power on?
by Wayde Schroeder

A35_PowerOn.tiff ¬
Have you ever turned on your NeXT computer to find an error message that leaves you baffled? Did you ever wish there was a great support bulletin article that helped explain what you can do to make your computer work normally again? Do you also want to know what startup options you can use when you're at the ROM monitor prompt ªNeXT>º?

This article explains the startup (boot) flags and operating system (kernel) variables you can use when you start your computer, whether for common startup solutions and options or disaster recovery.

standard boot devices
The entry normally held by the boot parameter is sd, od, en, or fd. When stored in the boot parameter, this entry is all that's needed. From the ªNeXT>º prompt, you type a b designating a boot command, followed by the appropriate entry. The general usage when working from the ROM monitor is:

b*xx*
where: *xx* = sd, od, en, fd

sd	SCSI disk
od	optical disk
en	Ethernet (network booting)
fd	floppy disk

The discussion of the boot command will be expanded later in this article to include boot flags and kernel variables.

setting the default boot device
Normally, once you set the default boot device, it doesn't change. You may, however, occasionally need to change this setting.   This section explains how to change the default boot device when working from the ROM monitor and installing a 68040 upgrade board.

New 040 upgrade boards are sometimes shipped with the default boot device set for od (optical disk). If you normally booted on the internal SCSI disk, for example, your computer may stop and ask you to insert an optical disk after you install the new 040 board. If your computer doesn't have an optical disk drive, the system will then try to boot through Ethernet.

In either situation, you'll need to stop the boot sequence and change the boot parameter so the SCSI disk will be the default boot device. To accomplish this, after the Testing System panel appears, press and hold down the right Command key and press the ~ key on the numeric keypad. This action takes the computer to the ªnmi>º prompt or, sometimes, back to the ROM monitor. Go to the ROM monitor from the ªnmi>º prompt:

```
    nmi> halt
```

This brings you to the ªNeXT>º prompt. From here type p, and you'll be prompted for a boot device. Continue by typing sd and pressing Return, then Control-D, and Return, as follows:

```
    NeXT> p
    Boot command: od? sd
    DRAM tests: yes? ^d
    NeXT>
```

You have now set the default boot device to the SCSI disk.

kernel name

When you boot from the ªNeXT>º prompt, the argument after the boot device is the kernel name.The default kernel name is mach. You rarely want to change this name. NeXT software developers will change the kernel name when testing a new version of the mach kernel. From the ROM monitor, general usage includes the optional kernel name:

```
    bxx [kernel]
```

Example:

```
    bsd mykernel
```

boot flags

Boot flags are used to specify boot options. The flags are preceded by a hyphen (-). When you specify boot flags, if no kernel name is given, the default mach kernel will be used. General usage when working from the ROM monitor includes adding the boot flags:

```
    bxx [kernel] -[flags]
```

flags include:

-a	Ask for the name of the root device.
-b	Boot without running /etc/rc.boot.
-s	Boot in single-user mode.
-i	Ask for the name of the init program.

-p   Don't auto-reboot on a panic.

Examples:

```
    bsd -s
    ben -bs
    bfd mymach -as
```

booting with boot flags
The boot flags are used to boot the computer in alternate modes not normally used. The uses range from asking the system to stop instead of rebooting after a system panic (bsd -p), to using boot recovery modes to recover from booting problems. Some common boot recovery problems and solutions are given here.

running fsck manually
Occasionally when the system is shut down brutally, for example when the power goes out, the computer   goes through an extended check of the system disk. When problems are too severe to be fixed automatically, you'll be asked to run fsck manually. You do this by booting with the -s flag and entering single-user mode. From single-user mode, you can run the fsck program manually and fix any disk errors. Here's a walk-through example:

Boot the computer in single-user mode, using the boot flag -s:

```
    NeXT> bsd -s
```

You'll see some verbose messages as the single-user system starts, ending with the cursor flashing at the Bourne shell ª#º prompt. If you don't make it this far, skip to the next section, ªkernel variables.º

From the ª#º prompt, start the fsck disk check with the command:

```
    #fsck /dev/rsd0a
```

If errors are found on the hard disk, you are prompted with yes-or-no questions. In answering ªyes,º you're telling fsck to do the best job it can in repairing the disk. The fsck command notifies you if it's deleting a file or if it's trying to recover that file. If a file is deleted, try to restore a backup copy. If the file is recovered, it may be moved to the /lost+found directory with the file name changed to a # sign followed by a random-looking number. The same is true of deleted or recovered directories.

Answering ªyesº to other questions like ªupdate block count?º or ªSalvage?º means the fsck command has calculated updated values for incorrect or missing file system data. Be warned that repairing the damage may cause a file and/or directory to be deleted. fsck will cut off a finger to save the hand. The error messages inform you of the consequences of the repair.

You must repeat the running of fsck until a clean run with no errors is reported. Occasionally after running fsck, you are prompted to ªreboot without sync.º To reboot without syncing the disk, use the following command:

```
    # reboot -n
```

When fsck runs without error, reboot in normal multiuser mode with the command:

```
    # reboot sd
```

kernel variables

The complete list of kernel variables may be unique to each kernel built. This section discusses variables that   will continue to be included in future releases and describes how they are used in various situations. General usage when working from the ROM monitor includes adding the kernel variables:

b*xx* [*kernel*] -[*flags*] [*kvars*]

where *kvars* includes the following kernel variables:

mem=*nnnn*	Use only *nnnn* KB of memory even though more is actually installed (for example, 1024 is 1 MB). Useful when measuring the impact of adding more (or less) memory.
init=*path*	Run *path* instead of /etc/mach_init.
rootdev=*xxx*	Use *xxx* (for example, sd1, od0, en0, fd0) as root device when boot device is different
rootdir=*path*	Use directory *path* as root point.
rootrw=1	Mount the root read/write initially. At boot, the root is normally mounted read-only. If fsck runs clean during rc.boot, it's remounted read-write.
cache=0	Disable the on-chip caches.
pagesize=4096	Use physical pages half as large (default is 8192). This usually makes things worse.

Examples:

`bod - rootdev=sd0` (- needed when kernel name not given)
```
bsd newmach -p mem=4096
```

when loading init or mach_init fails
Don't panic if you boot your computer in verbose mode and you see a message like:
ªload of /etc/init or /etc/mach_init failed, errno n.º

First look for the error code n. The following   are the most common codes and their meanings.

error number *n*	**meaning**
2	No such file or directory
83	Bad executable

The file /usr/include/sys/errno.h contains a complete list of codes. If you get error number 2, ªNo such file or directory,º check for the existence of the files /etc/init and /etc/mach_init. The best way to look for these files is to boot from an alternate device, like an optical disk or floppy disk. For more information on booting from an alternate device, see the next section.

If you don't have an alternate device to boot from, you can still try a
few things. The actual files init and mach_init are located in the directory /usr. The directory /etc is a link to private/etc. The files /private/etc/init and /private/etc/mach_init are actually symbolic links to ../../usr/etc/init and ../../usr/etc/mach_init, respectively. If the files can't be found in the paths /etc/init and /etc/mach_init, maybe they can still be found in /usr. The links for /etc or /private/etc/init or /private/etc/mach_init   could be missing. Here's an example of booting and repairing a disk when only the link for /etc is

missing:

```
    NeXT> bsd -bs init=/usr/etc/mach_init rootrw=1
```

The above command must be entered on a single line at the ªNeXT>º prompt. If the problem was a missing link for /etc, this   command brings you to single-user mode, and the disk will be mounted read/ write. At this point, you can look for and recreate, if necessary, the link for /etc. You should make sure that the directory /private/etc exists on the disk. If only the link is missing, use the following command to repair the link:

```
    # ln -s private/etc /etc
    # reboot sd
```

booting from an alternate device
An alternate boot device is any device the computer can boot from other than the default device. The most common alternate boot device on the NeXT computer is the optical disk drive. The NeXT release software is shipped on optical disk, and you can boot the computer from this disk with the bod command. The NeXTstation and NeXTcube come standard with a floppy disk drive instead of an optical disk drive. The 2.88 MB floppy disk has the capacity to contain enough software to boot the computer in single-user mode. You can create a boot floppy disk using the MkMagic script that is available through the NeXTedge Technical Support group. With a copy of the operating system on the optical or floppy disk, it's easy to boot from the ROM monitor with the bod and bfd commands respectively. Booting from an external device, which is more complicated, is explained in the following example, ªbooting from an external floppy disk drive.º The same principles hold for booting on any external SCSI device that isn't normally the boot device.

booting from an external floppy disk drive
When the computer won't boot from the internal hard disk, one way to try to examine

and repair the damage is to boot from an alternate device. This example assumes that you have an external floppy disk drive connected to the SCSI port, and a boot floppy disk created with the MkMagic script.

First, you must determine the   disk number of the external floppy disk drive. This can be determined from the SCSI ID or target numbers of your SCSI devices. The lowest SCSI target number is 0, and SCSI target 0 is known to the system as sd0, the normal boot device. The internal hard disk is shipped from the factory as target 1. The external SCSI devices should have SCSI target numbers between 2 and 6. The disk with the next-lowest SCSI target will be known to the computer as sd1, and so on. The internal SCSI controller is SCSI target 7. If an external SCSI device is re-set to be target 0, since this is lower than the target of the internal disk (target 1), the device set to be SCSI target 0 will become known to the computer as sd0, and the internal disk will be the next-lowest target number and will become sd1. The easiest way   to boot   from an external floppy disk drive may be to change the SCSI target number of the external floppy disk drive to 0 and boot with the command bsd as you normally would.

If it's not possible to change the target number of the external device to be lower than the internal disk, such as when the target number can't be changed on the external device, or the internal disk is already set to target 0, you can use an alternate approach. You must first use the information about how the SCSI target numbers map to the sdx numbers, the logical numbers used by the computer to reference the drive to each SCSI disk found on the bus. The computer will assign an sdx number when the computer starts. For example, if the internal disk is set to SCSI target 1, and the external floppy disk is the only other SCSI device on the computer and is set to target 2, the internal disk will be known to the computer as sd0, and the external floppy disk drive will be known to the computer as sd1, as shown here:

	SCSI target	**sdx number**
internal disk	1	sd0
external disk	2	sd1

General usage when working from the ROM monitor involves adding the device arguments:

```
    bxx [kernel][(ctrl,unit,part)] -[flags] [kvars]
```

where ctrl, unit, part are numbers determined as follows:

ctrl	Controller number, determined from SCSI target number. Lowest target is 0, next-lowest is 1, and so on.
unit	Unit number for the controller. Used when a SCSI device has one SCSI controller, therefore one SCSI target number, yet has more than one device. For example, a SCSI device with two floppy disk drives in one enclosure may have a unit 0 and a unit 1.
part	Partition number. This number represents the partition on the disk. Partition A will be 0, partition B will be 1, and so on.

Examples:

`bsd(0,0,0)`	
`bsd(1,0,0)`	Boot from SCSI disk sd1, first unit, partition A.

Here's an example of booting from an external floppy device when the internal disk is sd0, and the external floppy disk is sd1. Be sure to insert the MkMagic bootable floppy disk in the external floppy disk drive before you execute the following command:

```
    NeXT> bsd(1,0,0) - rootdev=sd1a
```

This will put you in single-user mode. If you're repairing a damaged hard disk, first you want to check the file system on the hard disk with the fsck command:

```
    # fsck /dev/rsd0a
```

Once you have run fsck on the disk with no errors, you can safely mount the hard disk and take a look at the files:

```
    # mount /dev/sd0a /HardDisk
```

If you are trying to recover from an error in loading /etc/init or /etc/mach_init, now you can examine the hard disk and check these files. If the files are corrupted, you can copy them from the boot floppy disk to the hard disk. Be sure that the MkMagic bootable floppy disk was built on the same release version as the version you are repairing on the hard disk. If the error code for ªLoad of /etc/init or /etc/mach_init failed, errno *n*º was 83, bad executable, you should also check that the shared library /usr/shlib/libsys_s.B.shlib isn't corrupted. You can use the sum command to check for corruption; run sum and compare the output. The init and mach_init programs are linked with the shared library.

from now on

Your best defense against booting problems is to be prepared. The NeXTedge Technical Support Hotline staff is ready to help you with any booting problems you may have. You should also be prepared with an alternate boot deviceÐa boot floppy disk, optical disk, or systemÐto act as a network boot server.

references

Network and System Administration manual, Chapter 11, ªSystem Startup and

Shutdown.º NeXT Computer, Inc., 1990.

UNIX manual pages for: fsck(8), reboot(2), and sum(1).