

Linux Disaster Recovery

En Francais

Having spent in excess of fifteen years breaking various flavours of SCO UNIX/XENIX, I became fairly familiar with the procedures required to restore a system from a cpio tape backup. This usually involves the custom root/boot floppy disks, repartitioning the disk and watching the tape drive wobble on for what seems like years. After you've done it a few times, the panic subsides and it becomes routine.

The Linux bug has finally caught up with me, largely because work decided that we ought to move that way. I was duly despatched to become RedHat certified. I was hoping to learn how to use the skills honed on SCO systems over the years, on Linux systems.

Unfortunately there was very little in the RHCE course about tape backups and restoration, a critical part of any sysadmin's arsenal, IMHO. I suppose this must be something to do with the proliferation of RAID controllers, the potential for a catastrophic disk failure is much more remote.

I wasn't impressed when I discovered that the rescue kernel supplied with RedHat 6.2, 7 and now 7.2 don't let you access your tape drive as part of rescue! I resolved to find out how to do it, firm in the belief that if I know how to do it, I'll never have to!

Before we go any further, I did have a read through the boot disk HOWTO. I recoiled in horror at the amount of meddling and kernel trimming involved in getting such a disk prepared to my liking. Best leave that to the likes of RedHat etc. You should be able to recover from tape by loading the appropriate driver module in conjunction with the install CD.

The information that follows is largely the result of various meddlings on my part, and should be viewed in the knowledge that it may be dodgy or just plain *wrong!*. You have been warned! My experiences are exclusively with RedHat 6.2, 7.0 and 7.2. Why not 7.1? By the time I got around to looking at it, it had been superseded. It is entirely possible that some or all of the following information may be relevant to other distributions.

Requirements

1. A recent cpio format backup of at least the root and boot filesystems
2. Red Hat 6.2, 7.0 or 7.2 installation CD's and possibly the boot floppy if your system can't boot from CD
3. A tape driver module compatible with your hardware and compiled for the **same kernel version** as the installation CD.
4. A SCSI driver module compatible with your hardware and compiled as above. This may be required if your SCSI controller is not detected at start up, or is not supported by the rescue kernel.

The driver module may be tricky to obtain. You can't just pull any old module off a running system and load it. I know this, because I tried it and ended up in the pub! It must be compiled for the kernel version present on the installation CD. The **st.o** module is required for use with the majority of SCSI tape devices. Alternatively you might need the **tpqic02.o** module if, like me, you are using an antiquated QIC02 drive such as an Archive/Wangtek/Emerald etc.

Building The Modules

Neither RedHat 6.2, 7.0 or 7.2 provide a suitable module on the installation CD. RedHat 7.0 does provide an st.o module, but it is to be found on the credit card sized **Sysadmin Survival CD** and is present in the file **/lib/modules-2.2.16-22.tar.bz2**. RedHat 6.2 does not provide a suitable module. Even if you have this CD, you'll need to extract the module in advance of any recovery work. I'm not sure if 7.2 includes a Sysadmin CD. I downloaded my copy as ISO images. More fun than watching paint dry, but not much.

If you're using 6.2/7.2, or you need QIC02 support for RedHat 7.0, you'll need to compile the modules from the appropriate source. The kernel version varies by distribution. No surprise there:

- For 6.2 **2.2.14-5.0BOOT**
- For 7.0 **2.2.16-22BOOT**
- For 7.2 **2.4.7-10BOOT**

Compiling a kernel is all relatively straight forward, the source code is included on the installation CD's. Instructions for compiling kernel/modules can be found in numerous places, but try <http://www.cpqlinux.com/kernel.html> as a starting point.

There are some points to note

- use the same kernel configuration as that used to build the kernel found on the installation CD. This is typically provided by the kernel-BOOT rpm.
- you must append **BOOT** to the EXTRAVERSION setting in the Makefile, otherwise you won't be able to load the modules when running rescue.
- when you run menuconfig or xconfig etc, you will need to enable QIC02 support if you need the tpqic02.o module. Make sure that you specify that QIC02 support is to be modular.

If you need QIC02 support, it's probably a good idea to specify that you want runtime configuration, otherwise you're stuck with one model support and you won't be able to change hardware configuration such as DMA, IRQ etc. Very inconvenient if you can't remember those pesky hardware settings and can't be bothered to take the lid off.

Runtime configuration requires a configuration utility such as **qic02conf** found in **tpqic02-support-1.9b.tar.gz** available from <http://penguinfiles.com>

Copying Modules To A Filesystem Floppy

Once you've compiled the modules, you'll need to copy them to a floppy disk that has a Linux file system on it. You need a file system because the floppy has to be mounted under the rescue environment so that you can access the modules.

- `mke2fs /dev/fd0`
- `mount /dev/fd0 /mnt/floppy`
- `cp st.o /mnt/floppy`
- `umount /mnt/floppy`

Fortunately, all this hard work can be omitted because the modules and support file for 6.2 and 7.0 are available from <http://www.jezndi.org/pub/recover.img.gz>. This floppy also contains the SCSI tape module st.o for RedHat 6.2, 7.0 and 7.2 rescue kernels.

This is a Linux filesystem image. Once you've downloaded it, copy it to a floppy with:

- `gunzip -c recover.img.gz > /dev/fd0`

or:

- `zcat recover.img.gz > /dev/fd0`

Hey it's *nix right? there's more than one way to do it!

If you're reading this and the wheels have come off and you don't have access to a Linux system, you can use the RAWRITE.exe DOS/Windows utility on the installation CD to create the floppy image, after you've decompressed it.

Running Rescue

Boot from either the CD or the installation floppy. At the boot prompt type *linux rescue*. Assuming that your hardware is detected all the appropriate modules should be loaded. If not you may have to modprobe them, or maybe specify boot parameters.

I have heard rumours that RedHat 7.2 can be a little reluctant to boot from the CD. I've certainly had fun and games getting it to go, succeeding about 1 time out of 5. I set the floppy drive setting in the BIOS to pretend that the floppy is a 2.88MB drive. This seems to have helped matters, but not much. If you're using Dell hardware, it may be worth checking whether your CD is set to master/slave rather than cable select, judging by this account.

Once the boot process has finished, you'll be dropped out to a bash prompt. The ramdisk created as part of the boot process doesn't contain the appropriate device nodes to access devices such as the floppy, tape drives or the hard disks. Therefore you need to create them.

Exactly which device nodes you need depends upon your hardware. Some devices also require you to specify the major and minor numbers. Common examples are listed below:

Generic SCSI devices - major/minor numbers not required

- `mknod /dev/sda # 1st drive whole device`
- `mknod /dev/sdb # 2nd drive whole device`
- `mknod /dev/sda1 # 1st drive partition 1`
- `mknod /dev/sda2 # etc.`

Generic IDE devices - major/minor numbers not required

- `mknod /dev/hda # 1st drive whole device`
- `mknod /dev/hdb # 2nd drive whole device`
- `mknod /dev/hda1 # 1st drive partition 1`
- `mknod /dev/hda2 # etc.`

For Compaq Smart Array devices, you need to create the `/dev/ida` directory before creating the nodes. You must also specify the block major and minor numbers:

- `mkdir /dev/ida`
- `mknod /dev/ida/c0d0 b 72 0 # 1st drive whole device`
- `mknod /dev/ida/c0d1 b 72 16 # 2nd drive whole device`
- `mknod /dev/ida/c0d0p1 b 72 1 # 1st drive partition 1`
- `mknod /dev/ida/c0d0p2 b 72 2`
- `mknod /dev/ida/c0d0p3 b 72 3 # etc..`

The Compaq CCISS devices require similar setup to the Smart Array Devices:

- `mkdir /dev/cciss`
- `mknod /dev/cciss/c0d0 b 104 0 # 1st drive whole device`
- `mknod /dev/cciss/c0d0p1 b 104 1 # 1st drive partition 1`
- `mknod /dev/cciss/c0d0p2 b 104 2 # 1st drive partition 2`
- `mknod /dev/cciss/c0d0p3 b 104 3 # 1st drive partition 3`

You also need to create devices for the tape drives:

QIC02 devices

- `mknod /dev/rct0 c 12 9`
- `mknod /dev/tape c 12 8`

Generic SCSI tape

- `mknod /dev/st0 c 9 0`

Floppy disk

- `mknod /dev/fd0`

Once the nodes are created, you should be able to access the appropriate device and list the partition table `fdisk -l <device>`, which should produce something like this:

Disk /dev/hda: 240 heads, 63 sectors, 776 cylinders
Units = cylinders of 15120 * 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	278	2101648+	6	FAT16
/dev/hda2	*	279	283	37800	83	Linux
/dev/hda3		284	318	264600	83	Linux
/dev/hda4		319	776	3462480	5	Extended
/dev/hda5		319	336	136048+	82	Linux swap
/dev/hda6		337	614	2101648+	83	Linux
/dev/hda7		615	776	1224688+	83	Linux

If you've had to replace the disk, it's more than likely that you won't have a partition table at all, in which case you'll need to create one using

- `fdisk <device>`

A couple of points to remember:

1. make sure that the boot and root partitions are within the 1024 cylinder boundary, otherwise you may run into BIOS related problems booting.
2. don't forget to allocate a swap partition. The system type for a swap partition is 82.
3. partitions 5 onwards are extended partitions that live within one of the four primary partitions.

If you've had to recreate a partition table, or you've got a particularly badly damaged filesystem, you need to create filesystems on the partitions.

- `mke2fs <devicename>`

will do this. For the swap partition, use

- `mkswap <devicename>`

For **RedHat 7.0**, you must specify the volume label when creating the filesystem. E.G. for the /boot filesystem use:

- `mke2fs -L /boot <devicename>`

and for root use:

- `mke2fs -L / <devicename>`

Failing to add the volume label renders the filesystem unusable.

If the volume label is missing, you'll get a message like this when you reboot and try to mount it:

```
The superblock could not be read or does not describe a correct ext2
filesystem. If the device is valid and it really contains an ext2
filesystem (and not a swap or ufs or something else), the the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
```

```
e2fsck -b 8193
: No such file or directory while trying to open LABEL=/boot
```

```
*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell
```

Thanks to Stéphan Evrat for pointing this out.

Monsieur Evrat also tells me that you can update the label with

- `tune2fs -L <label> <devicename>`

Alternatively you could use

- `e2label <devicename> <label>`

If the optional argument `<label>` is not present, `e2label` will simply display the current filesystem label.

RedHat 7.2 introduced the `ext3` journalled filesystem. If you need to recreate an `ext3` filesystem, use the syntax for RH7.0, but use the `-j` options as well.

Mounting Filesystems

Create a mount point for the root file system. Create a mount point for the floppy. For the sake of argument I'll use **sysimage** and **floppy**.

Mount the root file system on **sysimage** Change directory to **sysimage** and make mount points for the other file systems normally mounted when the system is up and running. These will typically include `boot` `usr` `var` `home` etc. You should also create a `proc` directory otherwise the system won't be able to mount the `/proc` file system when it reboots. Mount each of the filesystems you wish to restore on the appropriate mount point:

- `mount /dev/hda2 sysimage`
- `cd sysimage`
- `mkdir boot usr home var proc`
- `mount /dev/hda1 boot`
- `mount /dev/hda4 usr`

RedHat 7.2 offers to do the hard work for you. It creates device nodes in `/tmp` (relative to the rescue environment) and mounts the root filesystem on `/mnt/sysimage`. This can be quite useful, but it's possible that it will fail to unmount the root filesystem when you reboot. This then means that it'll need to clean it up again when you reboot.

If you're using existing filesystems and the mount fails with no error message, it's likely that the file systems need cleaning. **fsck** is your friend here. If you're using an ancient EISA Compaq Smart Array controller, read this.

Insert the modules floppy disk and mount it on **floppy**.

- `mount /dev/fd0 /floppy`

Change directory to floppy and load the tape driver module. Confirm it has loaded with **lsmod**.

- `cd floppy`
- `insmod <advansys.o>` (if you're `<SCSI>` driver needs loading)
- `insmod st.o`
- `lsmod`

If you're using the **QIC02** devices, you'll need to use **tpqic02.o** instead. You will then need to configure the tape device. You'll need the **qic02conf** binary to do this. You'll find it is on the floppy image detailed above, along with the modules. You'll also need a config script unless you really want to specify about 10 command line options to `qic02conf`. Suitable ones are included with the `qic02conf` distribution, or on the floppy image.

Change directory to the root mount point and restore your file systems:

- `cd sysimage`
- `cpio -iuvdmBc --no-absolute-filenames < /dev/st0`

or `/dev/rct0` for QIC02 devices.

If you're only restoring one file system, and you have a full system backup you will probably have to figure out what options `cpio` requires. Such runes are outside the scope of this (and probably all ;-)) documents and you are referred to the `cpio` man page.

The file systems are now being restored!

RedHat 7.2 and cpio

RedHat 7.2 does not use the GNU cpio binary. Instead it uses the BusyBox program to perform a variety of functions. BusyBox contains a cut down set of command switches for cpio (and everything else), so the cpio syntax for RedHat 7.2 is

- `cpio -iuvdm < /dev/st0`

There is a problem, however. The version of Busybox shipped with the RedHat 7.2 distro (0.52pre) has no support for tape devices in cpio. This shows itself when you try and access the tape. Mostly, nothing appears to happen, or at best the tape drive gurgles. Support is being added, but this isn't going to help the CD rescue environment.

A work around is to be had by using cat.

- `cat /dev/st0 | cpio -iuvdm`

This solution was suggested by one of the developers of BusyBox, whilst he attempted to fix the problem. I tested the tape drive patch which seems to work OK. I included the patched BusyBox/cpio on the floppy image in the bin directory. Why? because I could.

Reinstalling LILO

Once the restore has finished, you may need to reinstall the lilo boot manager.

Execute **lilo** from within the **sysimage** directory. Notice that this is a relative path and we're using the newly restored lilo binary and the newly restored **etc/lilo.conf** file.

- `cd sysimage`
- `sbin/lilo -r /sysimage`

There is an implicit assumption here that your root filesystem remains on the same device that it started on. If it isn't, use **pico** to edit `etc/lilo.conf` before you install lilo. You will also need to update **/etc/fstab** as well, otherwise when the system boots, it will get confused about which partitions are to be mounted where.

Unmount all the filesystems, including the floppy and reboot. If all has gone well, your system should now reboot.

- `umount usr`
- `umount var`
- `umount boot`
- `cd ..`
- `umount sysimage`
- `umount floppy`
- `<ctrl-alt-del>`

RedHat 7.2 has an alternate boot manager, grub. I've no experience with it (yet), so I won't attempt to document its installation here.

SCSI Modules Compiled for RedHat Rescue Kernel

Since writing this page, I've acquired a DAT drive and hence a SCSI controller. Alas, it is an Advansys ABP-510 ISA bus card. It's not fast, but it works (once I'd tracked down the utility to set it up and stop it hanging the whole system because of IRQ conflicts!

Naturally I tested out all the above just to make sure I wasn't wide of the mark. I found that the `advansys.o` driver isn't loaded automatically, and that it fails to load at all with the rescue kernel. More of the same, I concluded and set about compiling the appropriate version. I've included this on the floppy image. The SCSI module needs to be loaded before the `st.o` module.

Whilst I was at it, I copied all the scsi modules into a tar file which can be found at <http://www.jezndi.org/pub/RH6.2-recover-scsi.tgz>. Together, they're too big to fit on the floppy. Modules for RedHat 7.0 can be found at <http://www.jezndi.org/pub/RH7.0-recover-scsi.tgz>.

Because the floppy image is a filesystem, you can mount it and copy files to and from it as if it was actually a disk. The magic string here is:

- `mount /tmp/recover.img /mnt/recovery -o loop`

This saves time mucking about with floppy disk and all that dd nonsense you'd otherwise have to go through.

Making a CPIO Backup

By popular demand, a 30 second tutorial on making a CPIO backup. You really should read the cpio man page for the full story, but here are some quick and dirty examples to make the lights blink on your tape device for a minute or two:

Root filesystem:

- `find / -mount -print | cpio -ocB -O /dev/st0`

Boot filesystem:

- `find /boot -mount -print | cpio -ocB -O /dev/st0`

Both together:

- `find /boot / -mount -print | cpio -ocB -O /dev/st0`
-

Typical lilo.conf

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

image=/boot/vmlinuz-2.4.18-17.7.x
    label=linux
    initrd=/boot/initrd-2.4.18-17.7.x.img
    read-only
    root=/dev/sda7
```

Useful Links

- [En Francais](#)
- [PDF Version](#)
- [Installing RedHat Linux 6.2 on a Compaq Prosignia 300](#)
- [Strangeness with 6.2 Rescue and Compaq EISA SMART Controllers](#)
- [Red Hat](#)
- [Compiling the Linux kernel](#)
- [How I earn my living](#)
- [Your comments are welcome](#)