New Drivers for Kickstart Disks

Driver Injection

The Kickstart mechanism provides a convenient way of installing Red Hat Linux automatically. However, the drivers on the Kickstart disk may not always match the hardware you are using – do not fear as help is at hand as we guide you through the utility that once configured will save you many hours otherwise spent installing.

BY OLIVER SCHADE

If you need to install Linux on a large number of computers, you will probably want to avoid clicking through the installation menus on each system. Servers without a keyboard or monitor also need an alternative approach.

Kickstart [1] is the Red Hat Installer's answer to this problem: all you need to install the complete system across the wire is a boot disk. Partitioning, packet selection, X11 configuration and similar details are stored in a configuration file, meaning that no admin interaction is required at any point.

If the boot disk supplied with the Red Hat does not have a suitable drivers for the hardware in question, Kickstart needs access to the additional driver disks. The disks contain the required kernel modules, but the admin user has to insert them manually. Also, the software will then require some interaction, so this mechanism is unsuitable for automatic installations without a keyboard or a display.

If you still want to use Kickstart, you can modify your boot disk to reflect your hardware. To do so, you will need a development system with an identical Red Hat Linux version; the computer hardware is not important in this case.

Our example uses an HP Proliant Server DL380 GE. The test System has dual Xeon 2.6 GHz CPUs, 6 Gbytes of RAM, an additional RAID controller (four channel, 256 Mbytes of cache RAM), six U160 hard disks with a capacity of 73 Gbytes and two Broadcom 10/100/1000TX network adapters. The Red Hat 7.3 Kickstart disks recognize neither the RAID controller nor the Broadcom NICs.

The boot floppy is formatted with a FAT filesystem. Lilo uses the parameters in *syslinux.cfg* and any options the user typed at the boot prompt to launch the *vmlinuz* kernel. The drivers are all stored in the image file, *initrd.img*. The floppy also provides **.msg* help texts that the installer can display if needed. These are ASCII files with Escape sequences that define the color scheme. The structure of the Kickstart disk is shown in Listing 1.

Extracting Drivers

The */mnt/cdrom/images/bootnet.img* directory on the Red Hat CD contains a

Listing 1: Structure of the Kickstart disk

```
/mnt/floppy
|-- boot.msg
|-- general.msg
|-- initrd.img
|-- ldlinux.sys
|-- param.msg
|-- rescue.msg
|-- snake.msg
|-- syslinux.cfg
`-- vmlinuz
```

disk image. To change the content of the image file, you first need to make a copy: */tmp/bootnet.img*. A loopback mount as */mnt /loop0* will allow access to the filesystem:

```
mkdir /mnt/loop0
mount -o loop /tmp/bootnet.img ➥
/mnt/loop0
```

Lilo loads the *initrd.img* file before loading the (*vmlinuz*) kernel, thus providing an initial RAM disk with drivers, binaries and device files for the kernel. Initrd is Gzip compressed and contains an Ext-2 filesystem. Perform the following steps to access the content:

```
cp /mnt/loop0/initrd.img➥
 /tmp/initrd.ext2.gz
gunzip /tmp/initrd.ext2.gz
mkdir /mnt/loop1
mount -o loop /tmp/initrd.ext2➥
 /mnt/loop1
```

This places the contents of Initrd in */mnt/loop1*. The structure is shown in Listing 3.

The kernel modules are stored in a cpio archive. The default drivers are stored in the *modules* subdirectory. You will find the kernel modules in a compressed cpio archive called *modules.cgz*. *pcitable* assigns suitable drivers to PCI ID numbers, and the *modules.dep* and *module-info* describe the module dependencies and types.

The next step is to extract the drivers from the archive and replace unwanted types with self-compiled or manufacturer supplied modules. The following syntax extracts the drivers and creates a new directory:

```
gunzip -c /mnt/loop1/modules/➥
modules.cgz > /tmp/modules.cpio
cd /tmp
cpio -idv < modules.cpio
```

The name depends on the Red Hat version you are using and reflects the kernel version for the boot kernel on the installation disk. The directory for Red Hat 7.3 is called *2.4.18-3BOOT*, and contains the following drivers:

- *3c59x.o*
- *8139too.o*
- *8390.o*
- *eepro100.o*
- *lockd.o*
- *mii.o*
- *vne2k-pci.o*
- *nfs.o*
- *vpcnet32.o*
- *scsi_mod.o*
- *vsunrpc.o*
- *tulip.o*
- *usb-storage.o*

Any drivers you do not need can simply be deleted. There are several modules that our Proliant Server DL380 G3 does not need; we can use *rm* to remove them:

```
cd 2.4.18-3BOOT
rm -f 3c59x.o 8139too.o 8390.o➥
 tulip.o  eepro100.o mii.o➥
 ne2k-pci.o pcnet32.o
```

Listing 2: Creating a PCI Table

```
0x14e4  0x1644  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x14e4  0x1645  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x14e4  0x1646  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x14e4  0x1647  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x14e4  0x164D  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x14e4  0x16A6  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x14e4  0x16A7  "bcm5700"  "BROADCOM NetXtreme 10/100/1000TX [onboard]"
0x0e11  0x4070  "cciss"    "Compaq RAID Controller Card"
0x0e11  0x4080  "cciss"    "Compaq RAID Controller Card"
0x0e11  0x4082  "cciss"    "Compaq RAID Controller Card"
0x0e11  0x4083  "cciss"    "Compaq RAID Controller Card"
```

Listing 3: Initrd structure

```
/mnt/loop1
|-- bin -> sbin
|-- dev
|   |-- agpgart
|   |-- console
|   [...]
|   |-- ttyS3
|   |-- ttyp0
|   `-- zero
|-- etc
|   |-- fonts.cgz
|   |-- keymaps.gz
|   |-- kon.cfg
|   |-- lang-table
|   |-- loader.tr
|   |-- minikon.fnt
|   |-- mtab -> /proc/mounts
|   |-- passwd
|   |-- ramfs.img
|   `-- terminfo
|       |-- k
|       |   `-- kon
|       `-- l
|           `-- linux
|-- linuxrc -> /sbin/init
|-- lost+found
|-- modules
|   |-- module-info
|   |-- modules.cgz
|   |-- modules.dep
|   `-- pcitable
|-- proc
|-- sbin
|   |-- continue -> loader
|   |-- init
|   |-- insmod -> loader
|   |-- loader
|   |-- modprobe -> loader
|   |-- rmmod -> loader
|   `-- sh -> /usr/bin/sh
|-- tmp
`-- var
    `-- state
        `-- xkb -> /tmp
```

The Compaq support site [2] has SRPM drivers for the Broadcom NICs. The standard *cciss* Linux kernel drivers are recommended for the Smartarray RAID controller. You need the Red Hat kernel sources and a matching configuration file on your development system to create the modules:

```
rpm -i kernel-source-2.4.18➥
-3.i386.rpm
cd /usr/src/linux-2.4.18-3
cp configs/kernel*BOOT .config
```

To ensure that the newly compiled modules will have the right Kernel version number, you will need to change the *EXTRAVERSIONS* line in the Makefile to *-3BOOT*. For Red Hat 7.3 the APM option must additionally be disabled using *make menuconfig*.

Compiling New Drivers

This might also be a good opportunity to enable a driver module for the array controller. The driver is located in the *Block Devices* kernel configuration menu under *Compaq Smart Array 5xxx support*. When we launch the compiler one of the modules it creates is *cciss.o*; this module must be placed on the Kickstart disk:

```
make dep clean bzImage
make modules modules_install
cp /lib/modules/2.4.18-3BOOT/➥
kernel/drivers/block/cciss.o➥
 /tmp/2.4.18-3BOOT/
```

SRPM makes compiling the Compaq driver for the NICs simple – although you do need to ensure that you modify *.config* and *Makefile* in the kernel sources, as previously described, before doing so. We also need the *bcm5700.o* module on the disk:

```
rpm -i /tmp/bcm5700-*src.rpm
cd /usr/src/redhat/SPECS
rpm -ba bcm5700.spec
rpm -i /usr/src/redhat/RPMS/➥
i386/bcm5700*
cp /lib/modules/2.4.18-3BOOT/➥
kernel/drivers/net/bcm5700.o ➥
/tmp/2.4.18-3BOOT/
```

The *strings bcm5700.o | grep kernel* command tells us whether the correct version number has been assigned to the modules. The output must correspond to the *vmlinuz* kernel version on the *bootnet.img* disk.

Automatically Loading the Additional Modules

To ensure that the kernel automatically loads the new drivers, module dependencies and PCI ID assignments must be correct.

Module dependencies are configured in *modules.dep*. The template in */lib/modules/2.4.18-3BOOT/modules.dep* is fine for any standard Linux kernel drivers, but it is often a case of trial and error for other drivers.

cciss.o and *bcm5700.o* do not require any additional modules, not even a *scsi* driver. The *cciss* is a block device driver, so there are no changes required to *modules.dep*.

The driver assignment for PCI identification numbers is stored in the */mnt/loop1/modules/pcitable* configuration file. The file content is comprised of four tab-separated columns:

- Manufacturer identification
- Device identification
- Module name
- Description

Jim Boemler's web site [3] gives a list of possible manufacturer and device IDs. Several entries match the array controller and network adapters in our system (see Listing 3), and so they need to be added to *pcitable*.

Compiling a New Image

Now that we have all the drivers, we use them to start assembling a boot disk. We'll start with the drivers modules by creating a cpio archive, use Gzip to compress it, storing the results in *initrd.ext2* – which is still mounted as */mnt/loop1*:

```
cd /tmp
find 2.4.18-3BOOT -print ➥
-depth |  cpio -ov -H crc ➥
> modules.cpio
gzip -c9 modules.cpio ➥
> modules.cgz
cp -f modules.cgz /mnt/➥
loop1/modules/
```

The next step is to copy the modified Initrd back to the Kickstart image – it is available on */mnt/loop0*. The complete image can then be copied to the disk using *dd*:

```
umount /mnt/loop1
gzip -c9 initrd.ext2 ➥
> initrd.img
cp initrd.img /mnt/loop0/
umount /mnt/loop0
dd if=/tmp/bootnet.img ➥
of=/dev/fd0H1440  bs=512
```

A modified image may be too large to store on a floppy disk. You will need to delete several drivers from Initrd. Avoid making multiple changes to Initrd in one step: the Ext-2 filesystem will release any unused blocks when you delete an entry, but it does not remove their content. Data of this kind is difficult to compress and can increases the size of the image.

You need to type *ks* at the Lilo prompt to boot the disk with the Kickstart server. To install a server without a keyboard or screen, you should change the *default* line in *syslinux.cfg* to *default ks*.

Red Hat in 15 Minutes

This method allows you to install Red Hat Linux on a keyboardless Proliant Server DL380 G3 in just 15 minutes. As we had to install more than 20 machines of this type, it was well worth the effort of modifying a bootnet disk. ■

INFO

[1] Kickstart documentation: *http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/*

[2] Compaq support page for Linux drivers: *http://h18000.www1.hp.com/support/files/server/us/locOsCat/86.html*

[3] PCI Vendor and Device List: *http://www.yourvote.com/pci/*

THE AUTHOR

Oliver Schade is employed as a Linux and storage consultant and Teamleader for Pro Business Berlin AG, Berlin, Germany.

While still at university, studying computer science, Oliver worked for various Internet providers in entirely Linux based environments. His Linux experience dates back to 1992 and the 0.96b kernel.