

Saving and Restoring a Flash Based Linux Disk Image

I. *Preparing an SIB to Boot from a Hard Disk with DiskOnChip Drivers*

Note: This section is only necessary for boards that utilize the M-Systems DiskOnChip storage media. Those that utilize IDE-compatible Compact Flash do not need a kernel with custom drivers and can proceed to the next section '*Utilizing the put_image.sh and get_image.sh scripts*'.

1. Prepare a hard disk with a basic version of any distribution of Linux. The smaller and more basic, the better.
2. For applications utilizing the M-Systems DiskOnChip driver, you will need to compile a 2.4.14 kernel, with the included configuration file. Copy the DiskOnChip driver module located in (`/lib/modules/2.4.14doc/kernel/drivers/block/doc/doc.o`) to the modules directory (`/lib/modules/2.4.14/kernel/drivers/block/doc/`) and arrange for the distribution to load this module on bootup (On a Debian installation, simply add `doc` to the `/etc/modules` file).
3. When the driver loads, look for a line in the status messages that begins `fl_init: registered device at major: XXX` The XXX is the device major of the DiskOnChip driver. Note this number because you will need to create the device nodes to access the DiskOnChip.
4. Next, you will need to create the device nodes. These are special files that correspond to devices. You will need to create the first node for the whole disk, and the second node for the first partition. Create each node by issuing the following command, replacing Y with the minor number (0 for the whole device, and 1 for this first partition) and XXX with the device major number you determined earlier:

```
mkknod /dev/docY b XXX Y
```

5. Create the mountpoint for the DiskOnChip device:

```
mkdir /doc
```

6. The attached kernel configuration file has facilities for a serial console. On boards without video (such as the PCM-53e52), you will need to enable it. When using LILO as the bootloader, this is accomplished by adding the following line to the global section of the `lilo.conf` file:

```
serial=0,9600n8
```

and this line to the section which corresponds to the kernel you're using:

```
append="console=ttyS0,9600"
```

Now that you have a functional hard-disk based installation that utilizes the serial console, you can boot the board with it. Connect a null modem cable between the first COM port of the SIB, and a free COM port on a host PC. Use terminal settings of 9600 baud, 8 data bits, 1 stop bit, no parity, and no flow control.

II. *Utilizing the put_image.sh and get_image.sh scripts*

1. Boot the SIB, and watch for the LILO boot prompt. You should see it within 45 seconds.
2. After booting, log in as the superuser `root`. Utilize the `put_image` command to transfer a Flash Disk image (tarball) to the Flash Disk device, or `get_image` to transfer a Flash Disk image (tarball) from the Flash Disk device. These commands take one argument, the filename of the image.