program, you will probably need `mt` on your root floppy as well.
- Make sure your ftape device (typically `/dev/nrft0`) is present on your boot floppy.
- Finally: **TRY IT OUT!** Of course, I don't recommend that you destroy your hard disk contents to see if you are able to restore everything. What I do recommend, however, is that you try booting from your emergency disks and make sure that you can at least make a file listing of the contents of your backup tape.

8 Frequently Asked Questions

This is a collection of questions I get asked once in a while, which could fall into the category of FAQ's. If you feel that there is some question that ought to be added to the list, please feel free to mail me (but do include an answer, thanks!).

8.1 How fast is `ftape`?

You can achieve quite respectable backup and restore speeds with `ftape`: I have a Colorado DJ-20 and an Adaptec 1542CF controller, and have measured a 4.25Mbyte/min sustained data transfer rate (no compression) across a 70Mbyte tar archive, while comparing the archive on the tape with data on my 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).

8.2 How do I change the trace-level?

You can do this two ways: either change the default trace-level (the var '`tracing`' in file '`ftape-rw.c`') and recompile or say

```
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.

8.3 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.

8.4 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 | more`'.

If your version of `tar` is v1.11.1 or earlier, consider upgrading to v1.11.2 - 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.

8.5 `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.

8.6 `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.

8.7 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.

8.8 `insmod` says that kernel 1.2.0 and 1.2.0 differ

Did you remember to aply the `ksyms.c` patch to the kernel? If not, see 4.2.2 above.

8.9 `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`'

8.10 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 (by 10. january 96) are:

```
cpio:   2.4.1 (cpio-2.4.1.tar.gz)
dd:     3.12 (fileutils-3.12.tar.gz)
mt:     2.4.1 (cpio-2.4.1.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`. There is a patch for `mt` included in the `ftape` distribution, which makes the `mt status` command spew out usable information for `ftape` drives.

8.11 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.

8.12 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.

8.13 `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 behaviour 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.

9 Debugging the `ftape` driver

9.1 The kernel/`ftape` crashes on me when I do '...' - is that a bug?

No, that is a feature `;-)`

Seriously, reliable software do not crash. Especially kernels do not or rather **should** not crash. If the kernel crashes upon you when you are running `ftape`, and you can show that it is `ftape` that is messing things up, regard it as a Bug That Should Be Fixed. Mail the details to the maintainer (<`khp@dolphinics.no`>) and to the tape list.

9.2 OK, it's a bug ...ehhh... feature - How do I submit a report?

First, make sure you can reproduce the problem. Spurious errors are a pain in the ass, since they are just about impossible to hunt down `:-/` This is a quick check list:

- Kernel version, and patches applied
- `ftape` version
- tape drive model / manufacturer
- Expansion bus type (EISA, ISA, PCI, or VL-bus)
- What you did to expose the problem
- What went wrong on your system.
- Do not delete the kernel and the `ftape.o` file. I might want you run try some patches out or run a different test on your system.

Increase the tracing level to 7 (just below maximum tracing) and run the offending command again. Get the tracing data from the kernel log or `/proc/kmsg`, depending on where you harvest your error messages. Try to look at what `ftape` spews out at you. It may look in-comprehensible to you at first, but you can get valuable information from the logfile. Most messages have a function name prepended, to make it easier to locate the problem. Look through the source, don't just cry "WOLF!", without giving it a try. If your version of the kernel (or `ftape` for that matter), is "old", when compared to the newest version of the kernel, try to get a newer (or even the newest) kernel and see if the problem goes away under the new kernel. When you post your problem report, include the information about ftape version, kernel version, expansion bus type (ISA, VL-bus, PCI or EISA), bus speed, floppy controller, and tape drive. State exactly what you did, and what happened on your system. Some people have experienced that `ftape` would not run in a PCI based box, but ran flawlessly in a normal ISA based 386DX machine (see section 5.5 on PCI machines above)

Also, please think of the poor souls who actually *pay* the their Internet access (like me): avoid posting a (huge) log from the `ftape` run, without reason. Instead, you could describe the problem, and offer to send the log to the interested parties.

Send your bug report to <`linux-tape@vger.rutgers.edu`>. You might also want to mail the bug to <`khp@dolphinics.no`>.