

CHAPTER 3

Advanced Installation Tasks

Chapter 2, “Installing FreeBSD,” discussed a standard installation without complicating factors such as dual-boot disk configurations and coexistence with other operating systems. It also assumed that you were installing FreeBSD from a CD-ROM rather than via one of the many alternative installation media available.

This chapter serves mostly as a supplement to Chapter 2, by explaining a number of essential preinstallation tasks for those whose systems don’t fit the previously mentioned assumptions. If you are installing FreeBSD on a system with another operating system previously installed, or if you aren’t using a CD or DVD to install FreeBSD, you’ll want to read this chapter before performing the installation. After you have completed the necessary preinstallation tasks you learn here, you should return to Chapter 2 and proceed with the normal installation.

Many new users who are curious about FreeBSD install it on a workstation that is already running Windows. This chapter shows how to prepare for a FreeBSD installation that will enable the two systems to operate side by side. We’ll also look briefly at installing FreeBSD on a system that is running Linux. After reading this chapter, you will know how to install FreeBSD in such a way that you can choose at startup time whether to boot your computer into FreeBSD or into Windows or Linux.

In addition, we’ll briefly cover NFS and FTP network installs for situations where an installation CD or DVD is not available.

IN THIS CHAPTER

- Backing Up an Existing Windows or Linux Filesystem
- Nondestructive Hard Disk Partitioning with FIPS
- Working with FIPS
- Potential Problems with and Limitations of Dual-Boot Systems
- The FreeBSD Boot Manager
- Alternate Installation Methods

Backing Up an Existing Windows or Linux Filesystem

Before you go any further, back up any existing Windows or Linux filesystem you want to maintain. Although the next section shows you how to nondestructively create space for FreeBSD, mistakes can still happen, and programs can cause errors. It is best to have a backup of anything you want to keep—it's small comfort, after losing your precious data, to reflect that at least you saved yourself the time and cost of a backup solution.

Media that can be used for backup purposes include recordable CDs or DVDs, Zip or Jaz disks, tape drives, externally mounted USB or FireWire hard drives, or even floppy disks, if the amount of data you need to save is small. Keep in mind that backing up the operating system and installed applications is not necessary because they can be easily reinstalled as long as you still have the installation discs. The primary things to worry about are your files containing data that cannot easily be replaced. (These days, of course, a large MP3 collection or a photo library can easily necessitate that you buy a cheap external hard drive and copy them onto it for safekeeping.)

If you do not have an actual backup program, you can use an archiving program such as WinZip to help you compress data for backup and also to place that data onto disks. You also might find a program called Partimage, available at <http://www.partimage.org>, to be helpful in this capacity.

NOTE

The exact procedures for doing a system backup are beyond the scope of this book. See the documentation for the backup program you plan to use for information on how to complete the backup.

After you have backed up any existing files that you want to keep, you will need to free up some space on your hard disk for installing FreeBSD. There are a few ways that this can be done:

- *Simply delete the partitions and start over.* This causes you to lose all the existing data on your system and you will have to reinstall everything that is currently on your system once you have re-created the partitions. This is a desirable option when you don't really care what's on the disk, or if you don't have a nondestructive partitioning program such as Partition Magic—and if you aren't daunted by the idea of reinstalling Windows or Linux after you have made space for FreeBSD and installed it.
- *Use a commercial nondestructive partitioning program such as Partition Magic.* If you have Symantec's Norton Partition Magic, by all means use it. This \$69 solution is the "best of both worlds," allowing you to reallocate the space on your hard disk for a FreeBSD installation without losing any of your data or configurations. A discussion of how to use this commercial program is beyond the scope of this book. See the Partition Magic documentation for instructions. Partition Magic's website is http://www.symantec.com/home_homeoffice/products/system_performance/pm80/. There's also a non-commercial, open-source clone of Partition Magic called QtParted, available at <http://qtparted.sourceforge.net/>.

- *Use the FIPS utility.* This freely available program, written by Arno Schäfer, allows you to split an existing partition to create free space. FIPS is included on the DVD with this book, and it is the method we'll discuss in this chapter. It's not as polished a solution as Partition Magic, but it's free and can give you similar results.

Nondestructive Hard Disk Partitioning with FIPS

FIPS, which stands for *First (nondestructive) Interactive Partition Splitting*, is a partitioning program designed to run under DOS or Windows in DOS mode. It splits an existing DOS partition into two at the point you specify. You can then use the new partition it creates as the space for FreeBSD. Note that FIPS works only with DOS-style partitions (FAT16 or FAT32). FIPS does not work with Windows NT/2000/XP NTFS partitions, nor does it work with Linux Ext2FS/Ex+3FS partitions. Finally, those running OS/2 need to be aware that FIPS does not work with HPFS partitions, either.

FIPS has a couple of limitations:

- *It cannot split an extended DOS partition, only a primary one.* Extended DOS partitions are forms of second-level partitioning in common use in the DOS/Windows and Linux worlds; they allow a disk to contain more than the BIOS-supported four partitions by subdividing them and then providing access to them as though they were regular partitions. (FreeBSD doesn't use extended partitions; instead, it has the concept of *slices* and *BSD partitions*, which we will cover in Chapter 20, "Adding Hard Disk Storage.") If you are like most people running Windows, this will not be a problem—you likely have one primary partition that takes up the entire disk.
- *You cannot currently have more than three partitions on your disk.* FIPS creates a new primary partition with the free space it is assigned. Because you can only have four DOS-style partition entries on a disk, you can have no more than three existing partitions when you run FIPS. Note that this limitation only applies to IDE/ATA drives; owners of SCSI drives won't have this limitation.

If neither of these issues applies to you, and assuming you have made a backup, you are ready to begin the partitioning process.

CAUTION

Windows XP partitions are often formatted as NTFS. Because of this, FIPS may not work with Windows XP at all, even for Windows XP Home Edition. If you're using Windows XP or NT/2000 with an NTFS filesystem, you will need to look into the commercial Partition Magic for nondestructive disk partitioning.

Running ScanDisk and the Defragmenter

Before you use FIPS, you should first turn off your screensaver and any scheduled tasks that might be configured, and then run DOS or Windows ScanDisk to fix any problems on the disk. After ScanDisk has finished running, you need to run the disk defragmenter.

FIPS needs contiguous free space at the end of the drive in order to split the partition. It cannot split before the last sector on the disk containing data. Running the disk defragmenter moves all the data to the beginning of the disk without leaving holes in the middle.

CAUTION

Be aware that the defragmenter in Windows 2000 and XP often fails to move all files into a contiguous chunk; the NTFS filesystem tends to prevent this from working correctly, and defragmenting tools often leave large holes in the midst of the used disk space. Again, Partition Magic may be the best solution for these platforms if FIPS fails to find enough contiguous free space.

Depending on the speed of your computer, the size of your hard disk, how fast the disk is, and how badly fragmented it is, the defragmentation process could take anywhere from a few minutes to several hours.

Obtaining FIPS and Creating a Boot Disk

When the defragmentation process is finished, you are ready to start FIPS. You can download FIPS from the FreeBSD FTP server at <ftp.freebsd.org> or one of its mirror sites in the directory `/pub/FreeBSD/tools/fips.exe`. You can download `restorrb.exe` from the same directory. If none of these locations are available, the official FIPS website is <http://www.igd.fhg.de/~aschaefe/fips/>.

You should create a bootable floppy disk and copy the three files mentioned previously to it. This floppy disk will allow you to boot into MS-DOS (or FreeDOS—see <http://freedos.org/> for more information) and run FIPS, but nothing else. In DOS or Windows 98/Me, you can create a bootable floppy from a DOS prompt with the command `format a: /s`, assuming that you have a blank floppy in drive A. The following is a sample procedure for creating the boot disk:

```
C:\> format a: /s
```

```
Insert new diskette in drive A:  
and press ENTER when ready...
```

```
Checking existing disk format.  
Verifying 1.44M  
Format complete.  
System transferred
```

Volume label (11 characters, ENTER for none)?

```
1,457,664 bytes total disk space
 388,608 bytes used by system
1,069,056 bytes available on disk
```

```
512 bytes in each allocation unit.
2,088 allocation units available on disk.
```

Volume Serial Number is 031B-0831

Format another (Y/N)? **n**

```
C:\> d:
D:\> cd tools
D:\TOOLS> copy fips.exe a:\
      1 file(s) copied
d:\TOOLS> copy restorerb.exe a:\
      1 file(s) copied
D:\TOOLS> copy errors.txt a:\
      1 file(S) copied
D:\TOOLS>
```

NOTE

You cannot use `format a: /s` to create a bootable floppy under Windows 2000 or XP. These newer operating systems require you to use the Windows interface to create boot floppies. In Windows XP, insert the disk, and then browse to it under My Computer; right-click the disk and choose Format. Enable the Create an MS-DOS Startup Disk check box, and then click Start.

When you've created the bootable floppy, use it to reboot your system (you may need to make sure your BIOS is configured to boot from the floppy drive if a disk is present). After the system has finished booting, type **fips** at the DOS prompt to start the FIPS program.

Working with FIPS

When FIPS first launches, it gives you a warning not to use it in a multitasking environment, among other common-sense caveats. After you have read all the information, press any key to continue. If you have more than one hard disk in your system, FIPS will ask you which one you want to work on. Select the disk you want. FIPS will then show you the partition table of your disk. It will look something like the following:

Part.	bootable	Head	Cyl.	Start Sector	System	Head	Cyl.	End Sector	Start Sector	Number of Sectors	MB
1	yes	1	0	1	06h	12	983	32	32	409312	4149
2	no	0	0	0	00h	0	0	0	0	0	0
3	no	0	0	0	00h	0	0	0	0	0	0
4	no	0	0	0	00h	0	0	0	0	0	0

Checking root sector ... OK

Press any Key

If you have more than one partition on your disk, FIPS asks you to select which one you want to split. If you only have one partition on your disk, you are asked to press any key, as in the preceding example. After you have selected a partition or pressed a key to continue, FIPS reads the boot sector and presents some more information on the disk:

```

Bytes per sector: 512
Sectors per cluster: 8
Reserved sectors: 1
Number of FATs: 2
Number of rootdirectory entries: 512
Number of sectors (short): 0
Media descriptor byte: f8h
Sectors per FAT: 200
Sectors per track: 32
Drive heads: 13
Hidden sectors: 32
Number of sectors (long): 409312
Physical drive number: 80h
Signature: 29h
    
```

When FIPS has finished presenting information, it shows you the size of the new partition and the size of the old partition, and asks you to choose the starting cylinder that divides them. Use the left- or right-arrow key to decrease or increase the number the new partition will start on. In addition, you can use the up- or down-arrow key to increase or decrease the size of the new partition in increments of 10 cylinders. You can use the “Start” and “End” numbers of heads, cylinders, and sectors to gauge how you want to divide up your disk; you can decrease the size of the existing partition(s) until the remaining space is big enough for your FreeBSD installation (use the Size column to determine how big it is). Note that, as we’ll discuss shortly, a FreeBSD boot partition must exist entirely within the first 1,024 cylinders of the disk. When you have finished, press Enter to continue.

TIP

Write down the starting cylinder information for the new partition you create. This will help you verify later on during the FreeBSD installation that you have selected the correct partition on which to install FreeBSD.

After you have pressed Enter, FIPS shows you what the new partition table will look like. It then gives you the option to re-edit the partition table or continue. If you select Continue, FIPS asks you one last time if you are sure you want to write the changes to the partition table. Be careful at this point! Selecting Y will cause FIPS to write the changes and then exit. This is an irrevocable change; if you write the changes, you won't be able to revert back to the old configuration. After you commit the changes and exit FIPS, you need to reboot your system immediately.

If FIPS exits with any errors, see the README file included with FIPS for more information.

CAUTION

It is very important that after you exit FIPS, you do *not* create any new data on the hard disk until after you have rebooted. Doing so could corrupt the disk because DOS will not be aware that the partition table has changed until the system has been rebooted.

After you have rebooted, you should run FIPS again with the `-t` option. This will check to make sure the partition was split correctly. If errors are reported, restore the previous partition table by running `RESTORRB.EXE` and then reboot again.

CAUTION

After you have made *any* changes to the filesystems on the disk, you can no longer use `RESTORRB.EXE` to restore the old partition table. Therefore, it is very important that you run `fips` with the `-t` option after you reboot before you do anything else.

If `fips -t` doesn't report any errors, remove the floppy from the drive and then reboot again. When Windows or DOS has finished restarting, you should run ScanDisk on the partition you split to check for any errors.

Potential Problems with and Limitations of Dual-Boot Systems

It is possible to have a dual-boot system in which you have two (or even more) operating systems on the hard disk, and you can select which one you want at each system boot. This is a very useful configuration in many circumstances; if you only have one computer, and you want to use FreeBSD for programming and Windows for gaming, dual-booting is an ideal solution. Working with dual-boot systems can present some problems, however, if you don't take steps to avoid and/or deal with them.

First, be aware that all the information necessary to boot FreeBSD must be located within the first 1,024 cylinders of the hard disk. This is necessary for the FreeBSD boot manager to work; it means that when you partition the disk for FreeBSD using FIPS, either the root partition must be completely located within the first 1,024 cylinders, or you can use a separate boot partition that is completely located in the first 1,024 cylinders. (The crux of the matter is that the BIOS is only equipped to look for a bootable partition—one that contains `/boot`—in that first segment of the disk.) Use the “Start” and “End” cylinder readouts in FIPS to determine where your partitions start and end. If you choose the latter option (a bootable partition other than the root partition), the root partition does not have to be completely located in the first 1,024 cylinders. Note that “completely located” means that the partition has to both start and end below the 1,024th cylinder. Simply starting below the 1,024th cylinder is not good enough.

If you need more space for Windows or DOS than is available below 1,024 cylinders, you can use FIPS again to split the Windows or DOS partition into two partitions, giving you a C: drive and a D: drive in Windows or DOS. In between these C: and D: drives, you need to install a small partition for FreeBSD to boot from. This partition will be used as `/boot` later on during the install (30MB should be more than enough for this partition).

Second, when you reinstall programs after partitioning, be certain to install DOS or Windows before you install FreeBSD. DOS and Windows assume that they are the only operating system on the hard drive and will overwrite the master boot record without asking. If you install FreeBSD first, installing DOS or Windows later will “clobber” FreeBSD’s boot manager, and you will no longer be able to boot into FreeBSD. This problem is easily fixed, but save yourself the headaches and just install DOS or Windows first.

Dual-Booting with DOS, Windows 95, Windows 98, Windows Me, or Windows XP

FreeBSD comes with a boot manager that allows you to dual-boot with various operating systems. Simply select the “Install the FreeBSD Boot Manager” option during installation. Subsequently, each time you boot your system, you’ll be given a menu allowing you to choose between FreeBSD and your previously installed version of MS-DOS or Windows, by pressing the appropriate F-key.

TIP

GAG is an alternative, third-party boot manager that’s graphical and even more flexible than FreeBSD’s. Visit its website at <http://gag.sourceforge.net/>.

Dual-Booting with Linux

If you want to dual-boot with Linux and load Linux from the FreeBSD boot manager, install LILO or GRUB (Linux boot managers) at the beginning of your Linux boot partition rather than in the master boot record (MBR). See the LILO or GRUB documentation

for instructions on how to do this. After you have installed the Linux boot manager, you can boot Linux from the FreeBSD boot manager.

If you want to boot FreeBSD from LILO or GRUB, the following subsections explain how.

Booting FreeBSD from LILO

If you're running Linux and want to boot FreeBSD from Linux's LILO loader, it is fairly easy to do so. In Linux, edit the file `/etc/lilo.conf` and add the following lines:

```
other=/dev/hda2
  table=/dev/hda
  label=FreeBSD
```

Change the line beginning with `other=` to reflect whatever device name Linux uses to identify your FreeBSD drive. (Remember, Linux's device names, including hard drive names, are significantly different from FreeBSD's naming convention.)

After you have changed the configuration file, reinstall LILO by typing `lilo` as the root user.

Booting FreeBSD from GRUB

To dual-boot Linux and FreeBSD using the GRUB boot manager, edit `/boot/grub/grub.conf` and add the following lines:

```
title FreeBSD 6.1 #1
  root (hd0,0,a)
  kernel /boot/loader
```

The arguments in front of the `root` statement indicate where FreeBSD's root partition is located; in the examples shown previously, that has been in BSD partition `a` of slice `0` on the primary hard disk, thus `(hd0, 0, a)`. If your FreeBSD installation uses different disks and slices, adjust these numbers accordingly.

Refer to <http://geodsoft.com/howto/dualboot/grub.htm#freebsd> for more information on using GRUB along with Linux and FreeBSD.

The FreeBSD Boot Manager

You can install the FreeBSD boot manager during the FreeBSD installation to enable booting of multiple operating systems. After the install, you can configure the boot manager with the `boot0cfg` program.

`boot0cfg -B` will install the boot manager onto the hard disk's MBR. This is one way to restore the boot manager if Windows should wipe it out (if, for example, you install Windows after you install FreeBSD). Of course, you would have to boot from a FreeBSD boot disk to use `boot0cfg -B` if the boot manager has been wiped out. In addition, if you want to make changes to the boot manager configuration, you will need to reinstall it using this command, followed by the changes you want to make.

`boot0cfg` is command-line based. Fortunately, you probably do not need to be concerned with most of the options, although you might find them interesting in an academic sense, if nothing else. The options shown in Table 3.1 are the most important ones for making changes to the boot manager configuration.

TABLE 3.1 Boot Manager Configuration Options

Option	Description
<code>-v</code>	<code>boot0cfg</code> will be more verbose about what it is doing.
<code>-b image</code>	Here, <i>image</i> is the name of the boot image to use. The default is <code>/boot/boot0</code> .
<code>-d drive</code>	Here, <i>drive</i> is the drive number used by the PC's BIOS for referencing the disk. Usually this is <code>0x80</code> for the first drive, <code>0x81</code> for the second, and so on.
<code>-f file</code>	Here, <i>file</i> is the name of a file to which the original MBR should be backed up in case there are problems. If the file already exists, it will be replaced with the new version.

The `-o` option is also supported, and it allows a comma-separated list of arguments, whose meanings are listed in Table 3.2.

TABLE 3.2 Arguments for `boot0cfg -o`

Argument	Description
<code>packet</code>	If the PC's BIOS supports it, this will tell <code>boot0cfg</code> to use <code>int 0x13</code> extensions instead of the old-style CHS (Cylinder-Head-Sector) for disk I/O. This will get around the 1,024 cylinder boot limit described previously. However, if the PC's BIOS does not support this option, it may cause the system to hang on the next reboot.
<code>setdrv</code>	Forces <code>boot0cfg</code> to use the drive number specified in the <code>-d</code> option to reference the drive you're working with.
<code>noupdate</code>	By default, the boot manager can write to the MBR and update it (to set the active flag, primarily). This can cause problems if you have hardware antivirus support enabled that prevents writing to the MBR. The <code>noupdate</code> option will prevent the boot manager from attempting to write to the MBR.

`boot0cfg` also supports the `-s n` option, where *n* is a number from 1 to 5 that specifies the default slice (commonly referred to as a *partition* in MS-DOS/Windows) to boot if no selection is made. The `-t n` option is also supported, where *n* is a number representing the number of "ticks" to wait before booting the default operating system. There are approximately 18.2 ticks in a second.

Alternative Installation Methods

If you can't or (for whatever reason) don't want to install FreeBSD from the DVD included with this book, you have several other options available. These include network installs with FTP or NFS. Installation over the network is especially convenient when you are in a location with a lot of bandwidth and don't have the DVD handy, or if the machine

doesn't have a working DVD drive or a BIOS that can boot from it. For many servers or workstations in enterprise environments, installing from a network source is the quickest and most efficient way to get FreeBSD installed.

Installing FreeBSD over the network requires that you first boot a minimal version of FreeBSD that contains the kernel, the Sysinstall program, and almost nothing else. If you can't boot into Sysinstall from the installation DVD, you have to use boot floppy disks. Create these three disks as described in the "Creating Installation Floppies" section of Chapter 2. Then put the first disk (boot.flp) into the floppy drive and boot the computer, inserting the other two disks when requested.

Installing FreeBSD over FTP

FTP stands for *File Transfer Protocol*. It is one of the oldest methods of transferring files over the Internet, from one system to another. Despite being unencrypted and insecure, FTP is still widely used; it's the backbone of file transfers on the Internet, as we will discuss in Chapter 27, "Configuring an FTP Server."

You can install FreeBSD directly from an FTP server, but you should have an always-on, fast Internet connection available in order to do so. Performing an FTP install over a modem will take a *very* long time.

If you are installing FreeBSD from the FreeBSD sites or one of the official mirrors, you can log in as anonymous. In this case, you can skip the next section. If, however, you are installing from an FTP server that does not allow anonymous logins (such as an internal FTP server on a LAN), you will need to follow the procedures in the next section to configure the username first.

Configuring the Username

Sysinstall uses a preconfigured username and password for making FTP connections. From the Sysinstall main menu, shown in Figure 3.1, arrow down to Options and press Enter. In the Options menu, shown in Figure 3.2, you can set the FTP login name.



FIGURE 3.1 The FreeBSD Sysinstall main menu. From here, select the Options menu item.

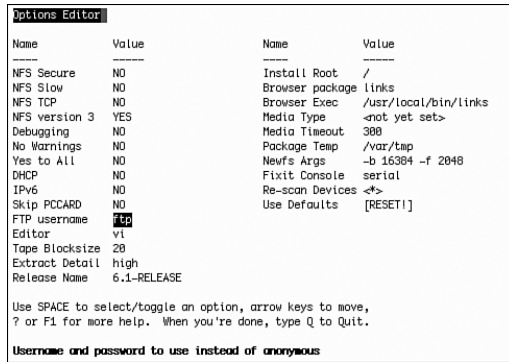


FIGURE 3.2 The Options menu, in which you can set the FTP login name (among other installation preferences).

Arrow down to the option that reads FTP Username and press the spacebar. A dialog asks you to enter the username for FTP access; the administrator of the FTP server you will be using should furnish you with a username and password that will let you into the server. Enter the username and press Enter. You are asked to supply a password. Enter the password you need to use and press Enter again.

After you have finished, press Q to quit, and you will be returned to the Main Sysinstall menu.

Selecting an FTP Install

After you have set the FTP username and password (if necessary), follow the instructions in Chapter 2, found in the “Creating Partitions and Assigning Mount Points” and “Selecting a Canned Distribution Set” sections, until you get to the screen where you are asked to choose the installation media. From this screen, select FTP or FTP Passive if the server you intend to install from is behind a firewall (ask your system administrator if you are not sure). You will then be asked to select a distribution site (see Figure 3.3).

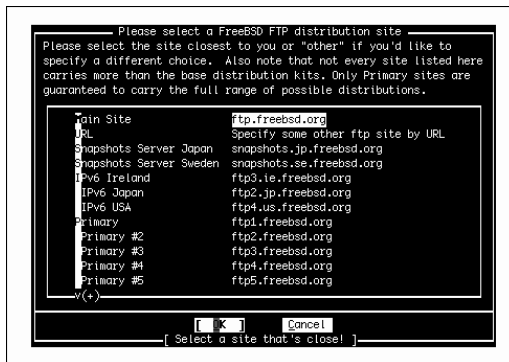


FIGURE 3.3 The list of available FTP servers from which you can install FreeBSD. Choose the public server closest to you or choose a custom URL.

If you are installing from one of the FreeBSD mirror sites, you can select the site from the list, using whichever server is listed as being geographically closest to you (to maximize download speed). Otherwise, select URL to specify an FTP server manually. In the dialog that opens (see Figure 3.4), you will be asked to specify the name of the FTP server as well as the path to where the FreeBSD installation files are located. Figure 3.4 shows an example for an FTP site with the hostname `lion` located on the network `example.com`, with the FreeBSD files located in the `/FreeBSD` directory.



FIGURE 3.4 Specifying a custom FTP server location by URL.

After you have configured the FTP server you want to install from, you will need to configure the network (see Figure 3.5). Follow the procedure in the “Configuring the Network” section of Chapter 2 to complete this configuration. This step only needs to be completed once; after your network has been configured, you can return to Sysinstall and perform other post-installation tasks without having to configure the network again. The Sysinstall program gives you the option to skip that step or re-enter other network data.

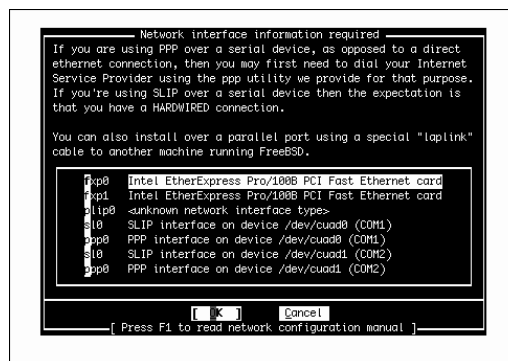


FIGURE 3.5 Configuring the TCP/IP networking parameters. This information is stored as the system’s TCP/IP configuration.

After you have finished configuring the network, the various distribution sets will be downloaded to your computer and automatically installed. This process will probably take a long time; you may want to turn your attention to something else for the next few hours. When files have finished copying, you can go ahead with the instructions offered in the “Post-Installation Configuration and Customization” section of Chapter 2. Because you have already configured the network to do the installation, you can skip the network-configuration portion of those instructions.

Note that this chapter has covered only how to install FreeBSD from an FTP server. If you want to set up an “installation server” that can be used by clients on your local network to install FreeBSD, see Chapter 27. After you have an FTP server set up and the FreeBSD installation files available on it, you can have other systems install from it using the procedures in this chapter.

Performing an NFS Install

NFS stands for *Network File System*. It provides a way for filesystems located on a server elsewhere on the network to be mounted into your own computer’s directory structure, and the files in it to be made accessible as though they were on your own hard disk. File sharing with NFS is fully covered in Chapter 33, “The Network File System (NFS).”

FreeBSD can be installed over NFS, assuming that there is an NFS server on your network that has the installation files available.

If your NFS server will work only on a secure port (or if you have an old computer with slow network hardware), follow the procedure in the following section. Otherwise, you can skip immediately ahead to “Selecting an NFS Install.”

Configuring Sysinstall for a Secure Port or Slow Connection

At the Sysinstall main menu (shown previously in Figure 3.1), select Options and press Enter. The first listing in the Options menu is NFS Secure. If your NFS server only works on a secure port (ask your system administrator if you’re not sure), press the spacebar to toggle this to Yes. The second option (NFS Slow) should be toggled to Yes if you have a slow PC (something from the 386 era) or an Ethernet card with very poor performance. This is almost always unnecessary; the only cases where an Ethernet card would be slow enough to necessitate this option is if your own computer’s filesystem is being remotely accessed, as from yet another NFS server, instead of being locally mounted. After you have made these changes, press Q to return to the Sysinstall main menu.

Installation then continues, as discussed in Chapter 2, up to the point where you are asked to choose the installation media.

Selecting an NFS Install

At the Choose Installation Media screen, select NFS. You will then be asked to enter the name of the NFS server followed by the path where the FreeBSD installation files are located. In the example in Figure 3.6, the server is `lion` and the installation directory is `install/FreeBSD`.

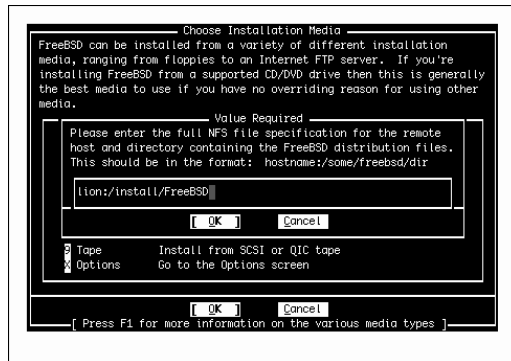


FIGURE 3.6 Specifying the NFS location of the remote FreeBSD installation files.

After you have entered this information, you will need to configure the network (shown previously in Figure 3.5). Follow the procedure shown in Chapter 2 under the “Configuring the Network” heading for instructions on how to do this.

After you have finished configuring the network, the various distribution sets will be downloaded from the remote server and installed. This will naturally take longer than installing from a CD-ROM, but it will likely be faster than an FTP installation. When the files have finished copying, you can go ahead with the “Post-Installation Configuration and Customization” section of Chapter 2. Because you have already configured the network to do the installation, you can skip the network-configuration portion.

Note that this chapter has covered only how to install FreeBSD from an NFS server. If you want to set up an installation server that can be used by clients to install FreeBSD over NFS, see Chapter 33. After you have an NFS server set up and the FreeBSD installation files available on it, you can have other systems install from it using the procedures discussed in this chapter.