NEXTSTEP In Focus, Summer/Fall 1994 (Volume 4, Issues 3 and 4).

One PC, More Than One OS?

Joe Keenan

Now that the NEXTSTEP computer on your desk is an ªindustry-standardº Intel-based PC, why can't you use it to run other operating systems when you need them? Well, you canÐit just takes a little planning.

A One-Computer Solution

Configuring one computer to work with multiple operating systems can be very easy. You can install NEXTSTEP in an unused portion of a disk that already has DOS on it. Or you can set aside part of the disk (a *partition*) for DOS when you install NEXTSTEP, and install DOS afterwards. However, things get more complicated if you want to work with more than two operating systems, or if you require a more complex partitioning scheme.

This article explores schemes for partitioning a disk and installing multiple operating systems, including DOS, NEXTSTEP, OS/2, and Linux. It also mentions some important points for working with other operating systems.

Partitioning your disk

To work with multiple operating systems on a disk, you need to do a couple of things to it.

First, you need to format the disk. Normally, formatting is done by the manufacturer before the disk leaves the factory. IDE disks, in particular, are always preformatted. In addition, you can use most SCSI adapters to format a disk. (See the owner's guide for your hard disk, IDE disk controller, or SCSI adapter for more information.)

Second, you need to partition the disk. As the name implies, a partition is part of a disk. You can logically subdivide a disk into multiple partitions, for different uses. In this article, we're mostly concerned with using the different partitions for multiple operating systems.

Just to keep things interesting, there are two distinct partitioning schemes that NEXTSTEP can use: UNIX™-style and DOS-style. However, only one of them allows you to work with multiple operating systems.

You can divide a disk into multiple UNIX-style partitions. UNIX-style partitions are useful for dividing up a very large disk into more manageable portions. They're also useful for providing fine-grained access control over different parts of the disk. However, only NEXTSTEP can work with disks divided up into UNIX-style partitions (though you can use the disk with NEXTSTEP running on NeXT, PA-RISC, and other kinds of computers).

You create UNIX-style partitions with the **disk** command. (If you have a large disk, or if you want to create more than two partitions, you also need to create a custom **disktab** file. See the Spring 1994 issue of *NEXTSTEP In Focus* for more on **disktab** files.) The **disk** program also writes one or more **boot** programs on the disk (more about that later).

You can also divide a disk into DOS-style partitions. This is the partitioning scheme that DOS (and other PC operating systems) understand, and it's the one to use if you want to work with more than one operating system on a disk. The DOS partitioning scheme defines a maximum of four *physical* partitions on a disk. A physical partition can be designated as either a *primary* partition or an *extended* partition. An extended partition can be further subdivided into one or more *logical* partitions.

When you partition a disk you write a partition table on it. In the DOS-style partitioning scheme, the partition table has four entriesÐone for each of the four possible partitionsÐand each entry includes the information shown in Figure 1.

Partition number	**Partition status**	**Partition type**	**Size (in MB)**
1	0x00	0x06	80
2	0x80	0xA7	480
3	0x00	0x00	0
4	0x00	0x00	0

Figure 1: *Information from a typical DOS-style partition table*

The Partition Status field can be either 0x00 (inactive) or 0x80 (bootable, active). (The prefix 0x indicates that a number is a hexadecimal number.) This field tells the booter which partition to use by default (as described later in this article). The Partition Type field contains a code that identifies what the partition containsÐit's zero if the partition isn't in use. Each operating system vendor defines the partition types that they use (see Figure 2). Knowing these codes can be very useful when trying to use any of the more advanced disk-partitioning tools.

Partition type code	**Vendor**
0x00	Unused partition
0x01	DOS, 12-bit FAT
0x02	XENIX root
0x03	XENIX usr
0x04	DOS, 16-bit FAT < 32MB
0x05	Extended DOS
0x06	DOS, 16-bit FAT > 32MB
0x07	OS/2 HPFS, QNX
0x08	AIX file system

0x09	AIX, Coherent
0x0A	OS/2 Boot Manager
0x10	OPUS
0x40	Venix 80286
0x52	CP/M, Microport SysV/AT
0x63	GNU HURD
0x64	Novell 2.xx
0x65	Novell 3.xx
0x75	PCIX
0x80	Old Minix
0x81	Linux/Minix
0x82	Linux swap
0x83	Linux native
0x93	Amoeba
0x94	Amoeba bad block table
0xA5	386BSD
0xA7	NEXTSTEP
0xB7	BSDI
0xB8	BSDI swap
0xC7	Syrinx
0xDB	CP/M
0xE1	DOS access
0xE3	DOS R/O
0xFF	Bad block table

Figure 2: *Type codes revealed*

You create DOS-style partitions and edit the partition table with the **fdisk** (fixed disk) program. Versions of this program are supplied with NEXTSTEP, DOS, OS/2, Linux, and other operating systems that can use the DOS-style partitioning scheme. Using **fdisk** is a matter of deciding how many partitions you want, what kinds, and how big each should be. You can use any of the **fdisk** programs tested for this article to create primary and extended partitions. You can use the DOS, OS/2, and Linux versions to create logical partitions, as well. The boot partition for both NEXTSTEP and DOS must be a primary partition, while the boot partition for OS/2 and Linux can be a logical partition.

NEXTSTEP (and many other operating systems) can boot only from partitions that are contained entirely in the first 1024 cylinders of a disk. This can be a problem for large disks, unless the SCSI adapter or hard disk controller does sector translation. See the owner's guide for your adapter or controller for more information.

Understanding the boot process

Booting is the incremental process of loading successively larger and more complex programs from a disk until, finally, your operating system is up and running. You need a basic understanding of the boot process to determine the appropriate partitioning scheme for your computer. First we'll describe the simplest boot sequence and then describe more advanced situations and the programs (called *boot managers*) you can use to control the process.

Booting DOS

If you have an unmodified copy of DOS on your hard disk, it starts up (or *boots*) in the following sequence:

1. When you turn on the computer, its ROM BIOS (the basic startup programs in chips in the computer) executes a power-on self-test sequence. If there are no errors, the BIOS loads cylinder 0, head 0, sector 1 of the primary disk into memory. Sector 1 must contain the partition table and partition booter, also known as the master boot record (MBR). If the sector has a valid MBR, then the BIOS executes the partition booter.

2. The partition booter examines the partition table and determines which partition is ª active.º If there's no active partition, the boot process stops. Otherwise, the first sector of the active partitionÐthe boot recordÐis loaded into memory and executed.

3. The DOS boot record reads the DOS hidden files (**IO.SYS** and **MSDOS.SYS**) into memory. **IO.SYS** reads the **CONFIG.SYS** file and loads device drivers, sets up low-level

data structures, and so on. **MSDOS.SYS** loads in **COMMAND.COM** (or another command processor), according to the **CONFIG.SYS** information loaded by **IO.SYS**.

4 **COMMAND.COM** executes the commands in **AUTOEXEC.BAT** and then displays the DOS prompt. If you use Windows, **AUTOEXEC.BAT** usually starts up Windows. That completes the DOS boot sequence.

When there's only one operating system on a disk, there isn't much need for a boot manager program. The MBR just looks at the partition table for the active partition and boots that one. No muss, no fuss. However, with more than one bootable partition on a disk, this doesn't work very well. That's where boot manager programs come in.

Booting NEXTSTEP

The NEXTSTEP boot manager functions almost the same as the DOS partition booter. The only difference is that the NEXTSTEP booter displays the prompt:

```
Type n for NEXTSTEP, d for DOS, 1-4 for partition #:
```

Then it waits for you to type a single character. If you don't type anything, it times out and does exactly the same thing that DOS does: It looks for the active partition (step 2 above) and continues the boot sequence.

If you type **n** at the prompt (or the number of a NEXTSTEP partition), the NEXTSTEP boot manager examines the partition table for a partition with the NEXTSTEP partition type (0xA7 from Figure 2) and selects that as the active partition. If you type **d** at the prompt, the boot manager does the same for the DOS partition types, and you're back at step 2 above.

If you select a NEXTSTEP partition, the boot sequence is considerably more complicated than for DOS. The NEXTSTEP boot sequence goes something like this:

1 The computer's ROM BIOS executes a power-on self-test sequence and loads cylinder 0, head 0, sector 1 of the primary disk into memory. Sector 1 contains the partition booter.

Sector 1 is actually the first sector in a track. The PC BIOS (and DOS) start counting at 0 for cylinders *and heads, but at 1 for sectors. Ours is not to wonder why....*

2 The partition booter (also known as **boot0**) determines the active partition, as described above. The first sector of the partition (the ªboot record,º also known as **boot1**) is loaded into memory and executed.

*If you have a NEXTSTEP-only disk, **boot0** isn't used. Instead, the **boot1** program is stored in the partition table sector on the hard disk. That means that there's no DOS-style partition info on the disk. That's OK, because NEXTSTEP doesn't need it.*

3 **boot1** is a ªsilentº booter. It usually doesn't display anything to the user. Its sole purpose is to load the **boot2** program from the NEXTSTEP partition.

4 The real NEXTSTEP boot program is **boot2**. This is the program that prompts you to type startup options at the **boot:** prompt. It knows how to put the display hardware into VGA graphics mode to display the fancy startup screens. **boot2** loads the operating system kernel (**mach_kernel** in the root directory on the disk) and passes it any options you typed at the **boot:** prompt.

5 **boot2** also loads the device drivers the kernel needs for initial system startup. Up to this point, all system I/OÐdisk, keyboard, and displayÐhas been done using BIOS routines. Before NEXTSTEP can go into virtual-memory, multitasking mode, **boot2** has to load device drivers the kernel needs to get up and running. These drivers are specified in the ªBoot Driversº line in the **Instance0.table** file. in **/private/Drivers/i386/System.config/**. (If you haven't configured NEXTSTEP yet, it uses **Default.table** file, in the same directory.)

6 Once **boot2** has loaded the drivers the kernel needs, the kernel can switch to virtual-memory, multitasking mode, and start up all other necessary processes. Additional device driversÐspecified in the ªActive Driversº line in **Instance0.table**Ðare loaded at this time.

7 The remainder of the startup sequence is fairly standard for a UNIX operating system. (For more information, see Chapter 9 in the *NEXTSTEP Network and System Administration* manual).

Restoring or Replacing the NEXTSTEP Partition Booter

If the NEXTSTEP booter gets replaced with the DOS booter, you can't start NEXTSTEP. To fix this, start up DOS and use **fdisk** to make the NEXTSTEP partition active. Then restart the computerÐ NEXTSTEP will start up automatically. Log in as root and reinstall the NEXTSTEP boot program by typing the following command in a Terminal window:

```
# /usr/etc/disk -b /dev/rsd0h
```

Substitute `rhd0h` for `rsd0h` with an IDE drive.

By the wayÐif you ever want to get rid of the NEXTSTEP booter and restore the DOS one, start up DOS and type the following command at the DOS prompt:

```
C> fdisk /mbr
```

Using Multiple Disks

Another option is to use multiple disks. NeXTanswer #1487 (see ªTips & Techniquesº for how to use NeXTanswers) describes a method of installing NEXTSTEP on a secondary hard disk. You'll still need to figure out the partitioning issues, because you'll need a small partition on the primary disk for the NEXTSTEP kernel, unless you want to start up from a floppy disk.ÐJK

The OS/2 boot manager

The OS/2 boot manager is a little different than the DOS and NEXTSTEP booters in that it uses a primary partition of its own, which is always set as the active partition. This allows it to use a more complex program than can fit in a normal boot sector and to store additional information about the various partitions that it displays at boot time. For instance, the boot manager displays a description string for all of the available (bootable) partitions it knows about, and allows you to set the delay time before the default partition is activated.

The OS/2 boot manager has one more capability that's significant if you want to have more than two operating systems on your disk. You can install OS/2 into a DOS logical partition, instead of a physical partition as required by DOS and NEXTSTEP. Why is this important? Well, there's another quirk to DOS and OS/2 that comes into play.

For DOS and OS/2, only one primary partition can be ªvisibleº at a time. This means that if you have a disk with three primary partitions (NEXTSTEP, DOS, and OS/2), then DOS will not be able to see the OS/2 partition, and OS/2 will not be able to see the DOS partition. Neither will see nor understand the NEXTSTEP partition, but that's not surprising. NEXTSTEP will see the DOS partition, because it doesn't follow DOS rules.

So if you want all three operating systems on a disk *and* you want to be able to access them from each other, you need to put OS/2 in a logical partition. You won't be able to access the OS/2 partition from NEXTSTEP, but DOS will, and OS/2 will see the DOS partition.

If you do decide to use the OS/2 boot manager, you'll probably want to delete the NEXTSTEP booter. If you don't, you'll be prompted twice to pick an operating system or partition. The first time is by the NEXTSTEP booter. If the active partition is the OS/2 boot manager, and you don't explicitly select NEXTSTEP or another partition, then the OS/2 boot manager presents you with a second list of bootable partitions.

LILO: The Linux loader

LILO is the boot loader from the Linux operating system suite. Although not as polished in appearance as, say, the OS/2 boot manager, it has many features and capabilities the others don't have. Like the OS/2 boot manager, LILO allows booting from either primary or logical partitions, and it allows you interactively to select the boot partition.

Just as you must be running OS/2 to use the OS/2 boot manager, you must be running Linux to configure LILO. You can either use the interactive configuration program to set up the LILO configuration file, or you can edit the file by hand.

Installing Multiple Operating Systems

Now that you know lots of trivia about how multiple operating systems work on a disk, you need to know how to put them there. Two, three, four, or even more operating systems? It can be done. Here's how.

NEXTSTEP and DOS

The simplest case of multiple operating systems on one disk is NEXTSTEP and DOS. They're easy to install, since this is the default installation scenario for NEXTSTEP. The most important thing to remember is that DOS is already taking up the entire disk; you have to erase the DOS partition. If you don't, you're guaranteed to have serious problems later onÐso be forewarned.

Boot DOS from a floppy and run **fdisk** to create just the DOS partition. Leave the rest of the disk unallocated. Then install DOS normally.

Once you've installed DOS, restart the computer with the NEXTSTEP installation disks and install NEXTSTEP (this is described in the *Installing and Configuring NEXTSTEP* book). After you select which disk to install NEXTSTEP on, you're asked how you want to partition the disk. Select the option to install NEXTSTEP in the unused space on the disk. Then finish installing NEXTSTEP.

You can also install DOS after you install NEXTSTEP. In this case, make sure you leave room for a DOS partition during the NEXTSTEP installation procedure. If you don't, you'll have to start all over again. Once NEXTSTEP is installed, follow the instructions that come with DOS for initial installation. Don't repartition the disk, and do make sure you select the partition that you reserved for DOS when you installed NEXTSTEP. You should be able to identify that partition by its size, or by the order in which you created the partitions.

If you do install DOS after NEXTSTEP, the DOS installation procedure usually replaces the NEXTSTEP partition booter with its own. This makes it impossible to start up NEXTSTEP. To fix this, restore the NEXTSTEP partition, as described in ªRestoring or Replacing the NEXTSTEP Partition Booter.º

NEXTSTEP, DOS, and OS/2

Here's where things start to get interesting. Let's assume that you have a 1-gigabyte disk and you want to install these three operating systems on it. You decide you need about 600 megabytes for NEXTSTEP, 200 megabytes for DOS, and 200 megabytes for OS/2. (Your mileageÐand numbersÐmay vary.) How to partition the disk?

Well, you know that you can't put both OS/2 and DOS in primary partitions if you want to be able to access the DOS partition from OS/2, and vice versa. This means that you need to install OS/2 in a logical partition, and you need to install the OS/2 boot manager so you can boot the OS/2 partitionÐthe NEXTSTEP booter can't do it. So now you need a boot manager partition as well. For this, use the partitioning scheme shown in Figure 3.

No.	Type	Operating system	Size(MB)	Label
1	Primary	DOS	200	C:
2	Primary	NEXTSTEP	600	Ð
3	Extended	Ð	Ð	Ð
	Logical	OS/2	199	D:
4	Primary	BootMgr	1	Ð

Figure 3: *A basic partitioning scheme for NEXTSTEP, DOS, and OS/2*

You have a considerable amount of flexibility here. You're stealing a little bit of disk space for the OS/2 boot manager from the OS/2 partition, but you could just as easily have taken it from one of the others. Also, depending on the size of your disk, your minimum partition size may be larger than 1 megabyte.

If you want to split up your DOS or OS/2 partitions into smaller pieces, you can make partition 1 smaller and split partition 3 into multiple logical partitions, as shown in Figure 4. This allows you to set up dedicated data partitions for use in DOS and OS/2.

No.	Type	Operating system	Size(MB)	Label
1	Primary	DOS	100	C:
2	Primary	NEXTSTEP	600	Ð

3	Extended	Ð	Ð	Ð
	Logical	OS/2	99	D:
	Logical	Data 1	100	E:
	Logical	Data 2	100	F:
4	Primary	BootMgr	1	Ð

Figure 4: *An alternative partitioning scheme for NEXTSTEP, DOS, and OS/2*

Don't forget that in either case, you can only access the first DOS primary partition from NEXTSTEP. You can't access subsequent primary partitions or any logical partitions.

To install all three operating systems, begin by following the instructions for the simple NEXTSTEP and DOS case, but remember to make the first two partitions the right sizes. Leave the rest of the disk unallocated. Install NEXTSTEP first, and then DOS. With three bootable partitions, you won't be able to use the NEXTSTEP booter, so don't bother with the gyrations required to put the NEXTSTEP booter back in place. (If you've already installed it, use the DOS **fdisk /mbr** command to get rid of it.)

Now it's time to install OS/2. Follow the normal OS/2 installation procedures, but use the advanced partitioning options to create the extended partition and its logical partitions. While the OS/2 **fdisk** program is running, don't forget to install the OS/2 boot manager and add the NEXTSTEP and DOS partitions (make them bootable).

You can install OS/2 either on a DOS file system or on an OS/2 HPFS file system. If you install OS/2 on an HPFS file system, you won't be able to access the OS/2 partition from DOS. See the OS/2 installation manual for details.

Solaris 2.1

If you're thinking about installing Solaris on your system, there are a couple of things to be aware of. First, Solaris requires its own primary partition. It puts all the individual file systems it needsÐ**/**, **/usr**, **/opt**, **swap**, and so onÐin that primary partition. Also, if the Solaris partition is set active, it displays an option screen when it starts up to allow you to select one of the other partitions to boot. If you don't select another partition, it completes the Solaris boot sequence.ÐJK

NEXTSTEP, DOS, OS/2, and Linux

All four? Well, yes, if you have a big enough hard disk.

Actually, this is pretty much the same as the NEXTSTEP, DOS, and OS/2 case, except that you have a new option to consider. You can use LILO as your boot manager instead of the OS/2 boot manager, but there's a trade-off: OS/2 is more verbose and gives you more information about your boot options, while LILO doesn't take up an extended partition of its own.

Install the operating systems as before, reserving a partition for Linux. A basic Linux installation requires 40 to 80 megabytes of disk space (about the same as required for OS/2). One possible partitioning scheme is shown in Figure 5.

No.	Type	Operating system	Size(MB)	Label
1	Primary	DOS	100	/dev/sda1
2	Primary	NEXTSTEP	600	/dev/sda2
3	Extended	Ð	Ð	/dev/sda3
	Logical	OS/2	100	/dev/sda5
2	Logical	Linux	100	/dev/sda6
2	Logical	Data	99	/dev/sda7
4	Primary	BootMgr	1	/dev/sda4

Figure 5: *A partitioning scheme for NEXTSTEP, DOS, OS/2, and Linus*

The last column in Figure 5 is the name by which Linux refers to each partition. The primary partitions are **/dev/sda1** through **/dev/sda4**, and the logical partitions go up from there. (These names are for a SCSI disk. If you're using an IDE disk, you'll have **/dev/hda1** through **/dev/hda4** instead.) Linux can access the raw data blocks in each partition, but it can only read Linux and DOS partitions as real file systems.

The Sky's the Limit

You have a lot of flexibility in your operating system and partitioning options. Just take the time to plan your strategy and you'll be able to switch OSs with relative ease.

Joe Keenan is a member of NeXT's Technical Support Team working out of the McLean, Virginia, office. He specializes in large network support for NeXT's commercial and government customers. You can reach him by e-mail at ***jkeenan@next.com****.*