# Linux Migration to Promise RAID Card HOWTO

Benjamin Simkin <bensimkin (at) union.org.za>

2002-12-16

Revision History

| | | |
|---|---|---|
| Revision 1.5 | 2003-03-25 | bs |
| Revision 1.4 | 2002-12-16 | bs |
| | Changed URL for Promise website | |
| Revision 1.3 | 2002-09-04 | bs |
| | Added Troubleshooting section | |
| Revision 1.1 | 2002-08-05 | bs |

**Abstract**

This document aims to outline the steps of how to migrate an existing (Red Hat) Linux install onto a Promise TX2 Card for the purposes of RAID-1 Mirroring. Although I am concentrating on Red Hat, the same principles may be applied to any other distribution.

## Table of Contents

# Introduction

This document contains two main sections, the module setup, and the migration. Neither of these are documented anywhere on the net.

## What?

The Promise Fasttrak is a PCI Ultra 100 RAID Card. It can be used as a RAID card, or as standard IDE interfaces. This documentation DOES NOT address the issues of the 'ataraid' module, but addresses the issues of setting up the card with the propriety module 'ft', as supplied by 'Promise Technologies'.

## Why?

The documentation that comes with the module contains erroneous instructions that if followed, cause the card to not work properly. Also, upon looking over the internet for such documentation, I found a lack.

## How?

Using the Promise Technology module. The module that Promise Technologies provide works well, but the documentation they provide is somewhat lacking.

## Copyright and License

## Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author does not take any responsibility.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

## Contact the Author

You may email me (`<bensimkin (at) union.org.za>`) with improvements and suggestions etc. Please include in the subject line: "Promise RAID HOWTO".

# Module Installation

## What?

You will need an existing Red Hat Linux Install, and you will need to obtain the correct driver from the Promise website [http://www.promise.com/support/download/download_eng.asp].

Choose your card type, and then choose *FastTrak Family Red Hat Linux Driver Uniprocessor*.

## How?

Put the RAID Card into the computer, but leave the hard drive(s) on the motherboard. Boot up into Linux. Login as root. You should have the tar file from the Promise site, if not, see "What?" section above.

## Extracting the module

- Put the tar file into `/tmp`.
- Issue this command: **tar vxf ftrhup_120b9.tar**
- Now we have access to the `modules.cgz` file. Issue the following:

  ```
  # gzip -dc modules.cgz > modules.cpio
  # cpio -idumv < modules.cpio
  ```

  A few directories will be created, so move to the correct directory, depending on your kernel version. You can get your kernel version, by issuing: **uname -r**
- You will see the `ft.o` file. Copy this file to the `/lib/modules/kernel-version/kernel/drivers/scsi` directory.

## Creating the Initial RAM Disk

The Fasttrak module, ft, needs scsi_mod. To use the card as a RAID Card, you will also need sd_mod.

To boot off the card, we will need to create a ramdisk, with the required modules, so that linux can see the hard drive(s).

```
# /sbin/mkinitrd --preload scsi_mod --preload sd_mod --with ft initrd-kernel-versi
```

If your kernel version is **2.4.2-2**, you would issue the command:

```
# /sbin/mkinitrd --preload scsi_mod --preload sd_mod --with ft initrd-2.4.2-2.img
```

Now move the newly created ramdisk image to `/boot`. Its time to edit `lilo.conf`.

Before we edit `lilo.conf`, its best to make a floppy boot disk with the ft module included.

## Creating the Boot Disk

This is very important, as something may go terribly wrong. If LILO stops working, then we just put the disk in and fix the problem. You will need the mkbootdisk program.

```
# /sbin/mkbootdisk --device /dev/df0 --mkinitrdargs '--preload scsi_mid --preload
```

If your kernel version is **2.4.2-2**, then issue the following

```
# /sbin/mkbootdisk --device /dev/df0 --mkinitrdargs '--preload scsi_mid --preload
```

# Linux Configuration

## What?

Instead of editing our current working system, we merely create an entry for our new system. So if something stuffs up, we just reboot and we're fine.

## How?

Example `lilo.conf` file:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux

image=/boot/vmlinuz-2.4.2-2
        label=linux
        read-only
        root=/dev/hda5
```

Just copy, and edit the kernel part, as follows:

```
image=/boot/vmlinuz-2.4.2-2
        label=LinuxFromRAID
        initrd=/boot/initrd-2.4.2-2.img
        read-only
        root=/dev/sda5
```

Note, how we setup the ramdisk, and we setup the root value. Don't set it as default yet. Now run lilo, in verbose mode.

```
# lilo -v -v
```

If you don't see any errors, we're fine, so halt the system.

Put only **ONE** hard drive on the Card, the hard drive with the existing Linux system.

Boot up, and go into the card bios. Delete the array of the current set.

# Array Configuration

## Building the Array

Add the second hard drive to the card, and turn on your computer. Go into the card bios, and create a mirror array, selecting the appropriate source disk. Make sure you *Create and Build* the array. After that is done, follow the instructions of the next section.

## Changing Configuration Files

Choose "LinuxFromRaid" Option at LILO.

It should boot up without any errors. It will, however complain about swap. Ignore that, as we will fix it in a moment.

You may receive an error about incorrect module version, or a kernel mismatch. Please see the "troubleshooting" section about this.

After we have booted up on the new system, we must edit a few files:

- /etc/lilo.conf
- /etc/fstab

## /etc/lilo.conf

The parts in italics have been edited. (See the original lilo.conf file as stated earlier on in this HOWTO)

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=LinuxFromRAID

image=/boot/vmlinuz-2.4.2-2
        label=linux
        read-only
        root=/dev/hda5

image=/boot/vmlinuz-2.4.2-2
        label=LinuxFromRAID
        initrd=/boot/initrd-2.4.2-2.img
        read-only
        root=/dev/sda5
```

Now, save the file, and run lilo to write the boot record.

## /etc/fstab

For every entry that pointed to the old location, you must now change it to the scsi device, but use the same partition. For example:

```
/dev/hda2               swap                    swap    defaults        0 0
```

If you saw this in your fstab, you would need to change /dev/hda2 to /dev/sda2 Now reboot your system. That's it. After the reboot, its best to check the status of the RAID.

# Monitoring the Array

You can get detailed information by looking at the directory /proc/scsi/fasttrak/

In that directory, you will see the file which corresponds to your array.

```
# cat /proc/scsi/fasttrak/0
Promise FastTrak Series Linux Driver Version 1.2.0.14
Adapter1 - FASTTRAK100 TX2
Array    - Array[1] : 1X2 Mirror (OK)
Drive    -
  1 : IC       Primary  /Master  41174MB BASE(0xb400) BM(0xc400) IRQ(11) UDMA5
  3 : IC       Secondary/Master  41174MB BASE(0xbc00) BM(0xc408) IRQ(11) UDMA5
```

That is all.

# Troubleshooting

You may get a kernel mismatch error message. This means that you don't have the correct version module for use with your kernel. If you cannot get the correct module version for use with your kernel, choose the closest one, and follow these steps;

This may not work in some instances, so remember to have your boot disk handy.

```
# cd /boot
# cat initrd-2.4.2-2.img |gunzip > /tmp/myimage
# mkdir /mnt/tmp
# mount /tmp/myimage /mnt/tmp -t ext2 -o loop=/dev/loop3
# cd /mnt/tmp
# vi linuxrc
```

Now modify 'insmod ft' to 'insmod -f ft'.

This will force the sytem to load the module. Save the file, and exit from vi (or emacs :))

```
# umount /mnt/tmp
# gzip /tmp/myimage
# cp myimage.gz /boot/initrd-2.4.2-2.img
```

You will obviously have to use the correct filename for your initrd file. I am using 2.4.2-2 as example.

Now, reboot, and all should be well.