Installing and Operating Academic Operating System 4.3

ABSTRACT

This article is a revision of an article entitled "Installing and Operating 4.3BSD on the VAX," written in April 1986 by Michael J. Karels *et al.*, and found in the *UNIX System Manager's Manual.* The revisions include additions and changes appropriate to the IBM RT PC and IBM 6152 Academic System.

The article contains the following chapters:

1. **Introduction** provides information helpful in installing and operating IBM/4.3. It covers hardware supported, distribution format, IBM/4.3 device naming and IBM/4.3 block and raw devices.
2. **Saving and Restoring Local Modifications** describes how to upgrade a currently running release to an entirely new release.
3. **Installation Procedures** describes how to install the IBM/4.3 system.
4. **System Setup** describes the procedures used to set up an IBM/4.3 system.
5. **Network Setup** describes how to configure your system to use the networking support.
6. **System Operation** describes some typical IBM/4.3 operations on an IBM RT PC and IBM 6152 Academic System.

Appendix A. AIX and IBM/4.3 Co-residence describes installing and using AIX on a IBM/4.3 system.

Appendix B. MINIROOT Kernel Configured Devices describes the configurable devices supported by the MINIROOT kernel.

Appendix C. Building a Master/Server Machine describes how very small installations with a single IBM RT PC can combine functions usually shared by several RTs.

Appendix D. Building Separate Master and Server Machines describes how larger installations with two or more IBM RT PCs can get a network of 6152s and RT PCs installed and working with the distributed file systems (AFS and NFS).

Note that throughout this article, prompts that appear on the screen are shown in *italics*, while values you are to type appear in **boldface**.

1. INTRODUCTION

This article explains how to install IBM/4.3. Most sites will have several IBM RT PCs and many IBM 6152s. However, in a very small installation, you might have a single IBM RT that functions as master for network installations, server for both the Andrew File System and NFS, and the uucp connection. See Appendix C of this article for guidance in creating such a machine. Appendix D should be used as a guide for setting up master machines and servers for larger installations.

The normal installation process differs slightly between the IBM RT PC and the IBM 6152 Academic System. For both, you begin by booting a system from diskette. For the IBM 6152 only, you copy a small root file system image onto a swap area, and boot this. For both, you then load the IBM/4.3 root, /usr and /usr/src file systems. Last, you reboot and then load the remainder of the distributed binaries and sources.

The technique for upgrading from prior releases is described in Chapter 2 of this article. Some applications that were compiled with release one or two of IBM 4.3 (or 4.2/RT) will not work under release three of IBM 4.3 due to changes introduced by the NFS port. In some cases relinking your programs with the new libraries is sufficient to get them working again. It is desirable however to recompile most local software after the conversion, as there are many changes and performance improvements in the compilers, standard libraries, and floating point linkage.

1.1. Software Supported

IBM/4.3 support all of 4.3BSD, excepting those items listed in the "Software Description" found in Appendix A of *IBM Academic Operating System 4.3.*

1.2. Hardware Supported

IBM/4.3 runs on the IBM 6152 Academic System and six models of the IBM RT PC: the IBM 6151 Model 010, 015 and 115, 130 (desk models) and the IBM 6150 Model 020, 025, and 125, 135 (floor models). This section describes the specific configurations supported.

Note: All configurations MUST have 2 MB of contiguous memory starting at address zero.

1.2.1. IBM 6152 Academic System

IBM Academic Operating System 4.3 supports the following configurations of this new system:

- A 6152 Academic System Unit and Enhanced Personal Computer Keyboard with one of the memory and fixed disk configurations listed below:
  - 1 IBM 20MB Fixed-Disk Drive (Model 022)

    2MB of memory (Model 022) or
    4MB of memory (Model 024)
  - 1 IBM 44MB Fixed-Disk Drive and

    4MB of memory (Model 044) or
    8MB of memory (Model 048)
  - 1 IBM 70MB Fixed-Disk Drive and

    4MB of memory (Model 074) or
    8MB of memory (Model 078)

1.2.1.1. Additional Standard Features of the IBM 6152

- An MC68881 Floating Point unit
- A Serial Asynchronous port
- A Parallel port
- A Pointing Device port
- A Video Graphics Array (VGA) and display port
- A Diskette Controller
- A 1.44MB (3.5 inch) Diskette Drive

1.2.1.2. Optional Features of the IBM 6152

- A second IBM 44MB Fixed-Disk Drive (Models 044 or 048)
- A second IBM 70MB Fixed-Disk Drive (Models 074 or 078)
- A second IBM 115 MB Fixed-Disk Drive (Models 074 or 078)
- The IBM 8503 PS/2 Monochrome Display
- The IBM 8604 PS/2 Monochrome Display
- An IBM 8770 PS/2 Mouse
- An IBM PS/2 Memory expansion for advanced color (#4081)
- The IBM 8513 PS/2 Color Display
- An IBM PS/2 Color Graphics Display Adapter 8514/A (#4054)
- The IBM 8514 PS/2 Color Display
- Up to two local area network adapters (one required; see "Local Area Networks" below)
- The IBM 4216 Personal Page Printer

1.2.2. IBM 6151 Model 010 Processor

IBM/4.3 for the Model 010 supports the following hardware:

- A 6151 Model 010 System Unit and Keyboard
- 2 MB of memory (but not as two separate 1 MB memory cards)
- An IBM PC AT Fixed-Disk and Diskette Drive Adapter (standard)
- An IBM RT PC 40 MB Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.3. IBM 6151 Model 015 Processor

IBM/4.3 for the Model 015 supports the following hardware:

- A 6151 Model 015 System Unit and Keyboard
- 2 MB of memory
- An IBM RT PC Enhanced Small Device Interface (ESDI) Magnetic Media Adapter (standard)
- An IBM RT PC 70 MB ESDI Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.4. IBM 6151 Model 115 Processor

IBM/4.3 for the Model 115 supports the following hardware:

- A 6151 Model 115 System Unit and Keyboard with 4 MB of memory
- Advanced Processor Card with built-in MC68881 floating-point unit
- An IBM RT PC Extended ESDI Magnetic Media Adapter
- An IBM RT PC 70 MB Extended ESDI Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.5. IBM 6151 Model 130 Processor

IBM/4.3 for the Model 130 supports the following hardware:

- A 6151 Model 130 System Unit and Keyboard with 16 MB of memory
- An enhanced CMOS RISC processor card with an enhanced level of the Advanced Floating Point Accelerator
- An IBM RT PC 114 MB Extended ESDI Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.6. IBM 6150 Model 020 Processor

IBM/4.3 for the Model 020 supports the following hardware:

- A 6150 Model 020 System Unit and Keyboard
- 2 MB of memory (but not as two separate 1 MB memory cards)
- An IBM PC AT Fixed-Disk and Diskette Drive Adapter (standard)
- 2 Asynchronous (RS-232-C) Serial Ports
- An IBM RT PC 40 MB Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.7. IBM 6150 Model 025 Processor

IBM/4.3 for the Model 025 supports the following hardware:

- A 6150 Model 025 System Unit and Keyboard
- 2 MB of memory
- An IBM RT PC ESDI Magnetic Media Adapter (standard)
- 2 Asynchronous RS232C Serial Ports
- An IBM RT PC 70 MB ESDI Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.8. IBM 6150 Model 125 Processor

IBM/4.3 for the Model 125 supports the following hardware:

- A 6150 Model 125 System Unit and Keyboard with 4 MB of memory
- Advanced Processor Card with built-in MC68881 floating-point unit
- An IBM RT PC Extended ESDI Magnetic Media Adapter
- 2 asynchronous RS232C Serial Ports
- An IBM RT PC 70 MB Extended ESDI Fixed-Disk Drive (standard)

- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.9. IBM 6151 Model 135 Processor

IBM/4.3 for the Model 135 supports the following hardware:

- A 6151 Model 135 System Unit and Keyboard with 16 MB of memory
- An enhanced CMOS RISC processor card with an enhanced level of the Advanced Floating Point Accelerator
- An IBM RT PC 114 MB Extended ESDI Fixed-Disk Drive (standard)
- An IBM PC AT High Capacity (1.2 MB) Diskette Drive (standard)

1.2.10. Peripherals and Optional Features

The following peripherals and options are supported on all 0xx-series models:

- An IBM RT PC 1 MB Memory Expansion (#8222) or an IBM RT PC 2 MB Memory Expansion (#4739) or an IBM RT PC 4 MB Memory Expansion (#3156) (up to 8 MB total)

The following peripherals and options are supported on all 1xx-series models:

- An IBM RT PC 4 MB Fast Memory Expansion (#7004) or an IBM RT PC 8 MB Fast Memory Expansion (#7008) (up to 16 MB total)
- The selectable Fixed Disk Drive Option (Feature 4300) substitutes a 310 MB Extended ESDI Fixed Disk Drive for the standard 70 MB drive supplied with the system. This feature is available with the initial installation of the system.

The following peripherals and options are supported on all models of the 6150, except where noted:

- Up to 2 additional 40 MB Fixed-Disk Drives, 70 MB ESDI Fixed-Disk Drives, or 70 MB, 114 MB, or 310 MB Extended ESDI Fixed-Disk Drives (Model 125, 130, and 135 only).

  Note: When mixing drives of different types, separate adapters are needed. No more than two adapters can be installed on a given machine, hence no more than two types of drive can be installed on a given machine. Except in the case where all three drives are of the extended ESDI type, two adapters are needed to install a third drive. The Extended ESDI Fixed-Disk Drive is only supported on the 1xx-series models.

- A second diskette drive, which may be either the IBM PC AT High Capacity (1.2 MB) Diskette Drive or the IBM PC AT Dual-Sided (360 KB) Diskette Drive (#0206)

The following optional features are supported on all models of the IBM RT PC:

- An IBM RT System Buffered Four-Port Asynchronous RS232 adapter (#1356)
- An IBM RT PC Four-Port Asynchronous RS232C Adapter (#4763)
- An IBM PC AT Serial/Parallel Adapter (#0215)
- An IBM 6157 Streaming Tape Drive with adapter (#4797)
- Either of two Local Area Network Adapters (see "Local Area Networks" below)
- IBM All-Points-Addressable Displays (see "IBM All-Points-Addressable Displays" below)

- An IBM 5151 Monochrome Display attached to an IBM Monochrome Display and Printer Adapter (#4900)
- An IBM RT PC Floating Point Accelerator (#4758)
- An IBM RT PC Mouse (#8426)
- An IBM 3812 Pageprinter
- An IBM 4201 Proprinter or an IBM 5152 Graphics Printer (withdrawn from marketing)
- An IBM RT PC Small Computer System Interface Adapter (#7000), used to attach IBM 9332 DASD Models 240, 250, 440, and 450

1.2.10.1. Local Area Networks

You can install up to two LAN adapters in a single IBM 6150, IBM 6151, or IBM 6152 system unit. The two-adapter configuration can be either two IBM Token-Ring adapters, two Ethernet adapters, or one of each. At least one adapter must be installed in the IBM 6152.

1.2.10.1.1. IBM Token-Ring Network:

An IBM Token-Ring Network Adapter provides connection of the IBM 6150, 6151, or 6152 to the IBM Token-Ring Network. The network uses the IBM Cabling System for physical connection and a Token-Ring access protocol for network traffic control.

1.2.10.1.2. Ethernet:

The IBM RT PC Baseband Adapter (#6810) provides connection of the IBM 6150 or 6151 to an Ethernet network. The Ungermann-Bass Ethernet Adapter NICps/2 provides connection of the IBM 6152 to an Ethernet network. This adapter is not available from IBM; you must supply it.

Two Ethernet cards cannot be used in conjunction with an 8514/A in the 6152.

1.2.10.2. IBM All-Points-Addressable Displays

IBM/4.3 supports the following All-Points-Addressable displays and their associated adapters.

On the IBM RT PC:

(1) An IBM 6153 Advanced Monochrome Graphics Display

(2) An IBM 6154 Advanced Color Graphics Display

(3) An IBM 6155 Extended Monochrome Graphics Display

(4) The IBM 5154 Enhanced Color Display with the IBM Enhanced Graphics Adapter (EGA)

(5) The IBM 5081 Display with MegaPel Adapter

(6) An IBM Academic Information Systems experimental display, which is no longer available.

On the IBM 6152 Academic System:

(1) The IBM 8503 PS/2 Color Display

(2) The IBM 8513 PS/2 Monochrome Display

(3) The IBM 8604 PS/2 Monochrome Display

(4) The IBM 8514 PS/2 Color Display

1.2.11. Minimum Configuration Requirements

At least one IBM RT PC (the master) must have at least two 70 MB Fixed-Disk Drives (of any type) and an IBM 6157 Streaming Tape Drive with adapter for tape installation. All other machines may use either tape installation or have a network connection to the master machine for network installation.

1.3. Distribution Format

The distribution includes a cover letter and program directory, installation diskettes, distribution streaming tapes, and copies of the book, *IBM Academic Operating System 4.3*.

The distribution includes all 4.3BSD source files, whether ported or not. (User-contributed software is not provided. The source distribution is available from the University of California at Berkeley.) However, the distribution includes binary files for only those programs which have been ported and tested.

1.4. IBM/4.3 Device Naming

IBM/4.3 has a set of names for devices which are different from the RT hardware names, such as:

Fixed (Hard) Disks	hd
Diskettes (Floppies)	fd
6157 Streaming Tape	st

The normal standalone system, used to bootstrap the full IBM/4.3 system, uses device names of the form:

xx(y,z)

where *xx* is either **hd** or **fd**. The value *y* specifies the adapter to use and also the device. It is computed as

*N * adapter + device*

where *N* is either 2 or 3 (it is 3 only for fixed disks when adapter 0 is of the Extended ESDI type); *adapter* is the adapter to which the device is connected; and *device* is the device number on the adapter.

For example, the first disk (disk 0) on the first adapter (adapter 0) would have a *y* value of 0. The first disk on the second adapter would have a *y* value of either 2 or 3. The value *z* (in the range 0-7) is interpreted as a disk *partition* (in the range 0-7) for the hard disk, and is ignored for the floppy disk.

A IBM/4.3 physical (hard) disk is divided into eight logical disk partitions, each of which may occupy any consecutive cylinder range on the physical device. The cylinders occupied by the eight default partitions for each drive type are specified in the disk description file /etc/disktab (see *disktab*(5)). Non-standard partition sizes may be specified on a per disk basis (see *minidisk*(8R)). Each partition may be used either to store a IBM/4.3 file system or as a raw data area (such as a paging area). Typically the following is used:

- The first partition (partition 0 on drive 0) stores a root file system from which IBM/4.3 can be bootstrapped. The name of this partition is hd(0,0) for standalone programs, and /dev/hd0a for programs run under the kernel.
- The second partition -- hd(0,1) or /dev/hd0b -- is a paging area.
- The third partition -- hd(0,2) or /dev/hd0c -- is the entire 4.3 disk. It should never be used as a file system. It is useful when arbitrary sectors on the disk need to be

accessed. When you format a disk for example, the bad block table in the first cylinder of the disk needs to be written to.

Loss of the information contained in this bad block table can compromise the integrity of the other data on the disk. It is therefore a good idea to copy the first 4 tracks of each disk in case of accident. The copy should be placed on a separate disk or diskette so it can be recovered if needed. The standalone COPY command can be used to copy and restore this information.

- The seventh partition -- hd(0,6) or /dev/hd0g -- is used for the /usr file system.
- On the 6152 only, the eighth partition -- hd(0,7) or /dev/rhd0h -- is used to access the entire disk including any DOS partitions. On the 6152, the third partition -- hd(0,2) or /dev/hd0c is a subset of this partition.
- The remaining partitions, hd(0,3), hd(0,4), hd(0,5) and hd(0,7) (RT PC only); or /dev/hd0d, /dev/hd0e, /dev/hd0f, and /dev/hd0h (RT PC only) are not normally used. They can be defined however. See minidisk(8R) for instructions on how to do this.

1.5. IBM/4.3 Devices: Block and Raw

IBM/4.3 makes a distinction between "block" and "raw" (character) devices. Each disk has a block device interface that makes the device byte-addressable; you can write a single byte anywhere on the disk. The system reads the data from the disk sector, inserts the byte to be written, and writes the modified data. Names like /dev/hd0a indicate block devices. There are also raw devices available. These have names like /dev/rhd0a, the "r" indicating "raw." Raw devices bypass the buffer cache and perform I/O directly to/from the program's I/O buffers; they are normally restricted to full-sector transfers. The bootstrap procedure often uses the raw device interfaces because these tend to work faster in some cases. Raw devices are used when making new file systems, when checking unmounted file systems, or for copying quiescent file systems. The block devices are used to mount file systems or to operate on a mounted file system such as the root.

Be aware that it is often important which interface is used: the character device interface (for efficiency), or the block device interface (to write specific bytes within a sector). Do not indiscriminately change the installation instructions to use the alternate type of device interface.

2. SAVING AND RESTORING LOCAL MODIFICATIONS

This chapter is intended for those sites which:

(1) Are currently running a release of 4.2/RT or IBM/4.3,

AND

(2) Are about to install an entirely new release of IBM/4.3 to replace the existing system.

If you do not have a previous release of 4.2/RT or IBM/4.3 installed, skip this chapter and go to Chapter 3, "Installation Procedures."

If you are installing an entirely new release of IBM/4.3 to replace an existing 4.2/RT or IBM/4.3 system, you will probably want to back up all files as described in the section "Backing Up Your System." Then, after you have installed the new version of IBM/4.3, you can selectively restore locally modified files.

2.1. Backing Up Your System

It is highly recommended that you do a complete system backup so that you can selectively restore the files you want after installing the new release. On a standard system there are three file systems to back up: hd0a, hd0g and hd1g. If your site has a standard procedure for backing up these file systems, use that procedure instead of the following. (For example, you might use *rdump*(8) over the network to another RT with a streaming tape drive attached.)

2.1.1. Using Streaming Tapes for Backup

If your site does not have its own backup procedures, you can use the following instructions to back up your file systems to streaming tape. This method assumes you have a streaming tape drive on your RT. You need one blank streaming tape for each of the three file systems.

To back up your system, do the following:

(1) Erase each of the three streaming tapes. To erase a tape, insert it in the streaming tape drive and type:

mt -f /dev/rst0 erase <Enter>

(2) Dump each file system onto tape. Begin by typing the superuser command (**su**) and supplying the superuser password.

(3) Insert an erased tape into the streaming tape drive and type:

dump 0sf 4400 /dev/st0 /dev/rhd0a <Enter>

This tape will be referred to later as the "rhd0a" tape.

(4) Remove the first tape from the streaming tape drive, insert the second tape and type:

dump 0sf 4400 /dev/st0 /dev/rhd0g <Enter>

This tape will be referred to later as the "rhd0g" tape.

(5) Remove the second tape from the streaming tape drive, insert the third tape and type:

dump 0f /dev/st0 /dev/rhd1g <Enter>

This tape will be referred to later as the "rhd1g" tape.

2.1.2. Using Diskettes for Backup

If your site does not have its own backup procedures, you can use the following instructions to back up your file systems to diskette.

To back up your system, do the following:

(1) Determine the size of your file systems. To do so, type the following command:

df

The size in Kbytes is listed under the column headed "used." Multiply that number by 1024 before using it in the formula in the next step.

(2) Determine the approximate number of diskettes needed to back up each file system, using the following formula:

number of diskettes = (file system size in bytes) / (capacity of diskette)

On an IBM RT PC, a 1.2M high density diskette holds 1228800 bytes. On a 6152, a 1.44M high density diskette holds 1474560 bytes.

Round fractions up, then total the number of diskettes needed for all file systems. Be sure to have enough usable diskettes before beginning.

(1) To format a high density diskette, insert the diskette in the upper diskette drive and type the following command:

fdformat -h /dev/rfd0
Format /dev/rfd0 to high density diskette? **yes**

Repeat this step for each diskette that needs formatting.

(2) Dump each file system onto diskette. Begin by typing the superuser command (**su**) and supplying the superuser password.

(3) Insert a formatted diskette into the upper diskette drive drive and type:

On an RT:

dump 0sf 117 /dev/rfd0 /dev/rhd0a <Enter>

On a 6152:

dump 0sf 141 /dev/rfd0 /dev/rhd0a <Enter>

As you remove each diskette during this step, be sure to label it appropriately, indicating both the file system (rhd0a) and which diskette (1st, 2nd, 3rd, etc.) it is in the sequence.

(1) Insert a formatted diskette into the upper diskette drive and type:

On an RT:

dump 0sf 117 /dev/rfd0 /dev/rhd0g <Enter>

On a 6152:

dump 0sf 141 /dev/rfd0 /dev/rhd0g <Enter>

As you remove each diskette during this step, be sure to label it appropriately, indicating both the file system (rhd0g) and which diskette (1st, 2nd, 3rd, etc.) it is in the sequence.

(1) Insert a formatted diskette into the upper diskette drive and type:

On an RT:

dump 0sf 117 /dev/rfd0 /dev/rhd1g <Enter>

On a 6152:

dump 0sf 141 /dev/rfd0 /dev/rhd1g < Enter >

As you remove each diskette during this step, be sure to label it appropriately, indicating both the file system (rhd1g) and which diskette (1st, 2nd, 3rd, etc.) it is in the sequence.

NOTES:

(1) Both *dump*(8) and *restore*(8) use the raw diskette device.

(2) Be sure to label the diskettes appropriately during the backup to facilitate a restore from diskettes.

2.2. Converting from 4.2/RT to IBM/4.3

Begin by reading the article entitled "Bug Fixes and Changes in 4.3 BSD" in the *UNIX System Manager's Manual* and the "Summary of Amendments" (if any) in *IBM Academic Operating System 4.3* to see what has changed since the last release of 4.2/RT. If you have local system modifications to the kernel to install, look at the article entitled "Changes to the Kernel in 4.3 BSD" (in the *UNIX System Manager's Manual*) to learn how the system changes will affect your local modifications.

If you are running 4.2/RT, upgrading your system involves replacing your kernel and system utilities. Some binaries compiled under 4.2/RT will work without recompilation under IBM/4.3, though they may run faster if they are relinked. There are several classes of programs that will have to be recompiled before they will work correctly. See the following section (Upgrading to an NFS system) for more information. The easiest way to convert to IBM/4.3 (depending on your file system configuration) is to install new root and /usr file systems from the distribution ROOT/USER tape on an unused disk's a and g partitions, boot the new system, and then copy any local utilities from your old root and /usr file systems into the new ones. All user file systems and binaries can be retained unmodified, except that the new *fsck* should be run before they are mounted (see Section 2.3.1).

Section 2.3 lists the files to be saved as part of the conversion process. Section 2.4 describes the bootstrap process. Section 2.5 discusses the merger of the saved files back into the new system. Section 2.6 provides general hints on possible problems to be aware of when converting from 4.2/RT to IBM/4.3.

2.3. Files to Save

The easiest upgrade path from 4.2/RT is to install new root and /usr file systems on an unused disk's a and g partitions, then copy or merge site specific files into their corresponding files on the new system. The list on the next page enumerates the standard set of files you will want to save and suggests directories in which site-specific files should be present. This list will likely be augmented with non-standard files you have added to your system. If you do not have enough space to create parallel file systems, you should create a *tar* image of the following files before installing IBM/4.3.

/.cshrc	†	root csh startup script
/.login	†	root csh login script
/.profile	†	root sh startup script
/.rhosts	†	for trusted machines and users
/dev/MAKEDEV	‡	in case you added anything here
/dev/MAKEDEV.local	*	for making local devices
/etc/disktab	‡	in case you changed disk partition or block sizes
/etc/fstab	#	disk configuration data
/etc/ftpusers	†	for local additions
/etc/gateways	†	routing daemon data base
/etc/gettytab	‡	getty data base
/etc/group	*	group data base
/etc/hosts	†	for local host information
/etc/hosts.equiv	†	for local host equivalence information
/etc/networks	†	for local network information
/etc/passwd	*	user data base
/etc/printcap	†	line printer data base
/etc/protocols	‡	in case you added any local protocols
/etc/rc	*	for any local additions
/etc/rc.local	*	site specific system startup commands
/etc/remote	†	auto-dialer configuration
/etc/services	‡	for local additions
/etc/syslog.conf	*	system logger configuration
/etc/securettys	*	for restricted list of ttys where root can log in
/etc/ttys	*	terminal line configuration data
/etc/ttytype	*	terminal line to terminal type mapping data
/etc/termcap	‡	for any local entries that may have been added
/lib	‡	for any locally developed language processors
/usr/dict/*	‡	for local additions to words and papers
/usr/hosts/MAKEHOSTS	†	for local changes
/usr/include/*	‡	for local additions
/usr/lib/aliases	†	mail forwarding data base
/usr/lib/crontab	*	cron daemon data base
/usr/lib/font/*	‡	for locally developed font libraries
/usr/lib/lib*.a	†	for locally developed libraries
/usr/lib/lint/*	‡	for locally developed lint libraries
/usr/lib/sendmail.cf	*	sendmail configuration
/usr/lib/tabset/*	‡	for locally developed tab setting files
/usr/lib/term/*	‡	for locally developed nroff drive tables
/usr/lib/tmac/*	‡	for locally developed troff/nroff macros
/usr/lib/uucp/*	†	for local uucp configuration files
/usr/man/manl	†	for manual pages for locally developed programs
/usr/msgs	†	for current msgs
/usr/spool/*	†	for current mail, news, uucp files, etc.
/usr/src/local	†	for source for locally developed programs
/sys/conf/HOST	†	configuration file for your machine
/sys/conf/files.HOST	†	list of special files in your kernel
/*/quotas	†	file system quota files

†Files that can be used from 4.2/RT without change. ‡Files that need local modifications merged into IBM/4.3 files. #The format of /etc/fstab has been changed to accommodate NFS. See section 2.8 (Upgrading to an NFS system)

for a description of the changes. * Files that require special work to merge and are discussed below.

2.4. Building a IBM/4.3 System

There are two approaches to upgrading from 4.2/RT to IBM/4.3. In the first approach, one loads the IBM/4.3 distribution into unused partitions of an existing disk, boots IBM/4.3 from this disk, optionally builds a tailored kernel for the configuration and boots it, mounts the old (existing) 4.2/RT partitions onto the IBM/4.3 system, merges the old files into the new system, and frees up the old partitions. In the second approach, one backs up the existing system, loads the IBM/4.3 distribution as if bootstrapping a brand-new system, and restores and merges the old files into the new system.

The method chosen depends upon a number of factors, including the amount of available disk space and the expertise of the installer.

Users without a third disk or without much expertise might find it easier to use the second approach, while experienced users may find the first approach faster.

The first approach is described below. If you elect to follow the second approach, you should still read the rest of this chapter. You will return to these instructions after you install IBM/4.3 using the instructions in the following chapter entitled "Installation Procedures."

To build a working IBM/4.3 system, following the steps in the next chapter, specifying an unused disk as the root device as described in the section "Changing Installation Options." The root and /usr file system dump on the tape could also be extracted directly, although this will require an additional file system check after booting IBM/4.3 to convert the new root file system. The exact procedure chosen will depend on the disk used for IBM/4.3, but the following procedure demonstrates extraction onto hd2. After becoming ROOT, issue the following commands:

```
umount -a
newfs /dev/hd2a[1] <Enter>
fsck /dev/rhd2a <Enter>
newfs /dev/hd2g <Enter>
fsck /dev/rhd2g <Enter>
mount /dev/hd2a /mnt <Enter>
mkdir /mnt/usr <Enter>
mount /dev/hd2g /mnt/usr <Enter>
cd /mnt <Enter>
restore xvf /dev/st0 <Enter>
```

If there is insufficient space to load the new root and /usr file systems before reusing the existing 4.2/RT partitions, it is strongly advised that you make full dumps of each file system on streaming tapes before beginning. It is also desirable to run file system checks of all file systems to be converted to IBM/4.3 before shutting down 4.2/RT. In either case, this is an excellent time to review your disk configuration for possible tuning of the layout. The "Disk Configuration" section in the "System Setup" chapter is required reading.

To ease the transition to new kernels, the IBM/4.3 bootstrap routines now pass the identity of the boot device through to the kernel. The kernel then uses that device as its root file system. Thus, for example, if you boot from /dev/hd1a, the kernel will use hd1a as its root file system. If /dev/hd1b is configured as a swap partition, it will be used as the initial

[1] You might need to change the free space percentage to four or five percent.

swap area. Otherwise, the normal primary swap area (/dev/hd0b) will be used. The IBM/4.3 bootstrap is backward compatible with 4.2/RT, so you can replace your 4.2/RT bootstrap if you use it to boot your first IBM/4.3 kernel.

Once you have extracted the IBM/4.3 system and booted from it, you will have to build a kernel customized for your configuration. If you have any local device drivers, they will have to be incorporated into the new kernel. See the section "Building New System Images" in the chapter "System Setup" of this article, and the article "Building IBM/4.3 Systems with Config."

The disk partitions are by default the same as those in the 4.2/RT. Release 3 of IBM/4.3 requires that you use non-standard partition sizes for the "root" disk (the disk with the root and /usr partitions on it). You may convert 4.2 partitions, with the exception of the root disk, to 4.3 using the IBM/4.3 version of *fsck* on each file system, allowing it to make the necessary corrections. The new version of *fsck* is more strict about the size of directories than the version supplied with 4.2/RT. Thus the first time that it is run on a 4.2/RT file system, it will produce messages of the form:

DIRECTORY ...: LENGTH xx NOT MULTIPLE OF 512 (ADJUSTED)

Length "xx" will be the size of the directory; it will be expanded to the next multiple of 512 bytes. Note that file systems are otherwise completely compatible between 4.2/RT and IBM/4.3, though running a IBM/4.3 file system under 4.2/RT may cause more of the above messages to be generated the next time it is *fsck*'ed on IBM/4.3.

In order to convert the Root disk to 4.3, you must use the minidisk program to alter the partition sizes. This is described in more detail in the "Installation Procedures" section.

2.5. Merging your files from 4.2/RT into IBM/4.3

When your system is booting reliably and you have the IBM/4.3 root and /usr file systems fully installed, you will be ready to continue with the next step in the conversion process, merging your old files into the new system.

If you saved the files on a *tar* tape, you can extract them into a scratch directory (say /usr/convert), as in the following example:

```
# mkdir /usr/convert <Enter>
# cd /usr/convert <Enter>
# tar xf /dev/rst0 <Enter>
```

If you used *dump*, see Section 3.7 below.

The data files marked in the previous table with a dagger (†) may be used without change from the previous system. Those data files marked with a double dagger (‡) have syntax changes or substantial enhancements. You should start with the IBM/4.3 version and carefully integrate any local changes into the new file. Usually these local modifications can be incorporated without conflict into the new file; some exceptions are noted below. The files marked with an asterisk (*) require particular attention and are discussed below.

If you have any home grown device drivers in /dev/MAKEDEV.local that use major device numbers reserved by the system you will have to modify the commands used to create the devices or alter the system device configuration tables in /sys/ca/conf.c. Otherwise /dev/MAKEDEV.local can be used without change from 4.2/RT.

System security changes require adding several new "well-known" groups to /etc/group. The groups that are needed by the system as distributed are:

name	number
wheel	0
daemon	1
kmem	2
sys	3
tty	4
operator	5
staff	10

Only users in the "wheel" group are permitted to *su* to "root." Most programs that manage directories in /usr/spool now run set-group-id to "daemon" so that users cannot directly access the files in the spool directories. The special files that access kernel memory, /dev/kmem and /dev/mem, are made readable only by group "kmem." Standard system programs that require this access are made set-group-id to that group. The group "sys" is intended to control access to system sources, and other sources belong to group "staff." Rather than make users' terminals writable by all users, they are now placed in group "tty" and made only group writable. Programs that should legitimately have access to write on user's terminals such as *talk* and *write* now run set-group-id to "tty." The "operator" group controls access to disks. By default, disks are readable by group "operator," so that programs such as *df* can access the file system information without being set-user-id to "root."

Several new users have also been added to the group of "well-known" users in /etc/passwd. The current list is:

name	number
root	0
daemon	1
operator	2
uucp	66
nobody	32767

The "daemon" user is used for daemon processes that do not need root privileges. The "operator" user-id is used as an account for dumpers so that they can log in without having the root password. By placing them in the "operator" group, they can get read access to the disks. The "uucp" login has existed long before IBM/4.3, and is noted here just to provide a common user-id. The password entry "nobody" has been added to specify the user with least privilege.

After installing your updated password file, you must run *mkpasswd*(8) to create the *ndbm* password data base. Note that *mkpasswd* is run whenever *vipw*(8) is run.

The format of the cron table, /usr/lib/crontab, has been changed to specify the user-id that should be used to run a process. The userid "nobody" is frequently useful for non-privileged programs.

Some of the commands previously in /etc/rc.local have been moved to /etc/rc; several new functions are now handled by /etc/rc.local. You should look closely at the prototype version of /etc/rc.local and read the manual pages for the commands contained in it before trying to merge your local copy. Note in particular that *ifconfig* has had many changes, and that host names are now fully specified as domain-style names (e.g, monet.Berkeley.EDU) for the benefit of the name server.

The C library and system binaries on the distribution tape are compiled with new versions of *gethostbyname* and *gethostbyaddr* which use the name server, *named*(8). If you have only a small network and are not connected to a large network, you can use the distributed library routines without any problems; they use a linear scan of the host table /etc/hosts if the name server is not running. If you are on the DARPA Internet or have a large local network, it is recommended that you set up and use the name server. For instructions on how to set up the necessary configuration files, refer to "Name Server Operations Guide for BIND." Several programs rely on the host name returned by *gethostname* to determine the local domain name.

If you want to compile your system to use the host table lookup routines instead of the name server, you will need to modify /usr/src/lib/libc/Makefile according to the instructions there and then recompile all of the system and local programs (see the section "Recompiling and Reinstalling System Software" in the chapter "System Operation" of this article). Next, you must run *mkhosts*(8) to create the *ndbm* host table data base from /etc/hosts.

The format of /etc/ttys has changed, see *ttys*(5) for details. It now includes the terminal type and security options that were previously placed in /etc/ttytype and /etc/securettys.

There is a new version of *syslog* that uses a more generalized facility/priority scheme. This has changed the format of the syslog.conf file. See *syslogd*(8) for details. *Syslog* now logs kernel errors, allowing events such as soft disk errors, file-system-full messages, and other such error messages to be logged without slowing down the system while the messages print on the console. It is also used by many of the system daemons to monitor system problems more closely, for example network routing changes.

If you are using the name server, your *sendmail* configuration file will need some minor updates to accommodate it. See the "Sendmail Installation and Operation Guide" and the sample *sendmail* configuration files in /usr/src/usr.lib/sendmail/nscf. Be sure to regenerate your *sendmail* frozen configuration file after installation of your updated configuration file.

The *init*(8) utility will prompt for the root password before invoking a super-user shell on the console (if the file /etc/nosingle exists).

The spooling directories saved on tape may be restored in their eventual resting places without too much concern. Be sure to use the 'p' option to *tar* so that files are recreated with the same file modes:

```
# cd /usr <Enter>
# tar xfp /dev/rst0 msgs spool/mail spool/uucp spool/uucppublic spool/news <Enter>
```

The ownership and modes of two of these directories need to be changed from their 4.2/RT values. *at* now runs set-user-id "daemon" instead of root. Also, the uucp directory no longer needs to be publicly writable, as *tip* reverts to privileged status to remove its lock files. After copying your version of /usr/spool, you should do the following:

```
# chown -R daemon /usr/spool/at <Enter>
# chown -R root /usr/spool/uucp <Enter>
# chgrp -R daemon /usr/spool/uucp <Enter>
# chmod -R o-w /usr/spool/uucp <Enter>
```

Whatever else is left is likely to be site specific or require careful scrutiny before placing in its eventual resting place. Refer to the documentation and source code before arbitrarily overwriting a file.

2.6. Hints on converting from 4.2/RT to IBM/4.3

This section summarizes the most significant changes between 4.2/RT and IBM/4.3, particularly those that are likely to cause difficulty in doing the conversion. It does not include

changes in the network; see Chapter 5 for information on setting up the network.

The mailbox locking protocol has changed; it now uses the advisory locking facility to avoid concurrent update of users' mail boxes. If you have your own mail interface, be sure to update its locking protocol.

The kernel's limit on the number of open files has been increased from 20 to 64. It is now possible to change this limit almost arbitrarily (there used to be a hard limit of 30). The standard I/O library autoconfigures to the kernel limit. Note that file ("_iob") entries may be allocated by *malloc* from *Ufopen*; this allocation has been known to cause problems with programs that use their own memory allocators. This does not occur until after 20 files have been opened by the standard I/O library.

Select can be used with more than 32 descriptors by using arrays of **int**s for the bit fields rather than single **int**s. Programs that used *getdtablesize* as their first argument to *select* will no longer work correctly. Usually the program can be modified to correctly specify the number of bits in an **int**. Alternatively the program can be modified to use an array of **int**s. There are a set of macros available in < sys/types.h > to simplify this. See *select*(2).

Old core files will not be intelligible by the current debuggers because of numerous changes to the user structure and because the kernel stack has been enlarged. The *a.out* header that was in the user structure is no longer present. Locally-written debuggers that try to check the magic number will need modification.

Find now has a data base of file names, constructed once a week from *cron*. To find a file by name only, the command *find name* will look in the data base for files that match the name. This is much faster than "find / –name name –print."

Files may not be deleted from directories having the "sticky" (ISVTX) bit set in their modes except by the owner of the file or of the directory, or by the superuser. This is primarily to protect users' files in publicly-writable directories such as /tmp and /usr/tmp. All publicly-writable directories should have their "sticky" bits set with "chmod +t."

The include file < time.h > has returned to /usr/include, and again contains the definitions for the C library time routines of *ctime*(3).

The *compact* and *uncompact* programs have been supplanted by the faster *compress*. If your user population has *compact*ed files, you will want to install *uncompact* found in /usr/src/old/compact.

The configuration of the virtual memory limits has been simplified. A MAXDSIZ option, specified in bytes in the machine configuration file, may be used to raise the maximum process region size from the default of 17 MB to 32 MB or 64 MB. The initial per-process limit is still 6 MB, but can be raised up to MAXDSIZ with the *csh limit* command.

Some IBM/4.3 binaries will not run with a 4.2/RT kernel because they take advantage of new functionality in IBM/4.3. One noticeable example of this problem is *csh*.

If you want to use *ps* after booting a new kernel, and before going multiuser, you must initialize its name list data base by running "ps –U."

2.7. Restoring Selected Files

The interactive option of *restore* lets you traverse the directories on the dump tape and select the files you want to restore.

(1) To restore selected files from the "rhd0a" tape, begin by typing the superuser command (**su**) and supplying the superuser password. Make sure the file systems are mounted, e.g. you are in multi-user mode.

(2) Type:

cd / < Enter >

restore if /dev/st0 < Enter >

(3) Respond to the *restore* prompt by typing:

restore > **add filenamex < Enter >**

where *filenamex* is the pathname of the file you want restored. For example, to restore the etc/hosts and etc/passwd files, you would type:

restore > **add etc/hosts < Enter >**
restore > **add etc/passwd < Enter >**

(4) To extract the desired file(s) from tape, type:

restore > **extract < Enter >**
restore > **quit < Enter >**

If at any time you are prompted for a tape or volume number, type **1**.

To restore files from the "rhd0g" tape, repeat the preceding steps, except in Step 2, type the following:

cd /usr < Enter >

instead of:

cd / < Enter >

To restore files from the "rhd1g" tape, repeat the preceding steps, except in Step 2, type the following:

cd /usr/src < Enter >

instead of:

cd / < Enter >

For more information on either *dump*(8) or *restore*(8), see the appropriate man page.

Note: 4.2/RT notesfiles are not readable by IBM/4.3.

2.8. Upgrading to an NFS system

In porting NFS to IBM/4.3, some changes were made to the system that may affect some user applications. There are several classes of programs that need to be recompiled in order to work correctly:

(1) Any program that looks at /etc/fstab. Since the format of /etc/fstab has changed, any programs that use the getfsent(3) functions will have to be relinked.

(2) Any program that does a read on a directory will have to be modified to use the functions listed in directory(3).

(3) Many programs that poke around in /dev/*mem* will have to be recompiled because the sizes and of some of the structures in the kernel have changed. Specifically, the *cmap, buf, mbuf, nameidate, file, text, unpcb, user,* and *inode* structures have changed.

Format changes in /etc/fstab require site-specific fstab entries to be converted. Here is a description of the changes:

(1) The fields are no longer colon separated; white space is used instead.

(2) There is now a third field that specifies the type of partition that is to be mounted. The old third field becomes the fourth field, and so on... The acceptable values for the type field are: (*ufs, nfs, swap,* or *ignore*). A typical fstab that used to look like:

```
/dev/hd0a:/:rw:1:1
/dev/hd0g:/usr:rw:1:2
/dev/hd1g:/usr/src:rw:1:2
/dev/hd2g:/space:rw:1:2
/dev/hd1b::sw::
```

Should look like:

```
/dev/hd0a / ufs rw 1 1
/dev/hd0g /usr ufs rw 1 2
/dev/hd1g /usr/src ufs rw 1 2
/dev/hd2g /space ufs rw 1 2
/dev/hd1b /swap swap sw 0 0
```

3. INSTALLATION PROCEDURES

This chapter, intended for the system manager, describes how to install IBM/4.3 on the IBM RT PC models 6150 and 6151 supported by this release and on the IBM 6152 Academic System.

3.1. Installation Procedures for the IBM RT PC

Instructions are provided for the following tasks:

- Changing disk partition sizes
- Installing IBM/4.3 on a model 6150 using streaming tape (**Note:** This requires two 70 MB disks)
- Installing IBM/4.3 on any model IBM RT PC from a model 6150 using a network connection

Upon successfully completing installation, a model 6150 with a system installed from streaming tape will contain a complete IBM/4.3 system, including all binary and source files. Any IBM RT PC model with a system installed across a network will contain the base system, which includes only binary files. If you want source files installed during a network installation, see the section "Changing Installation Options," later in this chapter.

There are five different configurations of systems that can be installed on IBM RT PC's or IBM 6152's: minimal, reduced, or full. The full installation is the only one that can be installed from an RT PC that has been installed using these directions. It is also the only system that can run standalone, that is without any other machines on a network. If you are going to be installing nfs minimal, minimal, nfs reduced, or reduced systems, you must set up a master machine and a server machine according to the directions in appendix C or appendix D. Here is a description of the five different system configurations:

nfs minimal:

The root partition is a bare-bones root. The real root partition is obtained via NFS. The /usr partition is also obtained via NFS.

minimal:

The root partition is a bare-bones root. The real root partition is obtained via NFS. The /usr partition (and any others) comes from the Andrew file system.

nfs reduced:

There is a normal root partition. The /usr partition is obtained via NFS.

reduced:

There is a normal root partition. The /usr partition (and any others) come from the Andrew File System.

full:

There are normal root and /usr partitions. Other partitions may be obtained from NFS or AFS.

3.1.1. Installing IBM/4.3 using Streaming Tape

This section describes how to use streaming tapes to install IBM/4.3 on a model 6150 IBM RT PC.

3.1.1.1. Before You Start

You should have the follow

- An installation "NIROOT
- A sautil diske
- Two stream ROOT/USER and SOURCE
- A model 6 ast two 70 MB hard disks

3.1.1.2. Tape Installati

Installing IBM/4.3 tape involves five major tasks:

(1) Form ng up minidisk tables with the SAUTIL diskette.

(2) Booting IBM/4.3 fro MINIROOT diskette

(3) Restoring the IBM/4.3 root and /usr file systems from the ROOT/USER streaming tape (about 30 minutes)

(4) Restoring the IBM/4.3 source file system (/usr/src) from the SOURCE streaming tape (about 30 minutes)

(5) Rebooting IBM/4.3 from the hard disk

(6) Making X-dependent special files

Note: By default, /usr/doc and /usr/sys are not loaded when you perform a tape installation. Actually, the header files in /usr/sys are installed, but the ".c" files are not. If you have a large enough root disk (114 Meg minimum), then you can use the "optional system components" menu option during the installation process to override the default in order to load either /usr/doc, /usr/sys, or both /usr/doc and /usr/sys.

If you need to load any components off of the root/usr tape at a later date, use /etc/restore. This will access any files on the tape that are normally installed in the root partition. To access any files that are normally installed in the usr partition, type the following before using /etc/restore:

```
# mt -f /dev/nrst0 rew <Enter>
# mt -f /dev/nrst0 fsf <Enter>
```

The next section provides step-by-step instructions for completing these tasks.

3.1.1.3. Tape Installation Steps

Information messages appear throughout the installation process. You may ignore all but those mentioned in these instructions.

If at any point in these installation steps an error should occur, see the section "Restarting After an Error," later in this chapter.

(1) Power on the tape drive and the display, if necessary.

(2) Insert the IBM RT PC Academic Operating Systems 4.3 standalone utilities diskette (sautil) in the diskette drive (the upper one, if there are two).

(3) Turn on the power. If it is already on, press **<Ctrl>-<Alt>-<Pause>**. (Press and hold the <Ctrl> key and the <Alt> key.

Then press the < Pause > key.)

(4) When the Standalone Utility Menu appears, respond to the *Choice?* prompt by typing **2 < Enter >** to select the "format" option. **Note:** If you do not respond to the Standalone Utility Menu within 30 seconds, the system will time out and the boot prompt will appear. If this occurs, press < Ctrl > -C to return to the Standalone Utility Menu.

You will now format the hard disk partitions for IBM/4.3.

(5) Respond to the prompt:

Device to format?

by typing:

hd(0,2) < Enter >

(6) When prompted to verify the format, type **yes < Enter >**.

(7) Press **< Enter >** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burnin is selected, you will be prompted for the "number of passes"; press **< Enter >** to select the default value.

(8) When asked whether to change the default parameters, type **no < Enter >**.

(9) When asked whether to preserve data already on the disk, type **no < Enter >**. If you have an old unix file system already on the disk and you want to preserve it, type **yes < Enter >**.

(10) When asked if the format process should start, type **yes < Enter >**.

For severe burnin, the system will take several hours to complete the format process.

(11) When the format is finished, press **< Enter >** to return to the main menu.

(12) At the *Choice?* prompt, type **10 < Enter >** to select the "minidisk" option, which creates IBM/4.3 minidisk partitions.

(13) When asked which disk to use, type **hd(0,2) < Enter >**.

(14) At the > prompt, type **init < Enter >**.

This initializes the minidisk table. The init command displays a warning message.

(15) Reply to the warning message:

Confirm (all data will be lost) [y/n]

by typing y to indicate you do want to proceed.

(16) Use the **standard** command to create default hd0a, hd0b, and hd0g partitions. Type:

standard root

(17) Type **quit < Enter >** to return to the main menu.

(18) At this point, if you are going to install utility source, the second disk, hd1, must also be formatted. If you are not installing utility source,

skip to step 33.

(19) Reply to the *Choice?* prompt, by typing **2** **<Enter>** to select the format option.

(20) Respond to the prompt:

Device to format?

by typing:

hd(1,2) **<Enter>**

(21) When prompted to verify the format, type **yes** **<Enter>**.

(22) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burning is selected, you will be prompted for the "number of passes"; press **<enter>** to select the default value.

(23) When asked whether to change the default parameters, type **no** **<Enter>**.

(24) When asked whether to preserve data already on the disk, type **no** **<Enter>**. If you have an old unix file system already on the disk and you want to preserve it, type **yes** **<Enter>**.

(25) When asked if the format process should start, type **yes** **<Enter>**.

For severe burnin, the system will take several hours to complete the format process.

(26) When the format is finished, press **<Enter>** to return to the main menu.

(27) At the *Choice?* prompt, type **10** **<Enter>** to select the "minidisk" option, which creates IBM/4.3 minidisk partitions.

(28) When asked which disk to use, type **hd(1,2)** **<Enter>**.

(29) At the > prompt, type **init** **<Enter>**.

This initializes the minidisk table. The init command displays a warning message.

(30) Reply to the warning message:

Confirm (all data will be lost) [y/n]

by typing y to indicate you do want to proceed.

(31) Use the **standard** command to create the hd1a and hd1g partitions, type:

standard source

This will create a "g" partition large enough (on a 70 megabyte disk) for the utility source with some room to spare. It will also create a 7 megabyte "a" partition that is convenient for the scratch space, or perhaps a "/tmp" partition.

If your second disk is smaller than 70 megabytes, 40 megabytes for example, then type:

standard single

instead of the above command. This will create a single partition that spans the entire disk.

(32) Type **quit** **< Enter >** to return to the main menu.

(33) Insert the diskette labeled MINIROOT in the diskette drive. When inserting a diskette into a diskette drive, always hold the diskette with the label side up and the notch on the side of the diskette facing left. If your IBM RT PC has two diskette drives, insert the diskette in the upper drive.

(34) Select the *boot* option by typing:

1 < Enter >

(35) The system will respond with a boot (:) prompt. Respond with:

fd(0,0)vmunix < Enter >

During the next two minutes, the IBM/4.3 kernel is read into memory and the system is initialized.

After initialization is complete, the screen will display instructions. Follow the instructions, which are paraphrased below.

Do **not** remove the MINIROOT diskette.

To correct typing errors, you can use the backspace key, located in the upper right corner of the typewriter keyboard and marked with a left-facing arrow.

(36) If this is the first time the system has been used, the time and date of the system clock may be incorrect. If so, the system clock will need to be reset before installing IBM/4.3.

Select the menu option to change the system clock. Note that the distributed system uses Pacific Time (Daylight or Standard, as appropriate). If your time zone is different, remember to allow for the difference until you configure a kernel for your time zone. For example, if the local time in New York is 11:05 am on 2 October 1987, one would type "8710020805". (8:05 is the Pacific Time corresponding to 11:05 Eastern Time.)

(37) Select the menu option to proceed with installation. You will be asked to specify tape or network installation. Select the menu option to specify tape installation. You will be asked if you wish to change the installation options.

If you do not want to load the utility source, you need to change the installation options. For "select optional system components," you need to select all except utility. That is, you need to select kernel, doc, man, font, learn, and notes. If you are loading the system onto a 70 meg disk, not all of the optional system components will fit in the /usr partition. By default, "doc", "kernel", and "AFS" are not installed. See the "Changing Installation Options", and "Installing the Andrew File System" sections later in this chapter for more information. For normal installation you should not change the installation options.

The tape installation program loads, and prompts you to insert the ROOT/USER tape cartridge:

> > > *Insert 'root/usr' tape and press <ENTER>*

(38) Insert the ROOT/USER tape cartridge in the tape drive and press **<Enter>**. When inserting a streaming tape cartridge into a tape drive, always hold the cartridge with the label side up and the "safe" turnscrew facing right.

During the next 30 minutes, the root and /usr file systems are restored from tape. The name of each file is displayed as the file is restored.

Skip to step 41 below if you did not select SOURCE as a system component. (See the section "Changing Installation Options" later in this chapter for more information.)

(39) The installation program prompts you to insert the SOURCE tape cartridge in the tape drive:

> > > *Insert 'source' tape and press <ENTER>*

Remove the ROOT/USER tape cartridge.

(40) Insert the SOURCE tape cartridge in the tape drive. Then press **<Enter>**.

During the next 30 minutes, the source file system (*/usr/src*) is restored from tape. After the restore is completed, remove the SOURCE tape cartridge.

(41) Select the menu option to halt the system for reboot.

When the system LEDs go out, remove the MINIROOT diskette and press:

<Ctrl>-<Alt>-<Pause>

(42) When the boot prompt (:) appears, press **<Enter>**. IBM/4.3 is loaded from the hard disk and proceeds with normal initialization, including autoboot and fsck. After about five minutes, you will see the multi-user prompt:

Academic Operating System 4.3 (master) (console)

login:

This completes the streaming tape installation. The model 6150 IBM RT PC contains all the binary files (and optionally all the source files) that comprise IBM/4.3.

3.1.1.4. Backing Up the Newly Installed System

Refer to the instructions in the section "Backing Up Your System" in Chapter 2, "Saving and Restoring Local Modifications."

3.1.1.5. Installation Verification

IBM/4.3 includes one guest account, named "guest," which has no password. To log in before your local accounts have been set up, respond to the login prompt

with "guest":

Academic Operating System 4.3 (master) (console)

login: **guest < Enter >**

Respond to the prompt

TERM = (ibm6155)

If you have an IBM 6155, simply press **< Enter >**.

If you have an IBM 6153, type:

ibm6153 < Enter >

If you have an IBM 6154, type:

ibm6154 < Enter >

If you have an IBM ACIS experimental display, type:

ibmaed < Enter >

If you have an IBM 5151 PC Monochrome display, type:

ibmmono < Enter >

If you have an IBM 5154 PC EGA display, type:

ibmega < Enter >

If you have an IBM 5081 Color display, type:

ibmmpel < Enter >

If you have an IBM vga display, type:

ibmvga < Enter >

If you have an IBM 8514 color display, type:

ibm8514 < Enter >

This completes the login process. The prompt *guest(n)* appears, where *(n)* is the chronological number of the prompt.

When you are logged in, you may run most of the commands found in Volume 1 of the *UNIX Programmer's Manual* or any of the sample programs supplied with this distribution.

After the *guest(n)* prompt appears, you may use the sample programs. To use the first sample program, type:

make sample1 < Enter >

When the *guest(n)* prompt reappears (in about one minute), type:

sample1 < Enter >

The message "Hello world" will appear on the screen, followed by the *guest(n)* prompt.

To use the second sample program, type:

make sample2 < Enter >

When the *guest(n)* prompt appears (in about one minute), type:

sample2 <Enter>

Wormlike patterns will appear to move around the screen. When you have seen enough, press **<Ctrl>-C** to terminate the program. To use the third sample program, type:

make sample3 <Enter>

When the *guest(n)* prompt appears (in about five minutes), type:

sample3 <Enter>

When the "$" prompt appears, you are in the Bourne shell, and you may issue commands such as "ls" or "date."

To exit the Bourne shell, press:

<Ctrl>-D

Note that, as distributed, neither the guest account nor the root account has any password security. Therefore, one of the first tasks of the system manager should be to establish user accounts and assign passwords. (See *passwd*(1).)

If you wish to log off the system at this time and do not intend to turn the power off, type:

logout <Enter>

This will return the *login* prompt. If you wish to turn the power off, see the section "Bootstrapping and Shutdown" in the chapter "System Operation," later in this document.

3.1.1.6. Customer Central Support Site Phone Numbers

When you have completed the system installation, and before proceeding to install the system across a network, you should add the telephone number of the designated customer support site to two files:

(1) Edit /etc/motd. (Its source is in /usr/src/etc/motd.) Replace nnn-nnnn with this telephone number, and delete the rest of the file.

(2) Edit /usr/ibm/support. (Its source is in /usr/src/ibm/support.sh.) Replace nnn-nnnn with this telephone number, and delete the rest of the file.

For further support information, see the article "Support Procedures" that accompanies the program directory.

3.1.1.7. Customizing Your System

Your system should now be customized, based on its intended use. During this process, you will be prompted for the type of each workstation on your system and, depending on type, for other information. The following are the workstation types and the additional information required for each:

Machine Type	Host Name	Serial Port Name	Modem Type	Long Distance Dial Digits	uucp Password	Network Device*
Master (m)	yes	no	no	no	no	yes
uucp connection (c)	yes	yes	yes	yes	yes	yes
Both master and connection (b)	yes	yes	yes	yes	yes	yes
Stand-alone or end use (s)	yes	no	no	no	no	yes

* Use un0 for Ethernet and lan0 for Token Ring.

We recommend you put your host names in the /etc/hosts file before you begin tailoring your system. Note also that if you specify a new host name, it will replace your old host name (if it exists) in the /etc/hosts file.

To tailor your system, first become the superuser by issuing the *su* command and entering the superuser password. Then use the command:

/etc/tailor <Enter>

3.1.2. Installing the Andrew File System

While installing from tape, it is possible to load the Andrew File System Client Binaries. All you have to do is to select the "afs" component from the miniroot's "Optional System Components" menu.

The Andrew File System Binaries can still be loaded onto an RT after the tape install. Follow the installation instructions in the document entitled "The IBM Andrew File System". If you do this now, return here when completed.

3.1.3. Installation Procedures for X and Andrew Toolkit

There are three tasks involved in installing X and Andrew Toolkit:

(1) Creating special devices for X

(2) Installing the X Window System

(3) Installing the Andrew Toolkit and application programs

The rest of this section describes these tasks.

3.1.3.1. Creating special devices for X

The set of devices that X needs has to be uniquely established on each machine. Since the RT PC and 6152 console and mouse devices share minor device numbers, only one set should exist on a given machine. Therefore, do the following for each machine, ignoring any *file exists* messages that may appear.

(1) Become the superuser by issuing the *su* command and supplying the superuser password.

(2) Type the following commands:

 # cd /dev <Enter>
 # ./MAKEDEV X <Enter>

If you are going to run X on an RT PC, the following command should also be issued:

 # rm *vga *8514 <Enter>

If you are going to run X on a 6152, issue the following command instead:

 # rm *mono *aed *ap* *mpel *cga <Enter>

3.1.3.2. Installing the X Window System

The X Window System is distributed on a single tape which contains both source (20 MB) and binaries (10 MB). We recommend you build the X Window System on a 70 MB file system. The installation involves copying the tape content to special directories, except for the man pages, which will be merged in with the system's other man pages, in /usr/man.

Where you decide to install X depends on how much space is available. If you will be installing X in /usr, proceed to section 3.1.3.2.1. If you will be installing X on another file system, proceed to section 3.1.3.2.2.

3.1.3.2.1. Install X in the /usr file system

Proceed with this section if you have enough space for the X Window System in the /usr file system. This procedure will remove and replace all X directories (/usr/include/X11, /usr/bin/X11, /usr/lib/*X*, /usr/src/X11, and part of /usr/man). You may wish to save these directories before proceeding.

To install X in the /usr file system, proceed as follows:

(1) If you are not the super user, become the super user by issuing the *su* command.

(2) Insert the tape labeled *X Version 11* into the tape drive.

(3) To install only binaries, type:

```
# rm -rf /usr/include/X11 /usr/bin/X11 /usr/lib/*X* <Enter>
# cd /usr <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - include bin lib man <Enter>
```

To install only source, type:

```
# rm -rf /usr/src/X11 <Enter>
# cd /usr <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - src <Enter>
```

To install both source and binaries, type:

```
# rm -rf /usr/include/X11 /usr/bin/X11 /usr/lib/*X* /usr/src/X11 <Enter>
# cd /usr <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - <Enter>
```

3.1.3.2.2. Installing X in Another File System

To conserve space or to isolate X from the rest of your system, you can install X in another file system other than /usr. We will use /space as an example. This procedure will remove and replace all X directories (/usr/include/X11, /usr/bin/X11, /usr/lib/*X*, /usr/src/X11, and part of /usr/man). You may wish to save these directories before proceeding.

Note: The following steps will install *most* of X11 in a separate file system. The man pages will still be installed in /usr/man however

To install X on a separate file system, proceed as follows:

(1) If you are not the super user, become the super user by issuing the *su* command.

(2) Insert the tape labeled *X Version 11* into the tape drive.

(3) To install only binaries, type:

```
# rm -rf /usr/include/X11 /usr/bin/X11 /usr/lib/*X* <Enter>
```

```
# cd /space <Enter>
# mkdir X11 <Enter>
# cd X11 <Enter>
# ln -s /usr/man . <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - include bin lib man <Enter>
# rm man <Enter>
# ln -s /space/X11/include/X11 /usr/include/X11 <Enter>
# ln -s /space/X11/bin/X11 /usr/bin/X11 <Enter>
# ln -s /space/X11/lib/*X* /usr/lib/ <Enter>
```

(4) To install only source, type:

```
# rm -rf /usr/src/X11 <Enter>
# cd /space <Enter>
# mkdir X11 <Enter>
# cd X11 <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - src <Enter>
# ln -s /space/X11/src/X11 /usr/src/X11 <Enter>
```

(5) To install both source and binaries, type:

```
# rm -rf /usr/include/X11 /usr/bin/X11 /usr/lib/*X* /usr/src/X11 <Enter>
# cd /space <Enter>
# mkdir X11 <Enter>
# cd X11 <Enter>
# ln -s /usr/man . <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - <Enter>
# rm man <Enter>
# ln -s /space/X11/include/X11 /usr/include/X11 <Enter>
# ln -s /space/X11/bin/X11 /usr/bin/X11 <Enter>
# ln -s /space/X11/lib/*X* /usr/lib/ <Enter>
# ln -s /space/X11/src/X11 /usr/src/X11 <Enter>
```

3.1.3.3. Installing the Andrew Toolkit and Application Programs

The Andrew Toolkit is distributed on a tape containing both source (24 MB) and binaries (16 MB). You must have installed X before installing the Andrew Toolkit and applications. Because the instructions in this section replace /usr/andrew, you may wish to save this directory before proceeding.

To install only binaries, proceed as follows:

(1) If you are not already the superuser, then issue the *su* command to become the superuser.

(2a) If you have 16 MB of free space in your /usr file system and you wish to locate Andrew there, type the following command:

```
# mkdir /usr/andrew <Enter>
```

(2b) If you do not have the space in your /usr file system or you would like to locate Andrew on another file system, issue the *mkdir* command on that file system. For example, if you have 16 MB of free space on /space, type the following:

```
# mkdir /space/andrew <Enter>
```

Then make a symbolic link to /space/andrew from /usr/andrew with the

following command:

```
# ln -s /space/andrew /usr/andrew <Enter>
```

(3) Insert the tape labeled "Andrew Toolkit" in your streaming tape drive and type the following commands:

```
# cd /usr/andrew <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - X11fonts bin dlib doc
                                    etc fonts help include lib <Enter>
```

To install BOTH source and binaries requires a 40 MB file system. If you plan to build Andrew, it will require a 70 MB file system. If you do not have a file system with the needed free space, you must build and mount one.

To install source and binaries proceed as follows:

(1) Become the superuser by typing the *su* command and entering the superuser password.

(2) Insert the tape labeled "Andrew Toolkit" in your streaming tape drive and type the following commands:

```
# mkdir /usr/andrew <Enter>
# cd /usr/andrew <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - < Enter>
```

3.1.3.3.1. Installing the Andrew File System console

The Andrew File System console is called "vopcon." It is furnished only in source form. If you intend to use vopcon to monitor the Andrew File System, you will have to remake the Andrew Toolkit after establishing the correct environment, as described below.

First, install the Andrew File System. Second, compile a new Andrew Toolkit. Do this by typing the following commands:

```
# cd /usr/andrew <Enter>
# make VICEDEFINES=-DVICE <Enter>
# make install <Enter>
# make ibmdoc <Enter>
```

3.2. Installing IBM/4.3 across a Network

This section describes installing IBM/4.3 on any model IBM RT PC across a network.

3.2.1. Before You Start

You should have the following:

- An installation diskette, labeled MINIROOT
- An sautil diskette
- A model 6150 IBM RT PC (with IBM/4.3 installed, both binary and source files) as the source machine
- Any model IBM RT PC as the target machine
- *Either*
  IBM Ethernet Adapters installed in both the source and target machines
  or
  IBM Token-Ring Adapters installed in both the source and target machines

- A network cable connecting the source and target machines

3.2.2. Network Installation Overview

Installing IBM/4.3 on the IBM RT PC through the network requires that there is a source IBM 6150 on the network and involves five major tasks:

(1) Formatting the disks and setting up minidisk partitions with the sautil diskette.

(2) Booting IBM/4.3 from the MINIROOT diskette on the target IBM RT PC

(3) Verifying the network connection between the source and the target machines

(4) Restoring the IBM/4.3 file systems from the source machine

(5) Rebooting IBM/4.3 from the target hard disk

Note that the network installation procedure installs only the base system by default. To include other system components, see the section, "Changing Installation Options," later in this chapter.

3.2.3. System Configurations for the IBM RT PC

The five available RT PC configurations are nfs minimal, minimal, nfs reduced, reduced, and full. (See the beginning of Chapter 3)

In order to install minimal or reduced systems on your RT, you must have already set up a master machine in the manner described in appendix C, or appendix D. In order to install an nfs reduced or an nfs minimal machine, you must have set up a master machine according to the instructions in appendix E.

3.2.4. System Type Requirements

You must have enough disk space available for whichever type of system that will be installed. The following table shows the minimum and recommended amount of disk space that is may be used.

	System Allocation	
System Type	minimum	recommended
nfs minimal	18 meg	20 meg
minimal	20 meg	20 meg
nfs reduced	33 meg	38 meg
reduced	36 meg	40 meg
full	60 meg	70 meg

3.2.5. Installing IBM/4.3 on an IBM RT PC

Follow these steps to install IBM/4.3 on an IBM RT PC.

(1) Insert the sautil diskette in the diskette drive (the upper one, if there are two).

(2) Turn on the power. If it is already on, press **<Ctrl>-<Alt>-<Pause>**. (Press and hold the <Ctrl> key and the <Alt> key. Then press the <Pause> key.)

(3) When the Standalone Utility Menu appears, respond to the *Choice?* prompt by typing **2** **<Enter>** to select the "format" option. **Note:** If you do not respond to the Standalone Utility Menu within 30 seconds, the system will time out and the boot prompt will appear. If this occurs, press <Ctrl>-C to return to the Standalone Utility Menu.

You will now format the hard disk partitions for IBM/4.3.

(4) Respond to the prompt:

Device to format?

by typing:

hd(0,2) **<Enter>**

(5) When prompted to verify the format, type **yes** **<Enter>**.

(6) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burnin is selected, you will be prompted for the "number of passes"; press **<Enter>** to select the default value.

(7) When asked whether to change the default parameters, type **no** **<Enter>**.

(8) When asked whether to preserve data already on the disk, type **no** **<Enter>**. If you have an old unix file system already on the disk and you want to preserve it, type **yes** **<Enter>**.

(9) When asked if the format process should start, type **yes** **<Enter>**.

For severe burnin, the system will take several hours to complete the format process.

(10) When the format is finished, press **<Enter>** to return to the main menu.

(11) At the *Choice?* prompt, type **10** **<Enter>** to select the "minidisk" option, which creates IBM/4.3 minidisk partitions.

(12) When asked which disk to use, type **hd(0,2)** **<Enter>**.

(13) At the > prompt, type **init** **<Enter>**.

This initializes the minidisk table. The init command displays a warning message.

(14) Reply to the warning message:

Confirm (all data will be lost) [y/n]

by typing y to indicate you do want to proceed.

(15) Use the **standard** command to create default hd0a, hd0b, and hd0g partitions. Note: the minimal system has no "g" partition. If you are installing an nfs minimal or a minimal system type:

standard minimal

Similarly, if you are installing an nfs reduced or a reduced system, type:

standard reduced

If you are installing a normal (full) system, type:

standard root

(16) Type **quit** **<Enter>** to return to the main menu.

(17) At this point, if you are going to install utility source, the second disk, hd1, must also be formatted. If you are not installing utility source, skip to step 32.

(18) Reply to the *Choice?* prompt, by typing **2** **<Enter>** to select the format option.

(19) Respond to the prompt:

Device to format?

by typing:

hd(1,2) **<Enter>**

(20) When prompted to verify the format, type **yes** **<Enter>**.

(21) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burning is selected, you will be prompted for the "number of passes"; press **<Enter>** to select the default value.

(22) When asked whether to change the default parameters, type **no** **<Enter>**.

(23) When asked whether to preserve data already on the disk, type **no** **<Enter>**. If you have an old unix file system already on the disk and you want to preserve it, type **yes** **<Enter>**.

(24) When asked if the format process should start, type **yes** **<Enter>**.

For severe burnin, the system will take several hours to complete the format process.

(25) When the format is finished, press **<Enter>** to return to the main menu.

(26) At the *Choice?* prompt, type **10** **<Enter>** to select the "minidisk" option, which creates IBM/4.3 minidisk partitions.

(27) When asked which disk to use, type **hd(1,2)** **<Enter>**.

(28) At the > prompt, type **init** **<Enter>**.

This initializes the minidisk table. The init command displays a warning message.

(29) Reply to the warning message:

Confirm (all data will be lost) [y/n]

by typing y to indicate you do want to proceed.

(30) Use the **standard** command to create the hd1a and hd1g partitions, type:

standard source

This will create a "g" partition large enough (on a 70 megabyte disk) for the utility source with some room to spare. It will also create a 7 megabyte "a" partition that is convenient for the scratch space, or perhaps a "/tmp" partition.

If your second disk is smaller than 70 megabytes, 40 megabytes for example, then type:

standard single

instead of the above command. This will create a single partition that

spans the entire disk.

(31) Type **quit** **< Enter >** to return to the main menu.

(32) Make sure that the source IBM RT PC is powered on, and that the files /etc/hosts and ~operator/.rhosts on the source IBM RT PC include appropriate entries for the target system. For the file ~operator/.rhosts, the entries must be of the form "*targetname* root" (ie *slave* root). the rest of the steps should be performed on the target system.

(33) Insert the diskette labeled MINIROOT in the upper diskette drive.

(34) Select the *boot* option by typing:

1 < Enter >

(35) The system will respond with a boot (:) prompt. Respond with:

fd(0,0)vmunix < Enter >

During the next two minutes, the IBM/4.3 kernel is read into memory and the system is initialized. Information messages appear throughout the installation process. You may ignore all but those mentioned in these instructions.

After initialization is complete, the screen will display instructions. Follow the instructions, which are paraphrased below.

Do **not** remove the MINIROOT diskette.

To correct typing errors, use the backspace key, located in the upper right corner of the typewriter keyboard and marked with a left-facing arrow.

(36) If this is the first time the system has been used, the time and date of the system clock may be incorrect. If so, the system clock will need to be reset before installing IBM/4.3.

Select the menu option to change the system clock. Note that the distributed system uses Pacific Time (Daylight or Standard, as appropriate). If your time zone is different, remember to allow for the difference until you configure a kernel for your time zone. For example, if the local time in New York is 11:05 am on 2 October 1986, one would type "8610020805". (8:05 is the Pacific Time corresponding to 11:05 Eastern Time.)

(37) Select the menu option to proceed with installation. You will be asked to specify tape or network installation. Select the menu option to specify network installation. You will be asked if you wish to change the installation options. See the "Changing Installation Options" section later in this chapter for more information. For normal installation you should not change the installation options. However, it is a good idea to say "yes" to verify that the correct system configuration has been selected (full, reduced, or minimal). If you are net-installing from a machine that has been set up in a manner such as that described in appendices C or D, that is, with the optional system components not really in the usr partition, but with symbolic links into the Andrew File System, then you should select all of the optional system components except source from the "Changing Installation Options" menu. This will insure that the symbolic links that are pointing into the Andrew File System get copied onto the target machine.

The network installation program loads, and verifies that the target system can connect to the source system via the network connection. The system

displays the following message:

> > > *establishing network connection from 'slave' to 'master'*

Once a connection is established, the installation program will begin to restore the root and /usr file systems (without kernel or documentation source files). This procedure will take approximately 25 minutes. If error messages occur during this procedure, the system may have failed to establish a network connection.

(38) This completes network installation. The target IBM RT PC contains all the binary files that comprise IBM/4.3.

Select the "Halt the system before rebooting from hard disk" menu option.

When the system LEDs go out, remove the MINIROOT diskette. To start the reboot, press:

< Ctrl > - < Alt > - < Pause >

(39) When the boot prompt (:) appears, press **< Enter >**.

IBM/4.3 is loaded from the hard disk and proceeds with normal initialization, including autoboot and fsck. After about five minutes, you will see the multi-user prompt:

Academic Operating System 4.3 (slave) (console)

login:

If you installed the RT from a master machine that already had X11 installed, or obtained X11 through the Andrew File System, and you want to use X11, then do the following to make sure the special devices required for X11 are set up:

(1) Log in as root.

(2) Type the following commands:

```
# cd /dev   < Enter >
# ./MAKEDEV X   < Enter >
# rm *8514 *vga   < Enter >
```

3.2.5.1. Installation Verification

See Section 3.1.2.5, "Installation Verification," above for instructions.

3.2.5.2. Customizing Your System

Your system should now be customized, based on its intended use as a standalone/end-user machine. Login as root, and use the command:

/etc/tailor < Enter >

You will be prompted for the workstation type (s for standalone/end-user machine) and network device (un0 for Ethernet and lan0 for Token Ring).

3.3. Installation Procedures for the IBM 6152 Academic System

3.3.1. Hardware Setup

These instructions assume you have set up the IBM 6152 hardware, as described in the manual *IBM 6152 Academic System Setup Guide*, (S68X-2209). After you have set up the system, be sure to run the reference diskette to initialize the configuration and date.

3.3.2. System Configurations for the IBM 6152

The five available 6152 configurations are nfs minimal, minimal, nfs reduced, reduced, and full. (See the beginning of Chapter 3)

In order to install minimal or reduced systems, you must have first set up a Master RT PC according to the instructions given in appendix C or appendix D. In order to install an nfs minimal or an nfs reduced system, you must have first set up a master machine according to the directions in appendix E. In order to install a full configuration 6152, you need an RT that has been installed with the base system and the 5 1/4" 6152 diskette. To install the 6152 diskette on the RT, login as root, insert the diskette, and type:

```
# cd /
# dd if=/dev/rfd0 bs=30b | uncompress | tar xfp -
```

3.3.3. Keyboard Notes

As you install IBM/4.3, keep these keyboard characteristics in mind:

- Use the Backspace key to correct errors, not the cursor control keys.
- These instructions will use the convention <Ctrl> to mean the *Control* key.

3.3.4. Configuring the Disk Partitions

You must determine the amount of space to allocate to each hard disk partition, using the information contained in this section. The partition sizes are in 512-byte blocks.

The following table lists the number of cylinders available on each type of disk, as well as the number of cylinders that comprise one megabyte (1 MB):

	20 MB	40 MB (type 31)	40 MB (type 32)	70 MB	114 MB
cylinders available	614	614	1022	70	114
cylinders/1MB	32	16	26	1	1

The following table shows the minimum and recommended sizes for the IBM/4.3 fdisk for the Unix partitions as a function of the system type:

	System Allocation	
System Type	minimum	recommended
nfs minimal	18 meg	20 meg
minimal	20 meg	20 meg
nfs reduced	33 meg	38 meg
reduced	36 meg	40 meg
full	60 meg	70 meg

You must set aside at least the above minimum values for IBM/4.3 on the hard disk as a function of the system type that you are setting up.

Note: The values for the full configuration workstation assume that you will not be installing any optional system components. You will probably need more space to load optional components. See "Changing Installation Options" later in this chapter for a table that shows disk space requirements for the individual components.

3.3.5. Installing DOS on a Hard Disk Partition

Skip this section if the entire disk is to be used for an IBM/4.3 filesystem. To proceed with this section, you must have a current DOS diskette.

(1) Follow the steps in the IBM DOS manual for installing DOS on a fixed disk. Use DOS to format your fixed disk.

(2) When the "A>" prompt returns, remove the DOS diskette and insert the IBM 6152 Academic System installation diskette.

(3) To copy the files that are on the installation diskette to the hard disk, type the following commands:

```
copy boot c:
copy unix.exe c:
copy nicpshh.exe c:
```

If you want IBM/4.3 to start automatically whenever you boot the system, type the following command:

```
copy autoexec.bat c:
```

If you do not copy autoexec.bat onto the hard disk, you can start IBM/4.3 by typing:

```
nicpshh
unix irq -3 go
```

This completes the steps necessary to install DOS on a hard disk partition, and prepare to install IBM/4.3. Continue now with the next section to install IBM/4.3.

3.3.6. Installing IBM/4.3 on an IBM 6152

Follow these steps to install IBM/4.3 on an IBM 6152.

(1) Insert the IBM 6152 Academic Operating Systems 4.3 installation diskette in the diskette drive (the upper one, if there are two).

(2) Turn on the power. If it is already on, press **<Ctrl>-<Alt>-<Delete>**. (Press and hold the <Ctrl> key and the <Alt> key. Then press the <Delete> key.)

(3) The Standalone Utility menu will appear after a moment. If you did not install a DOS partition earlier, then select the "installboot" option by typing "**12 <Enter>**". This will allow you to boot Unix off of the hard disk. Press **<Enter>** to return to the main menu.

(4) Respond to the *Choice?* prompt by typing **8 <Enter>** to select the "fdisk" option. **Note:** If you do not respond to the Standalone Utility Menu within 30 seconds, the system will time out and the boot prompt will appear. If this occurs, press **<Ctrl>**-C to return to the Standalone Utility Menu.

You will now create the hard disk partitions for IBM/4.3.

(5) Reply to the prompt:

Typical disk names are hd(unit,7) where unit = 0 or 1

by typing the following:

hd(0,7) <Enter>

(6) At this point, an IBM/4.3 partition must be created. If you are asked whether to initialize the fdisk table, type y **<Enter>**. Otherwise, a list of

existing fdisk partitions will be displayed. If there is already one, delete it and create a new one. This makes certain that the information that gets placed in the *config record* on the disk is correct. If there already is an IBM/4.3 partition on the disk, type:

delete <Enter>

and respond to the "which partition to delete" query with the number of the ACIS 4.3 partition.

Now type:

create <Enter>

(7) Respond to the prompt:

Starting Cylinder?

by typing the starting cylinder number for the IBM/4.3 partition. If you created a DOS partition in the preceding section, the starting cylinder number will be the same as the length of the DOS partition. Otherwise, the starting cylinder will be 0.

(8) If you plan on installing DOS later you will have to reserve space for it now. The DOS partition should be the first one on the disk. If you do not reserve space for DOS now but later decide you want to install it, you will have to backup and restore all the IBM/4.3 files on the disk.

(9) Reply to the prompt:

(nnn maximum) Partition Length?

by pressing **<Enter>** to accept the default maximum.

The partition length will be the maximum partition length displayed in the prompt.

(10) At the > prompt, type **quit <Enter>** to return to the main menu.

(11) At this point, unless the disk is of type ESDI, formatting is required. If you have an ESDI disk, skip to step 19 (the step where minidisk partitions are created).

At the *Choice?* prompt, type **2 <Enter>** to select the option to format this new partition.

(12) Respond to the prompt:

Device to format?

by typing:

hd(0,2) <Enter>

(13) When prompted to verify the format, type **yes <Enter>**.

(14) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burnin is selected, you will be prompted for the "number of passes"; press **<Enter>** to select the default value.

(15) When asked whether to change the default parameters, type **no <Enter>**.

(16) When asked whether to preserve data already on the disk, type **no <Enter>**. If you have an old unix file system already on the disk and you

want to preserve it, type **yes** <**Enter**>.

(17) When asked if the format process should start, type **yes** <**Enter**>.

For severe burnin, the system will take several hours to complete the format process.

(18) When the format is finished, press <**Enter**> to return to the main menu.

(19) At the *Choice?* prompt, type **10** <**Enter**> to select the "minidisk" option, which creates IBM/4.3 minidisk partitions.

(20) When asked which disk to use, type **hd(0,2)** <**Enter**>.

(21) If you are asked if you want to reinitialize th minidisk directory, type **y** <**Enter**>. Otherwise, at the > prompt, type **init** <**Enter**>.

This initializes the minidisk table. The init command displays a warning message.

(22) Reply to the warning message:

Confirm (all data will be lost) [y/n]

by typing y to indicate you do want to proceed.

(23) Use the **standard** command to create default hd0a, hd0b, and hd0g partitions. Note: the minimal system has no "g" partition. If you are installing an nfs minimal or a minimal system type:

standard minimal

Similarly, if you are installing an nfs reduced or a reduced system, type:

standard reduced

If you are installing a normal (full) system, type:

standard root

(24) Type **quit** <**Enter**> to return to the main menu.

(25) Reply to the *Choice?* prompt, by typing **9** <**Enter**> to select the "install" option. This copies the MINIROOT onto the hd0b (swap) partition and boots it.

(26) When the prompt to insert the installation MINIROOT diskette appears, simply press <**Enter**>. (The diskette is already in the drive.)

(27) At this point the standard installation miniroot menu appears. To correct typing errors, use the backspace key, located in the upper right corner of the typewriter keyboard and marked with a left-facing arrow.

(28) Make sure that the source IBM RT PC is powered on, and that the files /etc/hosts and ~operator/.rhosts on the source IBM RT PC include appropriate entries for the target system. For the file ~operator/.rhosts, the entries must be of the form "*targetname* root" (ie *slave* root). The rest of the steps should be performed on the target system.

(29) If this is the first time the system has been used, the time and date of the system clock may be incorrect. If so, the system clock will need to be reset before installing IBM/4.3.

Select the menu option to change the system clock. Note that the distributed system uses Pacific Time (Daylight or Standard, as appropriate). If your time zone is different, remember to allow for the difference until you configure a kernel for your time zone. For example, if the local time in

New York is 11:05 am on 2 October 1986, one would type "8610020805". (8:05 is the Pacific Time corresponding to 11:05 Eastern Time.)

(30) Select the menu option to proceed with installation. You will be asked if you wish to change the installation options. See the "Changing Installation Options" section later in this chapter for more information. For normal installation you should not change the installation options. However, it is a good idea to say "yes" to verify that the correct system configuration has been selected (full, reduced, or minimal). If you are net-installing from a machine that has been set up in a manner such as that described in appendices C or D, that is, with the optional system components not really in the usr partition, but with symbolic links into the Andrew File System, then you should select all of the optional system components from the "Changing Installation Options" menu. This will insure that the symbolic links that are pointing into the Andrew File System get copied onto the target machine.

The network installation program loads, and verifies that the target system can connect to the source system via the network connection. The system displays the following message:

> > > *establishing network connection from 'slave' to 'master'*

Once a connection is established, the installation program will begin to restore the root and /usr file systems (without kernel or documentation source files). This procedure will take approximately 25 minutes. If error messages occur during this procedure, the system may have failed to establish a network connection.

(31) This completes network installation. The target IBM 6152 contains all the binary files that comprise IBM/4.3.

Select the "Halt the system before rebooting from hard disk" menu option.

If you have set up the hard disk to boot IBM/4.3 then remove the MINIROOT diskette when the *DOS A>* prompt appears. To start the reboot, press:

< Ctrl > - < Alt > - < Delete >

(32) When the boot prompt (:) appears, press **< Enter >**.

IBM/4.3 is loaded from the hard disk and proceeds with normal initialization, including autoboot and fsck. After about five minutes, you will see the multi-user prompt:

Academic Operating System 4.3 (slave) (console)

login:

When you have completed the steps in that section, you will have successfully installed IBM/4.3 on your IBM 6152.

If you installed the 6152 from a master machine that already had X11 installed, or obtained X11 through the Andrew File System, and you want to use X11, then do the following to set up the special devices needed for X11:

(1) Log in as root.

(2) Type the following commands:

 # cd /dev < Enter >

```
# ./MAKEDEV X  <Enter>
# rm *ap* *ega *aed *mpel *mono  <Enter>
```

3.3.6.1. Installation Verification

See Section 3.1.2.5, "Installation Verification," above for instructions.

3.4. Changing Installation Options

Users may choose to modify the default installation options, which are described in this section.

Help This option displays memory-jogging information based on the rest of this section.

Select configuration

This option may be set to "nfs minimal", "minimal", "nfs reduced", "reduced", or "full". An "nfs minimal" system obtains both the root and /usr file system via the Network File System (NFS). A "minimal" system obtains the root File system via NFS, and the /usr file system via the Andrew File System (AFS). An "nfs reduced" system has a local root file system, and obtains the /usr file system via AFS. A "full" system has both the root and /usr file systems on the local hard disk.

The default configuration is whichever one that the disk is large enough to accommodate. For example, "full" is the default on a 70 MB hard disk. The "nfs minimal" and "nfs reduced" systems are never chosen as the default.

Select root device

This option specifies which hard disk to use as the root device for IBM/4.3. The default is hd0. For the 6152, hd1 is also supported. For the RT, hd1 and hd2 are also supported. Installing IBM/4.3 on an RT's hd1 or hd2 can be useful when it will co-reside with AIX. (Please refer to Appendix A of this article for more information on co-residence.) Note however that using hd2 on an RT for the root device prevents installing the source for utilities.

Change output

This option specifies either verbose or terse output. The default is verbose. Note that terse output results in long periods of time with no activity on the screen.

Select optional system components

In addition to the base IBM/4.3 system, the optional system components selected will be installed. The default is "none" for network installation. The default is "man font learn notes utility" for tape installation. The default for the network installation is "none". One or more of the following may be selected:

Name	Description
kernel	The kernel source in /usr/sys. This is not available on a 6152.
doc	The documentation source in /usr/doc.
man	The online man pages in /usr/man.
font	The 3812 font libraries in /usr/lib/font.
learn	The *learn*(1) data base in /usr/lib/learn.
notes	The *notes*(1) system in in /usr/spool/notes.
X11	The X11 binaries and source code in the /usr and /usr/src file systems. This option is only available for network installation.
utility	The utility source in /usr/src.
afs	The Andrew File System client binaries. this option is only available for the tape installation.
all	All of the above.
none	None of the above.

NOTE: For tape installation, /sys/support is not optional (it is part of the base system). It is not available for network installation.

The following table of component sizes should help you determine the minimum file system size necessary for the components you wish to install.

Component	Partition	Minimum Size	Standard Sizes 40M	70M	114M
base system	hd0a	11937	*	15267	15022
6152 feature	hd0a	729	*	15267	15022
base system	hd0g	20628	*	35180	73583
6152 feature	hd0g	1563	*	35180	73583
kernel	hd0g	6956	*	35180	73583
doc	hd0g	7081	*	35180	73583
man	hd0g	4011	*	35180	73583
font	hd0g	881	*	35180	73583
learn	hd0g	979	*	35180	73583
notes	hd0g	706	*	35180	73583
afs	hd0g	300#	*	35180	73583
utilities	hd1g	34778	40792	58511	97003

* not supported
# this does not include space required for the cache directory

Note: Since the Berkeley file system allocation strategy is only efficient when the file system is less than 90% full, an "artificial" 90 % maximum is imposed on all users except root. This means that you need to allow space for approximately 110% of all of the components you plan to install for each partition. Also, the above numbers are approximations. Therefore, it is a good idea to avoid using every last block on the file system; the files might not fit.

Change network preference

This option specifies which network adapter the installation procedure attempts to configure first. The option toggles between Ethernet (the default) and IBM Token Ring. If your system has only one network adapter, this option is irrelevant. This option applies only to network installations.

Select source host

This option specifies the system from which IBM/4.3 is to be downloaded. The value of the option may be a hostname (if the diskette was created with the site's local host table). Otherwise, the value is an internet address. The default is "master." This option applies only to network installations.

Select target host

This option specifies the system onto which IBM/4.3 is to be installed. The value of the option may be a hostname (if the diskette was created with the site's local host table). Otherwise, the value is an internet address. The default is "slave." This option applies only to network installations.

Additional Parameters for ifconfig

This option allows you to specify additional parameters that will be passed to ifconfig when the network is started up. It only applies to network installation.

If the host table on the MINIROOT diskette has been updated to include local host-names, they may be used. Internet addresses may be used regardless of the state of the MINIROOT diskette's host table.

Proceed with installation

Select this option when you are ready to proceed. You will be given an opportunity to verify your work before installation actually proceeds.

Abort installation

This option returns you to the main installation menu, where you can set the date, halt the system, invoke the Bourne shell, or initiate installation.

3.5. Notes for Experienced Users

The *restore.tape*(8R) and *restore.net*(8R) commands can also be used directly, by invoking an interactive Bourne shell from the main installation menu. This step replaces the use of the installation menu interface. When finished, leave the Bourne shell by typing **<Ctrl>-D**, and halt the system.

Non-standard file systems may be installed from *dump*(8) format tapes with *restore.tape*, and non-standard file systems may be installed over the network with *restore.net*. The user is referred to the appropriate man pages. Normal dump tapes of the root, /usr, and /usr/src file systems (but *not* the distribution tapes) may be installed with the command

restore.tape prompt=root hd0a prompt=user hd0g prompt=source hd1g <Enter>

Distribution tapes may be installed with the command

restore.tape prompt=root/user hd0a hd0g prompt=source hd1g <Enter>

3.6. When root is not hd0

You must take special steps to boot IBM/4.3 when the root partition is not */dev/hd0a*. You must explicitly give the kernel name to the boot program; also, you must give the generic (distribution) kernel the location of the root partition at boot time.

For example, assume that we have installed IBM/4.3 on */dev/hd1a*. The following dialogue would take place when booting a generic kernel. System messages are in *italic*; user responses are in **bold**.

4.2 BSD UNIX Standalone Boot Program $Revision: X.Y$

Default: hd(0,0)vmunix (just press Enter or wait ~30 seconds)

: **hd(2,0)vmunix**
. . . (many messages)
root device? **hd2**
. . . (more messages)
#

You will need to preen the disks manually (issue *fsck -p*), and then enter <Ctrl>-D to bring the system up in multiuser mode. Once you have a built a non-GENERIC kernel with hd1 as its root device, you will no longer see the *root device?* question, although you will have to provide the kernel name to boot. For these reasons, you will eventually want to move the root partition to hd0 (after building a kernel to support root on hd0).

4. SYSTEM SETUP

This chapter describes procedures used to set up a IBM/4.3 system. Use these procedures after you first install your system or when your system configuration changes. Procedures for normal system operation are described in Chapter 6.

4.1. Kernel Configuration

This section briefly describes the layout of the kernel code and how files for devices are made. For a full discussion of configuring and building system images, consult the article "Building IBM/4.3 Systems with Config" in the manual.

4.1.1. Kernel Organization

As distributed, the kernel source is in /usr/sys. The source may be physically located anywhere within any file system as long as a symbolic link to the location is created for the file /sys. (Many files in /usr/include are normally symbolic links relative to /sys.) In further discussions of the system source, all path names will be given relative to /sys.

The directory /sys/sys contains the mainline machine-independent operating system code. Files within this directory are conventionally named with the following prefixes:

init_	system initialization
kern_	kernel (authentication, process management, etc.)
quota_	disk quotas
sys_	system calls and similar
tty_	terminal handling
ufs_	file system
uipc_	interprocess communication
vm_	virtual memory

The remaining directories are organized as follows.

/sys/h	machine independent include files
/sys/conf	site configuration files and basic templates
/sys/net	network independent, but network related code
/sys/netinet	DARPA Internet code
/sys/netimp	IMP support code
/sys/netns	Xerox NS support code
/sys/ca	RT specific mainline code
/sys/caif	RT network interface code
/sys/caio	RT device drivers and related code
/sys/cacons	RT console device drivers and related code

Many of these directories are referenced through /usr/include with symbolic links. For example, /usr/include/sys is a symbolic link to /sys/h. The system code, as distributed, is totally independent of the include files in /usr/include. This allows the system to be recompiled from scratch without the /usr file system mounted.

4.1.2. Devices and Device Drivers

Devices supported by IBM/4.3 are implemented in the kernel by drivers whose source is kept in /sys/ca, /sys/caio, and /sys/cacons. These drivers are loaded into the system when included in a cpu-specific configuration file kept in the conf directory. Devices are accessed through special files in the file system, made by the *mknod*(8) program, and normally kept in the /dev directory. For devices supported by the distribution system,

files are created in /dev by the /dev/MAKEDEV shell script.

Determine the set of devices that you have and create a new /dev directory by running the MAKEDEV script. First create a new directory /newdev; copy MAKEDEV into it; edit the file MAKEDEV.local to provide an entry for local needs; run it to generate a /newdev directory. For instance, if your machine has one hard disk and a diskette, you would type the following:

```
#cd <Enter>
#mkdir newdev <Enter>
#cp dev/MAKEDEV newdev/MAKEDEV <Enter>
#cd newdev <Enter>
#MAKEDEV hd0 fd0 std local <Enter>
```

Note the "std" argument causes standard devices such as /dev/console (the machine console) to be created.

You can then type:

```
#cd <Enter>
#mv dev olddev ; mv newdev dev <Enter>
#sync <Enter>
```

to install the new device directory.

4.1.3. Building New System Images

The kernel configuration of each IBM/4.3 system is described by a single configuration file, stored in the /sys/conf directory. To learn about the format of this file and the procedure used to build system images, you should:

- Read "Building IBM/4.3 Systems with Config" in *IBM Academic Operating System 4.3.*
- Study the manual pages for the devices you have (See *IBM Academic Operating System 4.3* or the *UNIX Programmer's Reference Manual*).
- Review the sample configuration file in the /sys/conf directory.

The configured system image "vmunix"[2] should be copied to the root and then booted to try it out. It is best to name it /newvmunix so as not to destroy the working system until you are sure it does work:

```
# cp vmunix /newvmunix <Enter>
# sync <Enter>
```

It is also a good idea to save the old system under some other name. In particular, we recommend that you save the generic distribution version of the system permanently as /genvmunix for use in emergencies.

To boot the new version of the system, power on the IBM RT PC. If it's already on, you can perform a hardware boot by using the *reboot*(8) command. At the boot prompt, type:

```
hd(0,0)newvmunix
```

After having booted and tested the new system, it should be installed as /vmunix before going into multiuser operation. A systematic scheme for numbering and saving old

[2]A system configured with the debugger is called "vmunix.ws".

versions of the system may be useful.

4.2. Disk Configuration

This section describes how to lay out file systems to make use of the available space and to balance disk load for improved system performance.

4.2.1. Initializing /etc/fstab

Change into the directory /etc and copy the appropriate file from:

fstab.hd.1 (for a one-disk, desk model system)
fstab.hd.3 (for a two- or three-disk, floor model system)

to the file /etc/fstab, i.e.:

 # cd /etc <Enter>
 # cp fstab.hd.*x* fstab <Enter>

where *x* is either 1 or 3.

This will set up the initial information about the usage of disk partitions. Note that the installation procedures attempt to do this automatically during installation.

4.2.2. Disk Naming and Divisions

Each physical disk drive can be divided into up to eight partitions; IBM/4.3 typically uses only three or four partitions. The first partition (hd0a) stores a root file system from which IBM/4.3 can be bootstrapped. The second partition (hd0b) is used for paging and swapping. The third partition (hd0g) is used for the /usr file system.

The disk partition sizes for a drive are based on a set of defaults that are programmed into the minidisk program. The minidisk program manipulates entries in a partition table on the hard disk that defines where on the disk and how big each partition is. The "a" partition is roughly 32767 sectors on full configured systems. On reduced configurations the "a" partition is a little smaller, about 30000 sectors. Systems with minimal configurations have "a" partitions whose size is really a function of the size of the entire disk. Typically an "a" partition is about 20000 sectors on a minimal configured machine. The "b" partition is 33440 sectors on both full and reduced configured systems. The "b" partition on minimal systems is also a function of the size of the disk. Typically the size will be about 18000 sectors. The "c" partition is large enough to access the entire disk, including the space at the front of the disk reserved for the bad sector forwarding table, and the space at the end of the disk containing the pool of replacement sectors. The "g" partition is the remainder of the disk, after the "a", and "b" partitions. There is no "g" partition on minimal configured systems.

Non-standard partition sizes may also be specified on a per disk basis (see *minidisk*(8R)).

4.2.3. Layout Considerations

There are several considerations in deciding how to arrange your disks. Two major considerations are adequate space and adequate throughput. Paging space is an important parameter. The system, as distributed, sizes the configured paging areas each time the system is booted. Further, multiple paging areas of different size may be interleaved. Drives smaller than 356 megabytes have swap partitions of 16 megabytes, while drives larger than 355 megabytes have 32 megabytes. These values may be changed to get more paging space by changing the appropriate minidisk table on the disk.

Many common system programs (C, the editor, the assembler, etc.) create intermediate files in the /tmp directory, so the file system where /tmp is stored should be large

enough to accommodate most high-water marks. If you have several disks, mount /tmp in a root (i.e. first partition) file system on another disk. All programs that create files in /tmp also delete them but may leave dregs. Examine the directory periodically and delete old files.

On a single-disk system, there may not be sufficient free space on the root file system for /tmp. You can replace /tmp with a symbolic link to /usr/tmp on the hd0g partition, which should have sufficient space.

The efficiency with which IBM/4.3 is able to use the CPU is often strongly affected by the configuration of disk controllers. For general time-sharing applications, the best strategy is to try to split the root file system (/), system binaries (/usr), the temporary files (/tmp), and the user files among several disk arms, and to interleave the paging activity among several arms.

It is critical for good performance to balance disk load. There are at least five components of disk load that you can divide between available disks:

1. The root (/) file system.
2. The /tmp file system.
3. The /usr file system.
4. The user files.
5. The paging activity.

The following possibilities are ones Berkeley has used when they have had two or three disks:

	disks	
what	2	3
/	0	0
tmp	1	2
usr	1	1
paging	0+1	0+2
users	0	0+2

You should try to even out the disk load as much as possible by locating on separate arms those file systems between which heavy copying occurs. Note that long-term balancing of the load is not important; it is much more important to balance the load properly for when the system is busy.

Intelligent experimentation with a few file system arrangements can pay off in much improved performance. It is particularly easy to move the root, the /tmp file system and the paging areas. Place the user files and the /usr directory as space dictates, and experiment with the other, more easily-moved file systems.

4.2.4. File System Parameters

Each file system has associated parameters describing its block size, fragment size, and the geometry characteristics of the disk on which it resides. Inaccurate specification of the disk characteristics or haphazard choice of the file system parameters can cause substantial throughput degradation or significant wasted disk space. As distributed, file systems are configured according to the following table.

File system	Block size	Fragment size
/	8 Kbytes	1 Kbyte
usr	4 Kbytes	1 Kbyte
users	4 Kbytes	1 Kbyte

The root file system block size is made large to optimize bandwidth to the associated disk. This large block size is important because the /tmp directory is normally part of the root file or a similar file system. The large block size is also important because many of the most heavily used programs are demand-paged out of the /bin directory. The fragment size of 1 Kbyte is a "nominal" value to use with a file system. With a 1-Kbyte fragment size, disk space utilization is approximately the same as with earlier versions of the file system.

The /usr file system uses a 4-Kbyte block size with 1-Kbyte fragment size to achieve high performance.

File systems for users have a 4-Kbyte block size with 1-Kbyte fragment size. These parameters have been selected based on the performance of Berkeley's user file systems. The 4-Kbyte block size provides adequate bandwidth while the 1-Kbyte fragment size provides acceptable space conservation and disk fragmentation.

You may choose other parameters in constructing file systems, but the factors involved in block size and fragment size are many and interact in complex ways. Larger block sizes result in better throughput to large files in the file system, because larger I/O requests can be performed. However, you should consider the average file sizes found in a file system and the performance of the internal system buffer cache. The system provides space in the inode for 12 direct block pointers, one single indirect block pointer, and one double indirect block pointer.[3] If a file uses only direct blocks, you can optimize access time to it by maximizing the block size. If a file spills over into an indirect block, increasing the block size of the file system may decrease the amount of space used (by eliminating the need to allocate an indirect block). However, if you increase the block size, and an indirect block is still required, the file will use more disk space (because indirect blocks are allocated according to the block size of the file system).

In selecting a fragment size for a file system, you must consider at least two things. The major performance tradeoffs are between an 8-Kbyte block file system and a 4-Kbyte block file system. Because of implementation constraints, the ratio of block size to fragment size cannot be greater than 8. An 8-Kbyte file system will always have a fragment size of at least 1 Kbyte. If a file system is created with a 4-Kbyte block size and a 1-Kbyte fragment size, and then upgraded to an 8-Kbyte block size and 1-Kbyte fragment size, identical space conservation occurs. However, if a file system has a 4-Kbyte block size and 512-byte fragment size, converting it to an 8K/1K file system causes significantly more space to be used. A 4-Kbyte block file system which might be upgraded to 8-Kbyte blocks for higher performance should use fragment sizes of at least 1 Kbyte to minimize the amount of work required in conversion.

A second, more important, consideration when selecting the fragment size for a file system is the level of fragmentation on the disk. With a 512-byte fragment size, storage fragmentation occurs much sooner, particularly with a busy file system running near full capacity. By comparison, the level of fragmentation in a 1-Kbyte fragment file system is much less severe. On file systems where many files are created and deleted, the 512-

[3] A triple indirect block pointer is also reserved, but not supported.

byte fragment size is more likely to result in apparent space exhaustion because of fragmentation. That is, when the file system is nearly full, file expansion that requires locating a contiguous area of disk space is more likely to fail on a 512-byte file system than on a 1-Kbyte file system. To minimize fragmentation problems of this sort, a parameter in the super block specifies a minimum acceptable free space threshold. When anyone but the super-user attempts to allocate disk space and the free space threshold is exceeded, the user is returned an error as if the file system were actually full. This parameter is nominally set to 10%, and can be changed by supplying a parameter to *newfs*, or by updating the super block of an existing file system using *tunefs*(8).

In general, unless a file system is to be used for a special purpose application (for example, storing image processing data), Berkeley recommends using the default values supplied. Remember that the current implementation limits the block size to at most 8 Kbytes and the ratio of block size to fragment size must be 1, 2, 4, or 8.

The disk geometry information used by the file system affects the block layout policies employed. The file /etc/disktab, as supplied, contains the data for drives supported by the system. When constructing a file system you should use the *newfs*(8) program and specify the type of disk on which the file system resides. This file also contains the default file system partition sizes, and default block and fragment sizes. To override any of the default values you can modify the file or use an option of *newfs*.

4.2.5. Implementing a Layout

To put a chosen disk layout into effect, use *newfs*(8) to create each new file system, and add its name to the file /etc/fstab. The system will check and mount each file system found in /etc/fstab when the system is bootstrapped.

Consider a system with 70-megabyte drives. On the first drive, (hd0), we put the root file system in hd0a and the /usr file system in hd0g. The /tmp directory is part of the root file system because no file system is mounted on /tmp. On a one-drive model, we put user files in the hd0g partition with the system binaries.

On a three-drive model, we create a file system in hd1g and put user files there, calling the file system /mnt. We interleave the paging between the first and second drives. To do this, we build a system configuration that specifies:

```
config   vmunix root on hd0      swap on hd0 and hd1
```

to get the swap interleaved. /etc/fstab would then contain:

```
/dev/hd0a / ufs rw 1 1
/dev/hd0g /usr ufs rw 1 2
/dev/hd0b /swap swap sw 0 0
/dev/hd1g /mnt ufs rw 1 2
```

We keep a backup copy of the root file system in the hd1a disk partition.

To make the /mnt file system, we use the following commands:

```
#cd /dev <Enter>
#./MAKEDEV hd1 <Enter>
#newfs hd1g <Enter>
```

After information about the file system prints, we type the following commands:

```
#mkdir /mnt <Enter>
#mount /dev/hd1g /mnt <Enter>
```

4.3. Setup for Remote Virtual Disks (RVD)

For information regarding installing and setting up remote virtual disks, see the "The Remote Virtual Disk System" article.

4.4. Configuring Terminals

If IBM/4.3 is to support simultaneous access from multiple terminals, you must edit the file /etc/ttys. (See *ttys*(5).)

Terminals connected to the system via RS232 ports are conventionally named **ttyxx** where xx identifies the specific line. The lines on port RS232 cards are named /dev/tty00, /dev/tty01, . . . , /dev/tty15 (up to four cards may be installed). The planar serial ports are known as /dev/ttys0 and /dev/ttys1. The asynchronous communications cards are known as /dev/ttyc0 and /dev/ttyc1.

To add a new terminal device, be sure the device is configured into the system and that the special files for the device have been made by /dev/MAKEDEV. Then, enable the appropriate lines of /etc/ttys by setting the "status" field to **on** (or add new lines to /etc/ttys). Note that lines in /etc/ttys are one-for-one with entries in the file of current users (/etc/utmp), and therefore it is best to make changes while running in single-user mode and to add all of the entries for a new device at once.

The format of the /etc/ttys file is completely new in IBM/4.3. Each line in the file is broken into four tab separated fields (comments are shown by a '#' character and extend to the end of the line). For each terminal line the four fields are: the device (without a leading /dev), the program /etc/init should startup to service the line (or **none** if the line is to be left alone), the terminal type (found in /etc/termcap), and optional status information describing if the terminal is enabled or not and if it is "secure" (i.e. the super user should be allowed to login on the line). All fields are character strings with entries requiring embedded white space enclosed in double quotes. Thus a newly added terminal /dev/tty00 could be added as

```
tty00    "/etc/getty std.9600"    apa16    on secure        # mike's office
```

The std.9600 parameter provided to /etc/getty is used in searching the file /etc/gettytab; it specifies a terminal's characteristics (such as baud rate). To make custom terminal types, consult *gettytab*(5) before modifying /etc/gettytab.

Dialup terminals should be wired so that carrier is asserted only when the phone line is dialed up. For non-dialup terminals from which modem control is not available, you must either wire back the signals so that the carrier appears to always be present, or show in the system configuration that carrier is to be assumed to be present with *flags* for each terminal device. See *asy*(4) and *psp*(4) for details.

For network terminals (i.e. pseudo terminals), no program should be started up on the lines. Thus, the normal entry in /etc/ttys would look like

```
ttyp0    none    network
```

(Note the fourth field is not needed for these entries.)

When the system is running multi-user, all terminals that are listed in /etc/ttys as **on** have their line are enabled. If, during normal operations, it is desired to disable a terminal line, you can edit the file /etc/ttys to change the terminal's status to **off** and then send a hangup signal to the *init* process, using the following command:

```
# kill -1 1 <Enter>
```

Terminals can similarly be enabled by changing the status field from **off** to **on** and sending a hangup signal to *init*.

Note that if a special file is inaccessible when *init* tries to create a process for it, *init* will log a message to the system error logging process (/etc/syslogd) and try to reopen the terminal every minute, reprinting the warning message every 10 minutes. Messages of this sort are normally printed on the console, though other actions may occur depending on the configuration information found in /etc/syslog.conf.

Finally note that you should change the names of any dialup terminals to ttyd? where ? is in [0-9a-zA-Z], as some programs use this property of the names to determine if a terminal is a dialup. Shell commands to do this should be put in the /dev/MAKEDEV.local script.

While it is possible to use truly arbitrary strings for terminal names, the *ps*(1) command makes good use of the convention that tty names (by default, and also after dialups are named as suggested above) are distinct in the last 2 characters. We don't recommend you change this; the heuristics *ps*(1) uses that are based on these conventions may break down and *ps* will run MUCH slower.

4.5. Adding Users

You can add new users to the system by adding a line to the password file /etc/passwd. The procedure for adding a new user is described in *adduser*(8). You should add accounts for the initial user community, give each a directory and a password, and put users who wish to share software in the same group.

4.6. Site Tailoring

All programs that require the site name or some similar characteristic obtain the information through system calls or from files located in /etc. To supply a site name, edit the file /etc/rc.config. A line in this file,

hostname = *hostname*

defines the value returned by the *gethostname* system call. If you are running the name server, your site name should be your fully qualified domain name. Programs such as *getty*(8), *mail*(1), *wall*(1), and *who*(1) use this system call so that the binary images are site-independent. IBM/4.3 provides the *tailor*(8) command to facilitate site tailoring.

4.7. Setting Up the Line Printer System

The line printer system consists of at least the following files and commands:

/usr/ucb/lpq	spooling queue examination program
/usr/ucb/lprm	program to delete jobs from a queue
/usr/ucb/lpr	program to enter a job in a printer queue
/etc/printcap	printer configuration and capability data base
/usr/lib/lpd	line printer daemon, scans spooling queues
/etc/lpc	line printer control program
/etc/hosts.lpd	list of hosts allowed to use the printers

The file /etc/printcap is a master data base describing both line printers directly attached to a machine and printers accessible across a network. The manual page *printcap*(5) describes the format of this data base and shows the default values for such things as the directory in which spooling is performed. The line printer system handles multiple printers, multiple spooling queues, local and remote printers, and printers attached via serial lines that require line initialization such as the baud rate.

Remote spooling via the network is handled with two spooling queues, one on the local machine and one on the remote machine. When a remote printer job is initiated with *lpr*,

the job is queued locally and a daemon process is created to oversee the transfer of the job to the remote machine. If the destination machine is unreachable, the job will remain queued until it is possible to transfer the files to the spooling queue on the remote machine. The *lpq* program shows the contents of spool queues on both the local and remote machines.

To configure your line printers, consult the *printcap*(5) man page and the article entitled "4.3BSD Line Printer Spooler Manual" in the *UNIX System Manager's Manual.* Include a call to *lpd*(8) in /etc/rc. (See also *ibm3812*(8) and *ppt*(8).)

4.8. Setting up the IBM 4216 Personal Pageprinter

The PostScript interpreter and additional fonts for the IBM Personal Pageprinter are not shipped with IBM/4.3. They must be loaded by the user when the first 4216 is installed. After these files are loaded, they may be used with any number of printers on a single machine.

Setting up the 4216 Personal Pageprinter consists of:

- For the IBM 6152 only, the following steps must be done to begin with.

(1) Adding the 4216 Personal Pageprinter option to the 6152 reference disk. See "Configuring Your System the PS/2 Mod 60 Guide to Ops."

(2) Configuring the 6152 to include the 4216 adapter card in its configuration. The ps device number corresponds directly to the configured port number. Hence, the devices must be configured as follows: ps0 configured at port one, ps1 at port two, ps2 at port three, and ps3 at port four. This is true regardless to the number of adapters actually present. If a system has only one 4216 adapter, it should be configured as ps0. Automatic configuration will result in the card being configured at port four.

- For both the IBM RT PC and the IBM 6152 the following steps are required for setting up the 4216 Personal Pageprinter. The preceding steps for the IBM 6152 must be done first, if you are installing onto an IBM 6152.

(1) Loading the Postscript interpreter and fonts from the Personal Pageprinter program disks. This step is unnescessary if these files are already installed. Otherwise, run the shell script /usr/lib/pp4216/installps. This is done by the command sequence:

```
# cd /usr/lib/pp4216 <Enter>
# installps <Enter>
```

(2) Uncommenting the printcap entry or entries for the PostScript printer, lps, or adding new entries by copying the lps entry, for additional printers.

The following is an example of a commented printcap:

```
#
# sample for IBM 4216 Personal Pageprinter - local
#
#lps|local 4216 Personal Pageprinter:
#       :lp=/dev/ps0:sd=/usr/spool/lps:
#       :lf=/usr/adm/ps-errs:
#       :af=/usr/adm/acct-ps:
#       :sh:
#       :SF:
#       :if=/usr/lib/pp4216/pstext:
#       :vf=/usr/lib/pp4216/pps:
#       :pl#66:pw#80:
```

The following is an example of an uncommented printcap:

```
#
# sample for IBM 4216 Personal Pageprinter - local
#
```

```
lps|local 4216 Personal Pageprinter:
        :lp=/dev/ps0:sd=/usr/spool/lps:
        :lf=/usr/adm/ps-errs:
        :af=/usr/adm/acct-ps:
        :sh:
        :SF:
        :if=/usr/lib/pp4216/pstext:
        :vf=/usr/lib/pp4216/pps:
        :pl#66:pw#80:.fi
```

Note: As shipped, the system will support a single ps device.
If you want to use more than one 4216 printer on a single machine, see the PS(4) man page for the appropriate kernel config file entries.
Only one 4216 printer can be installed on the RT.

4.9. Setting Up the Mail System

The mail system consists of the following commands:

/bin/mail	old standard mail program, *binmail*(1)
/usr/ucb/mail	UCB mail program, described in *mail*(1)
/usr/lib/sendmail	mail routing program
/usr/spool/mail	mail spooling directory
/usr/spool/secretmail	secure mail directory
/usr/bin/xsend	secure mail sender
/usr/bin/xget	secure mail receiver
/usr/lib/aliases	mail forwarding information
/usr/ucb/newaliases	command to rebuild binary forwarding data base
/usr/ucb/biff	mail notification enabler
/etc/comsat	mail notification daemon

Normally, you use the *mail*(1) command to send and receive mail. This command provides a front end to edit the messages sent and received, and passes the messages to *sendmail*(8) for routing. To process each piece of mail, the routing algorithm uses knowledge of the network name syntax, aliasing and forwarding information, and network topology, as defined in the configuration file /usr/lib/sendmail.cf. The program /bin/mail delivers local mail by adding it to the mailboxes in the directory /usr/spool/mail/*username*, using a locking protocol to avoid problems with simultaneous updates. After mail is delivered, the local mail delivery daemon /etc/comsat is notified, which in turn notifies users who have issued a "biff y" command that mail has arrived.

Normally, mail queued in the directory /usr/spool/mail can be read only by the recipient. To send mail that is secure against any possible perusal (except by a code-breaker), you should use the secret mail facility, which encrypts the mail so that no one can read it.

To set up the mail facility, read the instructions in the file READ_ME in the directory /usr/src/usr.lib/sendmail. Then adjust the necessary configuration files. You should also set up the file /usr/lib/aliases for your installation, creating mail groups as appropriate. Documents describing *sendmail*'s operation and installation are also included in the distribution.

4.9.1. Setting Up a Uucp Connection

The version of *uucp* included in IBM/4.3 is an enhanced version of that originally distri-

buted with 32/V.[4] The enhancements include:

- support for many auto call units
- breakup of the spooling area into multiple subdirectories
- addition of an *L.cmds* file to control the set of commands that may be executed by a remote site
- enhanced "expect-send" sequence capabilities when logging in to a remote site
- new commands used to poll sites and obtain snapshots of *uucp* activity
- additional protocols for different communication media

This section gives a brief overview of *uucp* and points out the most important steps in its installation.

To connect two IBM/4.3 machines with a *uucp* network link using modems, one site must have an automatic call unit and the other must have a dialup port. It is best if both sites have both.

You should first read the article "Uucp Implementation Description" in the *UNIX System Manager's Manual.* It describes in detail the file formats and conventions, and will give you a little context. In addition, the document setup.tblms, located in the directory /usr/src/usr.bin/uucp/UUAIDS, may be of use in tailoring the software to your needs.

The *uucp* support is located in three major directories: /usr/bin, /usr/lib/uucp, and /usr/spool/uucp. User commands are kept in /usr/bin; operational commands are in /usr/lib/uucp; and /usr/spool/uucp is used as a spooling area. The commands in /usr/bin are:

/usr/bin/uucp	file copy command
/usr/bin/uux	remote execution command
/usr/bin/uusend	binary file transfer using mail
/usr/bin/uuencode	binary file encoder (for *uusend*)
/usr/bin/uudecode	binary file decoder (for *uusend*)
/usr/bin/uulog	scans session log files
/usr/bin/uusnap	gives a snapshot of *uucp* activity
/usr/bin/uupoll	polls remote system until an answer is received
/usr/bin/uuname	prints a list of known uucp hosts
/usr/bin/uuq	gives information about the queue

The important files and commands in /usr/lib/uucp are:

/usr/lib/uucp/L-devices	list of dialers and hardwired lines
/usr/lib/uucp/L-dialcodes	dialcode abbreviations
/usr/lib/uucp/L.aliases	hostname aliases
/usr/lib/uucp/L.cmds	commands remote sites may execute
/usr/lib/uucp/L.sys	systems to communicate with, how to connect, and when
/usr/lib/uucp/SEQF	sequence numbering control file
/usr/lib/uucp/USERFILE	remote site pathname access specifications
/usr/lib/uucp/uucico	*uucp* protocol daemon
/usr/lib/uucp/uuclean	cleans up garbage files in spool area
/usr/lib/uucp/uuxqt	*uucp* remote execution server

[4]The *uucp* included in this distribution is the result of work by many people; we gratefully acknowledge their contributions, but refrain from mentioning names in the interest of keeping this document current.

while the spooling area contains the following important files and directories:

/usr/spool/uucp/C.	directory for command (C.) files
/usr/spool/uucp/D.	directory for data (D.) files
/usr/spool/uucp/X.	directory for command execution (X.) files
/usr/spool/uucp/D.*machine*	directory for local D. files
/usr/spool/uucp/D.*machine*X	directory for local X. files
/usr/spool/uucp/TM.	directory for temporary (TM.) files
/usr/spool/uucp/LOGFILE	log file of *uucp* activity
/usr/spool/uucp/SYSLOG	log file of *uucp* file transfers

To install *uucp* on your system, start by selecting a site name (shorter than 14 characters). Next, create a *uucp* account in the /etc/passwd file and set up a password. Then, create the appropriate spooling directories with mode 755 and owned by user *uucp*, group *daemon*.

If you have an auto-call unit, create the L.sys, L-dialcodes, and L-devices files. The L.sys file should contain the phone numbers and login sequences required to establish a connection with a *uucp* daemon on another machine. For example, the L.sys file looks something like:

```
ibmsupt Any ACU 1200 out0123456789- ogin-EOT-ogin uucp
cbosg Never Slave 300
cbosgd Never Slave 300
chico Never Slave 1200 out2010123456
```

The first field is the name of a site; the second tells when the machine may be called; the third specifies how the host is connected (through an ACU, a hardwired line, etc.); the fourth is the baud rate; the fifth is the phone number to use in connecting through an auto-call unit; and the sixth is a login sequence. The phone number may contain common abbreviations that are defined in the L-dialcodes file. The device specification should refer to devices found in the L-devices file. Using only ACU causes the *uucp* daemon, *uucico*, to search for any available auto-call unit in L-devices. Berkeley's L-dialcodes file is of the form:

```
ucb 2
out 9%
```

while their L-devices file is:

```
ACU tty00 unused 1200 ventel
```

Refer to the README file in the *uucp* source directory for more information about installation.

As *uucp* operates, it creates (and removes) many small files in the directories underneath /usr/spool/uucp. Sometimes files are left undeleted; purge them with the *uuclean* program. The log files can grow without bound unless trimmed back; use *uulog* to maintain these files. Many useful aids in maintaining your *uucp* installation are included in a subdirectory UUAIDS beneath /usr/src/usr.bin/uucp. Peruse this directory, and read the "setup" instructions also located there.

The *tailor*(8) command may be used to set up the necessary uucp entries for connecting a uucp-connection work station to the IBM support network.

5. NETWORK SETUP

IBM/4.3 provides support for the DARPA standard Internet protocols IP, ICMP, TCP, and UDP. These protocols may be used on top of a variety of hardware devices ranging from the IMP's (PSN's) used in the ARPANET to local area network controllers for the Ethernet or IBM Token-Ring local area network. Network services are split between the kernel (communication protocols) and user programs (user services such as TELNET and FTP). This section describes how to configure your system to use the Internet networking support. IBM/4.3 also supports the Xerox Network Systems (NS) protocols. IDP and SPP are implemented in the kernel, and other protocols such as Courier run at the user level.

5.1. System Configuration

To configure the kernel to include the Internet communication protocols, define the INET option. Xerox NS support is enabled with the NS option. In either case, include the pseudo-devices "pty", and "loop" in your machine's configuration file. The "pty" pseudo-device forces the pseudo terminal device driver to be configured into the system, see *pty*(4), while the "loop" pseudo-device forces inclusion of the software loopback interface driver. The loop driver is used in network testing and also by the error logging system.

If you are planning to use the Internet network facilities on a 10 MB/s Ethernet or on the IBM Token-Ring local area network, the pseudo-device "ether" should also be included in the configuration; this forces inclusion of the Address Resolution Protocol module used in mapping between 48-bit Ethernet and 32-bit Internet addresses. Also, if you have an IMP connection, you will need to include the pseudo-device "imp."

Before configuring the appropriate networking hardware, you should consult the manual pages in section 4 of the *UNIX Programmer's Manual.* Software support exists for the device "un," the IBM Baseband Adapter for use with the Ethernet network, and for the device "lan," the IBM Token-Ring local area network interface.

All network interface drivers including the loopback interface, require that their host address(es) be defined at boot time. This is done with *ifconfig*(8C) commands included in the /etc/rc.local file. Interfaces that are able to dynamically deduce the host part of an address may check that the host part of the address is correct. The manual page for each network interface describes the method used to establish a host's address. *Ifconfig*(8) can also be used to set options for the interface at boot time. Options are set independently for each interface, and apply to all packets sent using that interface. These options include disabling the use of the Address Resolution Protocol; this may be useful if a network is shared with hosts running software that does not yet provide this function. Alternatively, translations for such hosts may be set in advance or "published" by a IBM/4.3 host by use of the *arp*(8c) command. Note that the use of trailer link-level is now negotiated between IBM/4.3 hosts using ARP, and it is thus no longer necessary to disable the use of trailers with *ifconfig*.

To use the pseudo terminals just configured, device entries must be created in the /dev directory. To create 32 pseudo terminals (plenty, unless you have a heavy network load) execute the following commands.

```
# cd /dev <Enter>
# ./MAKEDEV pty0 pty1 <Enter>
```

More pseudo terminals may be made by specifying *pty2*, *pty3*, etc. The kernel normally includes support for 32 pseudo terminals unless the configuration file specifies a different number. Each pseudo terminal really consists of two files in /dev: a master and a slave. The master pseudo terminal file is named /dev/ptyp?, while the slave side is /dev/ttyp?.

Pseudo terminals are also used by several programs not related to the network. In addition to creating the pseudo terminals, be sure to install them in the /etc/ttys file (with a 'none' in the second column so no *getty* is started).

5.2. Local Subnetworks

In IBM/4.3 the DARPA Internet support includes the notion of "subnetworks." This is a mechanism by which multiple local networks may appear as a single Internet network to off-site hosts. Subnetworks are useful because they allow a site to hide their local topology, requiring only a single route in external gateways; it also means that local network numbers may be locally administered. The standard describing this change in Internet addressing is RFC-950.

To set up local subnetworks one must first decide how the available address space (the Internet "host part" of the 32-bit address) is to be partitioned. Sites with a class A network number have a 24-bit address space with which to work, sites with a class B network number have a 16-bit address space, while sites with a class C network number have an 8-bit address space[5]. To define local subnets you must steal some bits from the local host address space for use in extending the network portion of the Internet address. This reinterpretation of Internet addresses is done only for local networks; i.e. it is not visible to hosts off-site. For example, if your site has a class B network number, hosts on this network have an Internet address that contains the network number, 16 bits, and the host number, another 16 bits. To define 254 local subnets, each possessing at most 255 hosts, 8 bits may be taken from the local part. (The use of subnets 0 and all-1's, 255 in this example, is discouraged to avoid confusion about broadcast addresses.) These new network numbers are then constructed by concatenating the original 16-bit network number with the extra 8 bits containing the local subnetwork number.

The existence of local subnetworks is communicated to the system at the time a network interface is configured with the *netmask* option to the *ifconfig* program. A "network mask" is specified to define the portion of the Internet address that is to be considered the network part for that network. This mask normally contains the bits corresponding to the standard network part as well as the portion of the local part that has been assigned to subnets. If no mask is specified when the address is set, it will be set according to the class of the network. For example, **at Berkeley** (class B network 128.32) 8 bits of the local part have been reserved for defining subnetworks; consequently the /etc/rc.local file contains lines of the form

```
/etc/ifconfig en0 netmask 0xffffff00 128.32.1.7
```

This specifies that for interface "en0," the upper 24 bits of the Internet address should be used in calculating network numbers (netmask 0xffffff00), and the interface's Internet address is "128.32.1.7" (host 7 on network 128.32.1). Hosts *m* on sub-network *n* of this network would then have addresses of the form "128.32.*n.m*;" for example, host 99 on network 129 would have an address "128.32.129.99." For hosts with multiple interfaces, the network mask should be set for each interface, although in practice only the mask of the first interface on each network is actually used.

5.3. Internet Broadcast Addresses

The address defined as the broadcast address for Internet networks according to RFC-919 is the address with a host part of all 1's. The address used by 4.2/RT was the address with

[5] If you are unfamiliar with the Internet addressing structure, consult "Address Mappings", Internet RFC-796, J. Postel; available from the Internet Network Information Center at SRI.

a host part of 0. IBM/4.3 uses the standard broadcast address (all 1's) by default, but allows the broadcast address to be set (with *ifconfig*) for each interface. This allows networks consisting of both 4.2/RT and IBM/4.3 hosts to coexist while the upgrade process proceeds. In the presence of subnets, the broadcast address uses the subnet field as for normal host addresses, with the remaining host part set to 1's (or 0's, on a network that has not yet been converted). IBM/4.3 hosts recognize and accept packets sent to the logical-network broadcast address as well as those sent to the subnet broadcast address, and when using an all-1's broadcast, also recognize and receive packets sent to host 0 as a broadcast.

5.4. Routing

If your environment allows access to networks not directly attached to your host you will need to set up routing information to allow packets to be properly routed. Two schemes are supported by the system. The first scheme employs the routing table management daemon /etc/routed to maintain the system routing tables. The routing daemon uses a variant of the Xerox Routing Information Protocol to maintain up-to-date routing tables in a cluster of local area networks. By using the /etc/gateways file created by *htable*(8), the routing daemon can also be used to initialize static routes to distant networks (see the next section for further discussion). When the routing daemon is started up (usually from /etc/rc.local) it reads /etc/gateways if it exists and installs those routes defined there, then broadcasts on each local network to which the host is attached to find other instances of the routing daemon. If any responses are received, the routing daemons cooperate in maintaining a globally consistent view of routing in the local environment. This view can be extended to include remote sites also running the routing daemon by setting up suitable entries in /etc/gateways; consult *routed*(8C) for a more thorough discussion.

The second approach is to define a default or wildcard route to a smart gateway and depend on the gateway to provide ICMP routing redirect information to dynamically create a routing data base. This is done by adding an entry of the form

/etc/route add default *smart-gateway* 1

to /etc/rc.local; see *route*(8C) for more information. The default route will be used by the system as a "last resort" in routing packets to their destination. Assuming the gateway to which packets are directed is able to generate the proper routing redirect messages, the system will then add routing table entries based on the information supplied. This approach has certain advantages over the routing daemon, but is unsuitable in an environment where there are only bridges (i.e. pseudo gateways that, for instance, do not generate routing redirect messages). Further, if the smart gateway goes down there is no alternative, save manual alteration of the routing table entry, to maintaining service.

The system always listens, and processes, routing redirect information, so it is possible to combine both of the above facilities. For example, the routing table management process might be used to maintain up-to-date information about routes to geographically local networks, while employing the wildcard routing techniques for "distant" networks. The *netstat*(1) program may be used to display routing table contents as well as various routing oriented statistics. For example,

#netstat -r <Enter>

will display the contents of the routing tables, while

#netstat -r -s <Enter>

will show the number of routing table entries dynamically created as a result of routing redirect messages, etc.

5.5. Use of IBM/4.3 Machines as Gateways

Several changes have been made in IBM/4.3 in the area of gateway support (or packet forwarding, if one prefers). A new configuration option, GATEWAY, is used when configuring a machine to be used as a gateway. This option increases the size of the routing hash tables in the kernel. Unless configured with that option, hosts with only a single non-loopback interface never attempt to forward packets or to respond with ICMP error messages to misdirected packets. This change reduces the problems that may occur when different hosts on a network disagree as to the network number or broadcast address. Another change is that IBM/4.3 machines that forward packets back through the same interface on which they arrived will send ICMP redirects to the source host if it is on the same network. This improves the interaction of IBM/4.3 gateways with hosts that configure their routes via default gateways and redirects. The generation of redirects may be disabled with the configuration option IPSENDREDIRECTS=0 in environments where it may cause difficulties.

Local area routing within a group of interconnected Ethernets and other such networks may be handled by *routed*(8c). Gateways between the ARPANET or Milnet and one or more local networks require an additional routing protocol, the Exterior Gateway Protocol (EGP), to inform the core gateways of their presence and to acquire routing information from the core. An EGP implementation for 4.2BSD was done by Paul Kirton while visiting ISI, and any sites requiring such support that have not already obtained a copy should contact Joyce Reynolds (JKReynolds@usc-isif.arpa) for information. That implementation works with IBM/4.3 without kernel modifications. The Kirton implementation must be modified, as packets from the ICMP raw socket include the IP header like other raw sockets in IBM/4.3. If necessary, contact the Berkeley Computer Systems Research Group for assistance.

5.6. Network Servers

In IBM/4.3 most of the server programs are started up by a "super server," the Internet daemon. The Internet daemon, /etc/inetd, acts as a master server for programs specified in its configuration file, /etc/inetd.conf, listening for service requests for these servers, and starting up the appropriate program whenever a request is received. The configuration file contains lines containing a service name (as found in /etc/services), the type of socket the server expects (e.g. stream or dgram), the protocol to be used with the socket (as found in /etc/protocols), whether to wait for each server to complete before starting up another, the user name as which the server should run, the server program's name, and at most five arguments to pass to the server program. Some trivial services are implemented internally in *inetd*, and their servers are listed as "internal." For example, an entry for the file transfer protocol server would appear as

```
ftp     stream  tcp     nowait  root    /etc/ftpd       ftpd
```

Consult *inetd*(8c) for more detail on the format of the configuration file and the operation of the Internet daemon.

5.7. Network Data Bases

Several data files are used by the network library routines and server programs. Most of these files are host independent and updated only rarely.

File	Manual reference	Use
/etc/hosts	*hosts*(5)	host names

/etc/networks	*networks*(5)	network names
/etc/services	*services*(5)	list of known services
/etc/protocols	*protocols*(5)	protocol names
/etc/hosts.equiv	*rshd*(8C)	list of "trusted" hosts
/etc/rc.local	*rc*(8)	command script for starting servers
/etc/ftpusers	*ftpd*(8C)	list of "unwelcome" ftp users
/etc/hosts.lpd	*lpd*(8C)	list of hosts allowed to access printers
/etc/inetd.conf	*inetd*(8)	list of servers started by *inetd*

The files distributed are set up for ARPANET or other Internet hosts. Local networks and hosts should be added to describe the local configuration; the Berkeley entries may serve as examples (see also the next section). Network numbers will have to be chosen for each Ethernet and for each Token-Ring network. For sites not connected to the Internet, these can be chosen more or less arbitrarily, otherwise the normal channels should be used for allocation of network numbers.

5.7.1. Regenerating /etc/hosts and /etc/networks

When using the host address routines that use the Internet name server, the file /etc/hosts is only used for setting interface addresses and at other times that the server is not running, and therefore it need only contain addresses for local hosts. There is no equivalent service for network names yet. The full host and network name data bases are normally derived from a file retrieved from the Internet Network Information Center at SRI. To do this you should use the program /etc/gettable to retrieve the NIC host data base, and the program *htable*(8) to convert it to the format used by the libraries. You should change to the directory where you maintain your local additions to the host table and execute the following commands.

```
# /etc/gettable sri-nic.arpa <Enter>
Connection to sri-nic.arpa opened.
Host table received.
Connection to sri-nic.arpa closed.
# /etc/htable hosts.txt <Enter>
Warning, no localgateways file.
#
```

The *htable* program generates three files in the local directory: hosts, networks, and gateways. If a file "localhosts" is present in the working directory its contents are first copied to the output file. Similarly, a "localnetworks" file may be prepended to the output created by *htable*, and 'localgateways" will be prepended to gateways. It is usually wise to run *diff*(1) on the new host and network data bases before installing them in /etc. If you are using the host table for host name and address mapping, you should run *mkhosts*(8) after installing /etc/hosts. If you are using the name server for the host name and address mapping, you only need to install networks and a small copy of hosts describing your local machines. The full host table in this case might be placed somewhere else for reference by users. The gateways file may be installed in /etc/gateways if you use *routed*(8c) for local routing and wish to have static external routes installed when *routed* is started. This procedure is essentially obsolete, however, except for individual hosts that are on the Arpanet or Milnet and do not forward packets from a local network. Other situations require the use of an EGP server.

If you are connected to the DARPA Internet, it is highly recommended that you use the name server for your host name and address mapping, as this provides access to a much larger set of hosts than are provided in the host table. Many large organization on the network, currently have only a small percentage of their hosts listed in the host

table retrieved from NIC.

5.7.2. /etc/hosts.equiv

The remote login and shell servers use an authentication scheme based on trusted hosts. The hosts.equiv file contains a list of hosts that are considered trusted and, under a single administrative control. When a user contacts a remote login or shell server requesting service, the client process passes the user's name and the official name of the host on which the client is located. In the simple case, if the host's name is located in hosts.equiv and the user has an account on the server's machine, then service is rendered (i.e. the user is allowed to log in, or the command is executed). Users may expand this "equivalence" of machines by installing a .rhosts file in their login directory. The root login is handled specially, bypassing the hosts.equiv file, and using only the /.rhosts file.

Thus, to create a class of equivalent machines, the hosts.equiv file should contain the official names for those machines. If you are running the name server, you may omit the domain part of the host name for machines in your local domain. For example, several machines on Berkeley's local network are considered trusted, so the hosts.equiv file is of the form:

```
ucbarpa
calder
dali
ernie
kim
matisse
monet
ucbvax
miro
degas
```

5.7.3. /etc/rc.local

Most network servers are automatically started up at boot time by the command file /etc/rc (if they are installed in their presumed locations) or by the Internet daemon (see above). These include the following:

Program	Server	Started by
/etc/rshd	shell server	inetd
/etc/rexecd	exec server	inetd
/etc/rlogind	login server	inetd
/etc/telnetd	TELNET server	inetd
/etc/ftpd	FTP server	inetd
/etc/fingerd	Finger server	inetd
/etc/tftpd	TFTP server	inetd
/etc/rwhod	system status daemon	/etc/rc
/etc/syslogd	error logging server	/etc/rc.local
/usr/lib/sendmail	SMTP server	/etc/rc.local
/etc/routed	routing table management daemon	/etc/rc.local
/etc/landump	IBM Token-Ring diagnostic daemon	/etc/rc.local

Consult the manual pages and accompanying documentation (particularly for sendmail) for details about their operation.

There are many configurable options in /etc/rc and /etc/rc.local that can be changed in the /etc/rc.config file. See rc.config(5) for more info.

To have other network servers started up as well, the appropriate line should be added to the Internet daemon's configuration file /etc/inetd.conf, or commands of the following sort should be placed in the site-dependent file /etc/rc.local.

```
if [ -f /etc/routed ]; then
        /etc/routed & echo -n ' routed'        > /dev/console
fi
```

5.7.4. /etc/ftpusers

The FTP server included in the system provides support for an anonymous FTP account. Because of the inherent security problems with such a facility you should read this section carefully if you consider providing such a service.

An anonymous account is enabled by creating a user *ftp*. When a client uses the anonymous account a *chroot*(2) system call is performed by the server to restrict the client from moving outside that part of the file system where the user ftp home directory is located. Because a *chroot* call is used, certain programs and files used by the server process must be placed in the ftp home directory. Further, one must be sure that all directories and executable images are unwritable. The following directory setup is recommended.

```
# cd ~ftp <Enter>
# chmod 555 .; chown ftp .; chgrp ftp . <Enter>
# mkdir bin etc pub <Enter>
# chown root bin etc <Enter>
# chmod 555 bin etc <Enter>
# chown ftp pub <Enter>
# chmod 777 pub <Enter>
# cd bin <Enter>
# cp /bin/sh /bin/ls . <Enter>
# chmod 111 sh ls <Enter>
# cd ../etc <Enter>
# cp /etc/passwd /etc/group . <Enter>
# chmod 444 passwd group <Enter>
```

When local users wish to place files in the anonymous area, they must be placed in a subdirectory. In the setup here, the directory ~ftp/pub is used.

Another issue to consider is the copy of /etc/passwd placed here. It may be copied by users who use the anonymous account. They may then try to break the passwords of users on your machine for further access. A good choice of users to include in this copy might be root, daemon, uucp, and the ftp user. All passwords here should probably be "*".

Aside from the problems of directory modes and such, the ftp server may provide a loophole for interlopers if certain user accounts are allowed. The file /etc/ftpusers is checked on each connection. If the requested user name is located in the file, the request for service is denied. This file normally has the following names on our systems.

```
uucp
root
```

Accounts with nonstandard shells should be listed in this file. Accounts without passwords need not be listed in this file, the ftp server will not service these users.

6. SYSTEM OPERATION

This section describes some typical IBM/4.3 operations on an IBM RT PC, including:

- Bootstrapping and shutdown
- Diskettes
- LED numbers
- Keyboard
- Screen Status Line
- Checking system and device error logs
- Checking file systems and performing backups
- Moving file systems
- Monitoring system performance
- Recompiling and reinstalling system software
- Making local modifications
- Accounting for connect time and process resources
- Controlling resources
- Network troubleshooting
- Monitoring specific files

Procedures described here are used periodically to reboot the system, analyze error messages from devices, do disk backups, monitor system performance, recompile system software and control local changes.

6.1. Bootstrapping and Shutdown

During a normal reboot, the system checks the disks and comes up in multi-user mode without intervention at the console. To bring the system up in single-user mode, press and hold down **< Ctrl > - < C >** as soon as the system prints the date. This interrupts the boot with only the console terminal active. It is also possible to allow the file system checks to complete and then to return to single-user mode by signaling *fsck* with a QUIT signal (^).

To boot from the console, press and hold down the following keys:

< Ctrl > - < Alt > - < Pause >

The system tries to boot from a diskette, then from the hard disk.

You can also boot in single-user mode by explicitly typing the system name in response to the boot prompt:

: hd(0,0)vmunix < Enter >

To bring the system up to a multi-user mode from single-user mode, press and hold down **< Ctrl > - < D >** on the console. The system executes /etc/rc (a multi-user restart script) and /etc/rc.local, and comes up on the terminals listed as active in the file /etc/ttys. See *init*(8) and *ttys*(5). Note, however, that this does not do a file system check. If the previous shutdown was not clean, you should run "fsck −p" or force a reboot with *reboot*(8) to check the disks.

When the system is in multi-user mode, you can take it to single-user mode with either:

 # kill 1 < Enter >

or the *shutdown*(8) command. The latter is much more polite if there are other users logged in. Either command will kill all processes and give you a shell on the console, as if you had just booted. File systems remain mounted after the system becomes single-user. To change to multi-user mode again, use the following commands:

```
# cd / <Enter>
# /etc/umount -a <Enter>
# <Ctrl>-<D>
```

Note that the file /usr/adm/shutdownlog records each system shutdown, crash, processor halt, and reboot, with its associated cause.

6.2. Diskettes

Diskettes normally load after the LED display reaches "22." If the LED remains at "22" and the diskette drive light flashes on and off, the diskette does not contain a boot record. Be sure you have a bootable diskette in the drive.

6.3. LED Numbers

The LED displays show the following numbers during a boot of a IBM/4.3 system:

- **00,01,03,09,10,14,15,16,17** - part of the internal power-on sequence.
- **22-29** - booting from diskette.
- **22, with diskette drive light flashing** - not an IBM RT PC boot diskette, no diskette in drive, or drive door not closed.
- **94** - kernel stack overflow.
- **96** - unsupported memory configuration.
- **98** - /boot not found on hd0a.
- **99** - key is in locked position.

If no numbers are displayed, the system has been halted by /etc/halt.

When IBM/4.3 is running, the LED display shows the "load average" for the system. A lightly loaded system will display numbers ranging between "00" and "15" (meaning 0.0 and 1.5). A heavily loaded system will display numbers greater than "50" (5.0).

6.4. Keyboard

The following key combination reboots the entire system (causing a warm IPL, similar to <Ctrl>-<Alt>-<Del> on an IBM 6152). Hold each key down in sequence.

<Ctrl>-<Alt>-<Pause>

The following key combination forces the system to panic, creating a coredump after rebooting. Hold each key down in sequence.

<Ctrl>-<Alt>-<Scroll Lock>

Note that you should NOT use this when the kernel is running, except in extreme emergencies. Instead, see "Bootstrapping and Shutdown" above.

Some users may wish to have their keyboards map to an IBM PC keyboard. The single change required is to swap the functions of the Caps Lock and Ctrl keys on the IBM RT PC keyboard. To do this, see *pf*(1) and *kbdemul*(4).

6.5. Screen Status Line

While IBM/4.3 is running, the last line of the screen is used to display keyboard status, such as CAPS for < Caps Lock > key pressed; SHIFT for < Shift > key pressed; ALT for < Alt > key pressed; CTRL for < Ctrl > key pressed; and ACTION for < Action > key pressed.

6.6. Checking System and Device Error Logs

When serious errors occur on peripherals or in the system, the system displays a warning diagnostic on the console. These messages are collected by the system error logging process *syslogd*(8) and written into a system error log file /usr/adm/messages. Less serious errors are sent directly to *syslogd*, which may log them on the console. The error priorities that are logged and the locations to which they are logged are controlled by /etc/syslog.conf. See *syslogd*(8) for details.

Error messages issued by the devices in the system are described with the drivers for the devices in Volume I, Section 4, of this manual. If errors occur suggesting hardware problems, you should contact your hardware support group. You should check the error log file regularly, using the command:

tail −r /usr/adm/messages < Enter >

6.7. Checking File Systems and Performing Backups

You should periodically check all file systems for consistency. Use the *fsck*(1) command weekly in the absence of problems, and always (usually automatically) after a crash. You can use the procedures of *reboot*(8) to put the system in a state where a file system check can be performed manually or automatically.

You should also back up file systems regularly. Use *dump*(8) for both complete and incremental dumps. Berkeley recommends a towers-of-hanoi dump sequence with full dumps taken every month.

Use three sets of dump media (streaming tape): daily, weekly, and monthly. Perform daily dumps circularly on the daily set with sequence '3 2 5 4 7 6 9 8 9 9 . . .' Each weekly is a level 1; daily dump sequence levels restart after each weekly dump. Full dumps are level 0; daily sequence levels also restart after each full dump.

Thus a typical dump sequence would be:

Dump ID	Level Number	Date	Opr	Size
FULL	0	Nov 24, 1984	sy	137K
D1	3	Nov 28, 1984	sy	29K
D2	2	Nov 29, 1984	ac	34K
D3	5	Nov 30, 1984	ac	19K
D4	4	Dec 1, 1984	ac	22K
W1	1	Dec 2, 1984	sy	40K
D5	3	Dec 4, 1984	ac	15K
D6	2	Dec 5, 1984	sy	25K
D7	5	Dec 6, 1984	sy	15K
D8	4	Dec 7, 1984	ac	19K
W2	1	Dec 9, 1984	sy	118K
D9	3	Dec 11, 1984	ac	15K
D10	2	Dec 12, 1984	sy	26K
D1	5	Dec 15, 1984	ac	14K
W3	1	Dec 17, 1984	sy	71K

D2	3	Dec 18, 1984	sy	13K
FULL	0	Dec 22, 1984	ac	135K

Take weekly dumps often enough that daily dumps always fit on one streaming tape, and never get to the sequence of 9's in the daily level numbers.

Operators can execute **/etc/dump w** at login to learn what needs to be dumped (based on the /etc/fstab information). Be sure to create a group "operator" in the file /etc/group so that *dump*(8) can notify logged-in operators when it needs help.

Dumping files by name is best done with *tar*(1) but the amount of data moved is limited to a single tape. If there are enough drives, you can copy entire disks with *dd*(1) using the raw special files and an appropriate blocking factor. The number of sectors per track is usually a good value to use; consult /etc/disktab.

You should also make full dumps of the root file system on a regular schedule. This is especially true on a system with only one disk. If the root file system is damaged by a hardware or software failure, you can rebuild a workable disk by restoring the dump, using the MINIROOT diskette.

Exhaustion of user file space is certain to occur now and then. You can impose disk quotas, or you might use the programs *du*(1), *df*(1), and *quot*(8) combined with messages of the day and personal letters.

6.8. Moving File Systems

If you have a streaming tape, the best way to move a file system is to dump it to tape using *dump*(8), create a new file system using *newfs*(8), and restore the tape using *restore*(8). If you do not have tape, *dump* accepts an argument telling where to put the dump; you might use another disk. Filesystems may also be moved by piping the output of *dump* to *restore*. The restore program uses an "in-place" algorithm that allows file system dumps to be restored without concern for the original size of the file system. Further, portions of a file system may be selectively restored in a manner similar to the tape archive program.

To merge a file system into an existing one, use *tar*(1).

To shrink a file system, dump the original and restore it onto the new file system. To shrink the root file system with only one disk drive, the procedure is more complicated:

(1) Dump the root file system to a remote streaming tape using *rdump*(8).

(2) Bring the system down.

(3) Use the MINIROOT diskette and *restore.tape*(8) to install the new root file system.

(4) Boot normally using the newly-created disk file system.

Note that if you add new disk drivers, you must modify the default disk partition tables in /etc/disktab and add the drivers to the standalone system in /sys/standca.

6.9. Monitoring System Performance

The *systat* program provided with the system is designed to be an aid to monitoring system-wide activity. The default "pigs" mode shows a dynamic "ps." By running *systat* when the system is active you can judge the system activity in several dimensions: job distribution, virtual memory load, paging and swapping activity, device interrupts, and disk and cpu utilization. Ideally, there should be few blocked (b) jobs; little paging or swapping activity; available bandwidth on the disk devices (most single arms peak out at 20-30 tps in practice); and high (above 50%) user cpu utilization (us).

If the system is busy, then the count of active jobs may be large, and several of these jobs may often be blocked (b). If the virtual memory is active, then the paging daemon will be running (sr will be non-zero). It is healthy for the paging daemon to free pages when the virtual memory gets active; it is triggered by the amount of free memory dropping below a threshold and increases its pace as free memory goes to zero.

If you run in the "vmstat" mode when the system is busy, you can find imbalances by noting abnormal job distributions. If many processes are blocked (b), then the disk subsystem is overloaded or imbalanced. If you have several non-dma devices or open teletype lines that are "ringing," or user programs that are doing high-speed non-buffered input/output, then the system time may go high (60-70% or higher). It is often possible to pin down the cause of high system time by seeing if there is excessive context switching (cs) and per-device interrupt counts, interrupt activity (in) or system call activity (sy).

If the system is heavily loaded or if you have little memory for your load (2 megabyte is little in most any case), then the system may be forced to swap. This is likely to be accompanied by a noticeable reduction in system performance and pauses when interactive jobs such as editors swap out. If you expect to be in a memory-poor environment for an extended period, you might consider administratively limiting system load.

6.10. Recompiling and Reinstalling System Software

It is easy to regenerate the system, and it is a good idea to try rebuilding pieces of the system to build confidence in the procedures. The system consists of two major parts: the kernel itself (/sys) and the user programs (/usr/src and subdirectories). The major part of this is /usr/src.

The major library is the C library in /usr/src/lib/libc. The library is remade by changing into the correct directory and typing:

 # make <Enter>

and then installed by typing:

 # make install <Enter>

Similarly, typing:

 # make clean <Enter>

cleans up.

The source for all other libraries is kept in subdirectories of /usr/src/usr.lib; each has a makefile and can be recompiled by the above recipe.

NOTE: The code to support IEEE floating point emulation is distributed only in object form on the system. If the system is rebuilt, be very careful not to delete the /usr/src/usr.lib/libfp/emulfp/*.o modules. (They will not be removed by a *make clean.*) It is strongly recommended that you *tar* these modules to a diskette before starting a rebuild.

If you look at /usr/src/Makefile, you will see that you can recompile the entire system source with one command. To recompile a specific program, find out where the source resides with the *whereis*(1) command, then change to that directory and remake it with the makefile present in the directory. For instance, to recompile "date," all one has to type is:

 # whereis date <Enter>
date: /usr/src/bin/date.c /bin/date /usr/man/man1/date.1 <Enter>
 # cd /usr/src/bin <Enter>
 # make date <Enter>

This will create an unstripped version of the binary of "date" in the current directory. To install the binary image, use the install command:

 # install −s date /bin/date <Enter>

The −s option will insure the installed version of date has its symbol table stripped. The install command should be used instead of mv or cp as it understands how to install programs even when the program is currently in use.

If you wish to recompile and install all programs in a particular target area, you can override the default target by typing:

 # make <Enter>
 # make DESTDIR=*pathname* install <Enter>

To regenerate all the system source you can type:

 # cd /usr/src <Enter>
 # make <Enter>

If you modify the C library (perhaps to change a system call) and want to rebuild and install everything from scratch, you have to be careful. You must insure the libraries are installed before the remainder of the source; otherwise the loaded images will not contain the new routine from the library. The following sequence will accomplish this:

 # cd /usr/src <Enter>
 # make clean <Enter>
 # make build <Enter>
 # make installsrc <Enter>

The first *make* removes any existing binaries in the source trees to ensure that everything is reloaded. The next *make* compiles and installs the libraries and compilers, then compiles the remainder of the sources. The final line installs all of the commands not installed in the first phase.

6.11. Making Local Modifications

To keep track of changes to system source, Berkeley migrates changed versions of commands in /usr/src/bin, /usr/src/usr.bin, and /usr/src/ucb in through the directory /usr/src/new and out of the original directory into /usr/src/old for a time before removing them. (Berkeley also uses /usr/new for the programs that constitute the contributed software portion of the 4.3BSD distribution.) Locally written commands that aren't distributed are kept in /usr/src/local and their binaries are kept in /usr/local. This allows /usr/bin, /usr/ucb, and /bin to correspond to the distribution tape (and to the manuals that people can buy). People wishing to use /usr/local commands are made aware that they aren't in the base manual. As manual updates incorporate these commands, they are moved to /usr/ucb.

A directory /usr/junk to throw garbage into, as well as binary directories /usr/old and /usr/new, are useful. The man command supports manual directories such as /usr/man/mano for old and /usr/man/manl for local to make this or something similar practical.

6.12. Accounting for Connect Time and Process Resources

IBM/4.3 optionally records two kinds of accounting information: connect time accounting and process resource accounting. The connect time accounting information is stored in the file /usr/adm/wtmp, and is summarized by the program *ac*(8). The process time accounting information is stored in the file /usr/adm/acct after it is enabled by *accton*(8), which is

analyzed and summarized by the program *sa*(8).

If you need to charge for computing time, you can implement procedures based on the information provided by these commands. A convenient way to do this is to give commands to the clock daemon /etc/cron to be executed every day at a specified time. This is done by adding lines to /usr/lib/crontab; see *cron*(8) for details.

6.13. Controlling Resources

Resource control in the current version of IBM/4.3 is elaborate compared to most UNIX operating systems. The disk quota facilities developed at the University of Melbourne have been incorporated in the system and allow control over the number of files and amount of disk space each user may use on each file system. In addition, the resources consumed by any single process can be limited by the mechanisms of *setrlimit*(2). As distributed, the latter mechanism is voluntary, though sites may choose to modify the login mechanism to impose limits not covered with disk quotas.

To use the disk quota facilities, the system must be configured with "options QUOTA." Then place file systems under the quota mechanism by creating a null file *quotas* at the root of the file system, running *quotacheck*(8), and modifying /etc/fstab to indicate the file system is read-write with disk quotas (an "rq" type field). Then run the program *quotaon*(8) to enable quotas.

Apply individual quotas using the quota editor *edquota*(8). Users may view their quotas (but not those of other users) with the *quota*(1) program. Use the *repquota*(8) program to summarize the quotas and current space usage on a particular file system or file systems.

You can enforce quotas with *soft* and *hard* limits. When a user first reaches a soft limit on a resource, a message appears on his/her terminal. If the user fails to lower the resource usage below the soft limit, the next *login* causes a warning about excessive usage. Should three login sessions go by with the soft limit breached, the system then treats the soft limit as a *hard* limit and disallows any allocations until enough space is reclaimed to bring the user back below the soft limit. Hard limits are strictly enforced, resulting in errors when a user tries to create or write a file. Each time a hard limit is exceeded the system will generate a message on the user's terminal.

Consult the document "Disc Quotas in a UNIX Environment" in the *UNIX System Manager's Manual* and the related manual pages for more information.

6.14. Network Troubleshooting

If you have anything more than a trivial network configuration, from time to time you are bound to run into problems. Before blaming the software, first check your network connections. On networks such as the Ethernet, a loose cable tap or misplaced power cable can result in severely deteriorated service. The *netstat*(1) program may be of aid in tracking down hardware malfunctions. In particular, look at the −i and −s options in the manual page.

Should you believe a communication protocol problem exists, consult the protocol specifications, and attempt to isolate the problem in a packet trace. The SO_DEBUG option may be supplied before establishing a connection on a socket, in which case the system will trace all traffic and internal actions (such as timers expiring) in a circular trace buffer. This buffer may then be printed out with the *trpt*(8C) program. Most servers distributed with the system accept a −d option that forces all sockets to be created with debugging turned on. Consult the appropriate manual pages for more information.

6.15. Monitoring Specific Files

As part of normal system operations, you should periodically review the following files (some of which are system-specific):

/etc/fstab	how disk partitions are used
/etc/disktab	disk partition sizes
/etc/exports	NFS file systems to export
/etc/printcap	printer data base
/etc/gettytab	terminal type definitions
/etc/remote	names and phone numbers of remote machines for *tip*(1)
/etc/group	group memberships
/etc/motd	message of the day
/etc/passwd	password file; each account has a line
/etc/rc.local	local system restart script; runs reboot; starts daemons
/etc/rc.config	allows customization of /etc/rc and /etc/rc.local
/etc/inetd.conf	local internet servers
/etc/hosts	host name data base
/etc/networks	network name data base
/etc/services	network services data base
/etc/hosts.equiv	hosts under same administrative control
/etc/syslog.conf	error log configuration for *syslog*(8)
/etc/ttys	enables/disables ports
/usr/lib/crontab	commands that are run periodically
/usr/lib/aliases	mail forwarding and distribution groups
/usr/adm/acct	raw process account data
/usr/adm/lpd-errs	line printer daemon error log
/usr/adm/messages	system error log
/usr/adm/ppd-errs	page printer error log
/usr/adm/shutdownlog	log of system reboots
/usr/adm/wtmp	login session accounting

Appendix A. Advanced Interactive Executive (AIX) and IBM/4.3 Co-residence

It is possible to have AIX and IBM/4.3 systems on the same machine. Each system should have its own disk or disks. For the two systems to co-reside, the following steps must be taken (a two-disk system is described; a three-disk system is similar):

1. Installing AIX on an Existing IBM/4.3 System

AIX normally expects to be booted from drive 0, so move the IBM/4.3 system to drive 1 or drive 2. This can be done by copying the file systems (with dump/restore) or by physically moving the disks. To move the disks, see *IBM RT PC User Setup Guide*, SV21-8020.

1.1. Creating a Minidisk (partition) Table on Existing IBM/4.3 Disks

For the AIX installation procedure not to use the IBM/4.3 disk (or disks), a minidisk (partition) table must be established on each IBM/4.3 disk that uses all the available space. This is done by the *minidisk*(8R) utility, part of the *sautil*(8R) standalone utility. This procedure assumes there is no existing IBM/4.3 minidisk (partition) table.

The procedure is:

(1) Boot up the *sautil* utility. It is located in /usr/stand/sautil on an installed system, and on the standalone SAUTIL diskette.

(2) Select the "minidisk" menu item.

(3) Initialize the minidisk directory - this makes all the space on the disk available.

(4) Create the standard partition tables by using the *standard* command (this corresponds to the normal a, b, and g partitions of a IBM/4.3 disk.) It also creates a boot partition to hold the bootstrap, since this exists before the "a" partition.

2. Installing AIX and IBM/4.3 on a New Machine

Install IBM/4.3 first, then AIX as follows:

- Create a IBM/4.3 minidisk (partition) table on each IBM/4.3 disk
- Install the IBM/4.3 root and /usr file systems onto an available disk (AIX uses drive 0 and possibly drive 1; IBM/4.3 uses drives after AIX.) See the section "Changing Installation Options," of the chapter "Installation Procedures," in this article.
- Install AIX -- it should only use the drives set aside for it, leaving the IBM/4.3 drives alone because they have valid minidisk tables using all the available space.

3. Booting AIX

After AIX is installed, it puts its bootstrap into the boot block of drive zero. This means when the system is booted, AIX runs (since the default boot order is f0, f1, d0, d1, d2).

4. Booting IBM/4.3

There are two alternatives here. The simplest solution is to use a IBM/4.3 boot diskette to boot IBM/4.3 from drive 1. This requires manual intervention (or a non-standard boot diskette) since the the standard boot diskette attempts to boot hd(0,0)vmunix. Since the IBM/4.3 root is on drive 1 (or drive 2), you should boot IBM/4.3 from hd(1,0)vmunix or hd(2,0)vmunix as appropriate. A generic kernel then asks for the root disk (which is either hd1 or hd2 depending upon which drive is used for the root).

The other alternative is to change the boot order in non-volatile ram so the boot order is f0, f1, d1, d0, d2. This should be done in those cases where AIX is not used frequently. The "iplsource" option of *sautil*(8R) describes how to do this. In this case, a non-generic kernel must be used. See the section "Building New System Images" in the chapter "System Setup" of this article for information on building kernels. A generic kernel will attempt to use hd0a as its root device and then panic as there is not a proper superblock there.

5. Creating a Minidisk Partition Table on a New IBM/4.3 Disk

There are two reasons for creating a partition table on a disk before the IBM/4.3 installation:

- To prepare for eventual AIX installation (this is not critical as the partition table can always be added later)
- To use non-standard partitions. This is often done because the standard partitions do not fit every situation. In particular, it is often the case on small (40Mb) disks, the swap partition is insufficient to run large applications (such as window managers) and more swap space must be allocated. In other cases, it is desirable to create more partitions, or to have only a swap area and a large filesystem (on a second drive for example) rather than the standard three partitions.

(**WARNING**: changing the size or location of a partition containing a filesystem effectively destroys all the contents of that filesystem. You must do a dump/restore to change a filesystem's size and keep the contents). The procedure for creating a non-standard minidisk (partition) table is:

(1) Boot up *minidisk*(8R) utility (as described earlier).

(2) Initialize the minidisk directory by using the *initialization* command to make all the space available.

(3) Create standard partitions by using the *standard* command.

(4) Delete the partitions not needed (but keep the partition named "boot" as this is required to align the IBM/4.3 minidisks on cylinder boundaries).

(5) Create (or recreate) the new partitions. Usually one makes the 'b' (swap) partition bigger, and the 'g' (usr) partition smaller.

6. Installing IBM/4.3 on an Existing AIX Machine

As AIX automatically uses all the available disks, this is only feasible in two cases:

- A new disk can be added. In this case, create the IBM/4.3 minidisk (partition) table and install IBM/4.3 on the new disk.
- The AIX system must be dumped to tape or diskette. Then IBM/4.3 is installed as described above and AIX is re-installed from the dumped tape or diskette.

7. Shared Swap Partitions

IBM/4.3 can use the AIX swap partition, but a non-generic kernel must be configured to do so. Assume AIX on drive 0, IBM/4.3 on drive 1. This kernel could have a configuration like:

```
config  vmunix          root on hd1 swap on hd0 and hd1
```

8. Generating a Sautil Diskette

A bootable standalone diskette is generated by taking the program (such as */boot*) and writing

it onto a diskette with *doswrite* (see *dosread*(1)) with the appropriate options.

To generate a standalone bootable sautil diskette, do the following:

```
cd /sys/standca <Enter>
make boot.out <Enter>
doswrite -i -b -v boot.out <Enter>
```

Appendix B. MINIROOT Kernel Configured Devices

This appendix lists the configurable devices supported by the MINIROOT kernel. For those adapters with selectable addresses, the supported addresses are listed.

Device	Maximum	Description	Address
Displays:			
apa16	1	IBM 6155 Extended Monochrome Graphics Display	
apa8c	1	IBM 6154 Advanced Color Graphics Display	
apa8	1	IBM 6153 Advanced Monochrome Graphics Display	
aed	1	IBM ACIS experimental display	
mono	1	IBM 5151 PC Monochrome Display	
mpel	1	IBM 5081 Megapel Color Display	
Networks:			
lan	1	IBM Token-Ring Network Adapter	f00001c0
un	1	IBM Baseband Adapter for Ethernet	f4080000 f4088000 f4090000 f4098000
Disks:			
hdc	2	IBM PC/AT, ESDI, or EESDI Controller (hard disk function)	f00001f0 f0000170
hd	3	Hard disk drives	
fdc	1	IBM PC/AT, ESDI, or EESDI Controller (diskette function)	f00003f2 f0000372
fd	2	Diskette drives	
Tape:			
stc	1	Streaming tape controller	f00001e8
st	1	Streaming tape drive	

This page intentionally left blank.

Appendix C. Building a Master/Server Machine

This appendix assumes that the reader has working knowledge of an Andrew File System Server, and systems level knowledge of the UNIX operating system.

This appendix describes how, in a very small installation, you can make a single IBM RT PC function as a master for network installations and a server for both the Andrew File System (AFS) and the Network File System (NFS) for a network of IBM 6152 and IBM RT PC workstations. At a larger installation it makes more sense to set up dedicated server machines, and dedicated master machines for network installations. See Appendix D for instructions on how to do this.

When you have completed these instructions, the RT will contain the following:

- the IBM 4.3 base system root and /usr partitions
- a "minimal root" partition (used during installation of minimal and reduced workstations)
- an Andrew File System "usr" volume (used during the operation of "minimal" and "reduced" configuration workstations)

The root partition will be served as an NFS file system to workstations with minimal configurations. This is not required if you aren't going to set up any minimally configured workstations.

Disk Partitioning

The RT needs the following partitions:

- A standard root (hd0a) partition
- A standard /usr (hd0g) partition
- A 60 Megabyte scratch partition (we used hd1g, which was most of hd1)
- A 2 Megabyte partition for the minimal root partition "/minroot" (we used hd1a)
- A 70 Megabyte partition for the Andrew File System (we used hd2g, which covered an entire 70 MB hard disk). Note that if you wish to load any of the system components like X11 or the Andrew Toolkit, more than 70 Megabytes are required. It is probably a good idea to have at least 200 Megabytes available for AFS partitions in order to be able to serve the entire system.

The minidisk partitions for the root disk are created as part of the procedure for installing the base IBM/4.3 system. A good way to create partitions for the 60 Megabyte scratch partition and the 2 Megabyte minimal root partition is to use the "standard source" command in the minidisk program. Similarly, you can use "standard single" to create the 70 Megabyte AFS partition. This will be described shortly. This appendix uses *hd2* for the single AFS partition. If you use a different disk (a SCSI disk for example), substitute *sc0* for *hd2* everywhere in this appendix.

The remainder of this appendix provides the installation steps required.

1. **Perform a tape installation of IBM/4.3 on the RT.** Follow the instructions for installing an RT from streaming tape, found in Chapter 3 of this article, with one significant change:

   Change the value for "optional system components" from the default to "afs."

   Note that making this change means you will not be loading source. All of the hard disks (except for SCSI disks) should still be formatted, however. Here's how to format an arbitrary

hard disk:

(1) Insert the sautil diskette into the floppy drive and turn on the power. If it is already on, press **<Ctrl>-<Alt>-<Pause>**. (Press and hold the <Ctrl> key and the <Alt> key. Then press the <Pause> key.)

(2) When the Standalone Utility Menu appears, respond to the *Choice?* prompt by typing **2 <Enter>** to select the "format" option. **Note:** If you do not respond to the Standalone Utility Menu within 30 seconds, the system will time out and the boot prompt will appear. If this occurs, press <Ctrl>-C to return to the Standalone Utility Menu.

You will now format a hard disk partition for IBM/4.3.

(3) Respond to the prompt:

Device to format?

by typing:

hd(1,2) <Enter>

This assumes you are formatting *hd1*. If you are formatting *hd2*, then substitute **hd(2,2)** for **hd(1,2)**.

(4) When prompted to verify the format, type **yes <Enter>**.

(5) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burnin is selected, you will be prompted for the "number of passes"; press **<Enter>** to select the default value.

(6) When asked whether to change the default parameters, type **no <Enter>**.

(7) When asked whether to preserve data already on the disk, type **no <Enter>**. If you have an old unix file system already on the disk and you want to preserve it, type **yes <Enter>**.

(8) When asked if the format process should start, type **yes <Enter>**.

For severe burnin, the system will take several hours to complete the format process.

(9) When the format is finished, press **<Enter>** to return to the main menu.

(10) You may at this time format another disk by going back to step 2.

2. Create minidisk partitions for the disks.

(1) Use the /etc/minidisk command to create partitions on all of the disks other than hd0. The following lines describe how to create partitions suitable for installation of the server partitions using hd1 and hd2 (both 70 MB disks). You may have to substitute other names for some of the disks if you decide to use different disk or partition names. Here's how to create partitions hd1a and hd1g which can be used for the "minimal root" and scratch partitions. First bring the machine up multi-user and login as root. Then type:

```
# /etc/minidisk /dev/rhd1c <Enter>
> init <Enter>
```

Type "y **<Enter>**" in response to "confirm (all data will be lost)".

```
> standard source <Enter>
> quit <Enter>
```

(2) Here's how to create a single 70 Megabyte partition (hd2g) that spans an entire disk:

```
# /etc/minidisk /dev/rhd2c  <Enter>
> init  <Enter>
```

Type "y <Enter>" in response to "confirm (all data will be lost)".

```
> standard single  <Enter>
> quit  <Enter>
```

(3) Reboot the system so that the new minidisk tables are read by the kernel.

3. Edit the /etc/hosts file.

(1) Insert the hostname "vice1" *before* the hostname "master," thus:

```
46.0.0.1  vice1  master
```

(2) Run the tailor command, and change the hostname to vice1. (See *tailor*(8).)

4. Convert the RT into an AFS server and client.

(1) Login as root and type the root password.

(2) Type the following command to initialize the AFS partition:

```
# newfs  /dev/hd2g  <Enter>
```

Add the following line to the end of /etc/fstab:

```
/dev/hd2g /vicepa ufs rw 1 2
```

(3) Reboot the machine, and bring it up single-user. To do this, type:

```
# reboot  <Enter>
```

When the boot prompt appears, type:

```
hd(0,0)vmunix  <Enter>
```

If a "root device?" prompt appears, type:

```
hd0  <Enter>
```

(4) Type the following command to adjust the minfree parameter on the /usr partition to 0 percent. This allows a reasonably-sized AFS cache and the AFS server binaries to both fit in a normal-sized /usr partition:

```
# tunefs -m  0  /dev/rhd0g  <Enter>
```

(5) Type the following commands to mount all of the file systems:

```
# cp  /dev/null  /etc/mtab  <Enter>
# mount  -vf  /  <Enter>
# mount  -atv  ufs  <Enter>
```

Ignore the error message you get that results from a missing /vicepa directory. That directory will be created when contents of the IBM Andrew File System tape are loaded onto the hard disk.

(6) Insert the Andrew File System (AFS) tape into the streaming tape drive.

(7) Issue the following commands to extract the server binaries from the tape:

```
# cd  /
# /usr/ibm/ftc  /dev/rst0  -  |  tar  xvfp  -  <Enter>
```

(8) Edit the /etc/fstab file and add the following line to it:

vice1 /andrew afs rw 0 0

(9) Type the following commands to add the admin account to your password data base:

 # cat /etc/passwd.adm > > /etc/passwd <Enter>
 # vipw <Enter>

This will start the *vi*(1) editor, allowing you to edit the password file.

(10) Simply exit by typing:

:q! <Enter>

The password data base will be updated automatically.

(11) Type the following commands to initialize the Andrew File System user data base:

 # cd /vice/db <Enter>
 # /vice/bin/pwd2pdb -p /etc/passwd -g /usr/admin/groups > vice.pdb <Enter>
 # /vice/bin/pcfgen vice.pdb <Enter>
 # echo 00000000 > /vice/vol/maxvolid <Enter>

Note that in the previous command, it is important that the string of zeroes contain eight (8) zeroes.

(12) Reboot the machine again; when it comes up the AFS server and client code will be running. Type:

 # reboot <Enter>

5. Create the root volume.

(1) Login as root and type the following command:

 # /vice/bin/vol-create /vicepa root <Enter>

Note that vol-create is used to create only the first AFS volume. Afterwards the createvol command is used.

(2) Build the volume data base with the following command:

 # /vice/bin/bldvldb.sh <Enter>

(3) Reboot the machine to force the client to find the new root volume. Type:

 # reboot

6. Create andrew and usr volumes.

(1) Login as root and type the following commands:

 # /vice/bin/createvol andrew vice1 /vicepa <Enter>
 # /vice/bin/createvol usr vice1 /vicepa <Enter>

(2) Type the following commands to give the admin account full privileges to the andrew and usr volumes, which will have no quotas:

 # login admin <Enter>
Password: **install <Enter>**
% cd /andrew <Enter>
% /usr/andrew/bin/fs sa . admin all <Enter>
% /usr/andrew/bin/fs mkmount andrew andrew <Enter>
% /usr/andrew/bin/fs sv andrew -a 0 <Enter>
% /usr/andrew/bin/fs sa andrew admin all <Enter>

```
% /usr/andrew/bin/fs mkmount usr usr <Enter>
% /usr/andrew/bin/fs sv usr -a 0 <Enter>
% /usr/andrew/bin/fs sa usr admin all <Enter>
```

If after making the mount points (with *fs mkmount*), you get errors trying to change the access list (with *fs sa*), and the quota (with *fs sv*), and the mount points look like symbolic links, or are not there, then something went wrong with the actual volume creation on the server. See the Andrew File System documentation for hints on debugging techniques.

7. Load the andrew volume.

(1) Type the following commands:

```
% /usr/andrew/bin/loadafs /usr/andrew /andrew "" <Enter>
```

If you changed your superuser password from what was in the original password file (a null password), then you will have to type it instead of the empty double quotes.

(2) Become the superuser again with the **su** command.

(3) Type the following commands:

```
# rm -rf /usr/andrew <Enter>
# ln -s /andrew/andrew /usr/andrew <Enter>
```

(4) You may at this point load the Andrew Toolkit binaries into the Andrew Volume. Since the Andrew Toolkit is rather large, it requires that you have more than 70 Megabytes of disk space to serve AFS from. If you are not going to load the IBM Andrew Toolkit tape, skip the next two steps.

(5) Insert the IBM Andrew Toolkit tape in the streaming tape drive.

(6) Type the following commands:

```
# newfs /dev/hd1g <Enter>
# mount /dev/hd1g /mnt <Enter>
# mkdir /mnt/usr /mnt/usr/andrew <Enter>
# cd /mnt/usr/andrew <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - X11fonts bin dlib doc
                             etc fonts help include lib man <Enter>
# exit <Enter>
% /usr/andrew/bin/loadafs /mnt/usr/andrew /andrew "" <Enter>
% cd / <Enter>
% su <Enter>
# umount /dev/hd1g <Enter>
```

Again, if you have changed your superuser password from what was in the original password file (a null password), then you will have to type it instead of the empty double quotes.

8. Load the usr volume.

(1) Type the following commands:

```
# newfs /dev/hd1g <Enter>
# mount /dev/hd1g /mnt <Enter>
# mkdir /mnt/usr <Enter>
# cd /mnt/usr <Enter>
```

(2) Insert the ROOT/USER tape into the streaming tape drive.

(3) Type the following commands:

```
# mt -f /dev/rst0 rewind <Enter>
# mt -f /dev/nrst0 fsf <Enter>
# /usr/ibm/ftc /dev/rst0 - | restore rvf - <Enter>
# rm restoresymtable
# ln -s /andrew/andrew /mnt/usr/andrew <Enter>
# rmdir src <Enter>
# ln -s /andrew/src src <Enter>
# cd /mnt <Enter>
```

(4) Follow the instructions in Chapter 3 for installing the other tapes and diskettes (if you have them) with one exception: prepend a "/mnt" to all pathnames that you cd to before loading tapes. This applies to the following tapes and diskettes:

- the X Version 11 tape (only the binaries should be loaded at this point).
- the 6152 Academic System 5-1/4 inch diskette (only if you are going to net-install to 6152 workstations.)
- the Professional Pascal diskette

(5) If you loaded the 6152 Academic System 5-1/4 inch diskette in the previous step, type the following commands:

```
# cd /mnt <Enter>
# mv vmunix.atr /vmunix.atr <Enter>
```

(6) Type the following commands to install the /mnt filesystem into the usr volume:

```
# exit <Enter>
% /usr/andrew/bin/loadafs /mnt/usr /andrew "" <Enter>
% /sys/dist/make.site <Enter>
```

Again, if you have changed your superuser password from what was in the original password file (a null password), then you will have to type it instead of the empty double quotes.

9. Load source into the Andrew File System.

This section describes how to load the IBM 4.3 source code onto the AFS server. If you do not want to serve source to all of your workstations, or do not have enough disk space, you may skip this section.

9.1. Load utility source.

Login as root and type the following commands:

```
# cd /vice/bin <Enter>
# ./createvol src vice1 /vicepa <Enter>
```

Now logout, and log back in as admin. Respond to the "*Password:*" prompt with "**install <Enter>**". Insert the tape labeled *source* into the tape drive, and type the following commands:

```
% cd /andrew <Enter>
% /usr/andrew/bin/fs mkmount src src <Enter>
% /usr/andrew/bin/fs sa src admin all <Enter>
% /usr/andrew/bin/fs sv src -a 0 <Enter>
% cd src <Enter>
% /usr/ibm/ftc /dev/rst0 - | /etc/restore rvf - <Enter>
```

```
% rm restoresymtable <Enter>
```

9.2. Load Andrew Toolkit Source.

If you are not already logged in as admin, then do so. Insert the IBM Andrew Toolkit tape into the tape drive, and type the following commands:

```
% cd /andrew/andrew <Enter>
% /usr/ibm/ftc /dev/rst0 - | tar xfp - be2 makefile example
                            ibmdoc ibmhelp ibmman overhead <Enter>
```

9.3. Load Andrew File System Source.

This section describes loading only the user-level code for the Andrew File System. The kernel code, which is on the IBM Andrew File System tape, is also on the root/usr tape, which has already been loaded into the Andrew File System.

If you are not already logged in as admin, then do so. Insert the IBM Andrew File System tape into the tape drive, and type the following commands:

```
% cd /andrew/andrew <Enter>
% mt -f /dev/nrst0 fsf <Enter>
% /usr/ibm/ftc /dev/rst0 - | tar xfp - server/src <Enter>
```

9.4. Load X11 Source.

Note: This section assumes you have already created and made a mount point for the utility source AFS volume. If you followed the above steps describing how to load utility source into the Andrew File System, then the volume will be have been already set up.

If you are not already logged in as admin, then do so. Insert the X Version 11 tape into the tape drive, and type the following commands:

```
% cd /andrew <Enter>
% /usr/ibm/ftc /dev/rst0 - | tar xfp - src <Enter>
```

10. Make the RT an NFS server.

Note: You may skip this section if you are not going to install any workstations with minimal configurations.

(1) Become the superuser again by typing the su command and supplying the superuser password.

(2) Add the following line to the /etc/exports file (create the file if it is not already there):

```
/ host1 host2 host3 ...
```

The hosts listed after the initial "/" are the names of all of the minimal workstations that are to use this server. You may leave this list null (Just put a "/" in the file) if you don't know all of the hostnames in advance. This makes it possible for any machine on the network to be served the NFS root partition.

11. Create the minimal root partition.

(1) Type the following commands:

```
# newfs /dev/hd1a <Enter>
# mkdir /minroot <Enter>
# mount /dev/hd1a /minroot <Enter>
```

(2) Edit the /etc/fstab file and add the following line:

```
/dev/hd1a /minroot ufs rw 1 3
```

(3) Type the following commands:

```
# cd /sys/dist <Enter>
# ./make.minimal <Enter>
```

This command will take approximately 5 minutes to run. If it does not print any error messages, all went well.

12. Set up symbolic links into AFS

Type the following commands to set up symbolic links in /usr to the AFS usr volume:

```
# cd /usr <Enter>
# rm -rf man doc sys lib/learn dict pub src lib/font <Enter>
# ln -s /andrew/usr/man /andrew/usr/doc /andrew/usr/sys . <Enter>
# ln -s /andrew/usr/dict /andrew/usr/pub . <Enter>
# ln -s /andrew/usr/lib/learn lib/learn <Enter>
# ln -s /andrew/usr/lib/font lib/font <Enter>
# ln -s /andrew/src /usr/src <Enter>
# ln -s /andrew/usr/include/X11 /usr/include <Enter>
# ln -s /andrew/usr/bin/X11 /usr/bin/X11 <Enter>
# ln -s /andrew/usr/lib/libX11.a /usr/lib/libX11.a <Enter>
# ln -s /andrew/usr/lib/libXtk11.a /usr/lib/libXtk11.a <Enter>
# ln -s /andrew/usr/lib/X11 /usr/lib/X11 <Enter>
```

This completes the special installation steps.

Appendix D. Building Separate Master and Server Machines

This appendix assumes that the reader has working knowledge of an Andrew File System Server, and systems level knowledge of the UNIX operating system.

This appendix describes how to set up a single RT to act as an AFS and NFS server for a network of minimal and reduced configuration IBM RT PC's and IBM 6152's. It also describes how to set up a master machine that can be used to install the workstations through the network. A single machine can be set up to do the job of both the server and the master, but that is not recommended except for very small sites where the number of minimal and reduced systems is small.

When you have completed these instructions, the AFS server RT will contain the following:

- the IBM 4.3 base system root and usr partitions
- the Andrew File System server and client code
- an Andrew File System "usr" volume (used during the operation of "minimal" and "reduced" configuration workstations)
- The root file system will be exported as an NFS file system.

The master RT will contain the following:

- the IBM 4.3 base system root, usr, and src partitions.
- the Andrew File System client code
- a "minimal root" partition (used during installation of minimal and reduced workstations)

1. Set up the AFS/NFS server

Disk Partitioning

The RT needs the following partitions:

- A standard root (hd0a) partition
- A standard /usr (hd0g) partition
- A 70 Megabyte partition that will be served by the AFS server (we used hd1g, which covered an entire 70 MB hard disk). Note that if you wish to load any of the system components like X11 or the Andrew Toolkit, more than 70 Megabytes are required. It is probably a good idea to have at least 200 Megabytes available for AFS partitions in order to be able to serve the entire system.

The minidisk partitions for the root disk are created as part of the procedure for installing the base IBM/4.3 system. A good way to create the partition to serve AFS from is to use the "standard single" command in the minidisk program. This will be described shortly. This appendix uses *hd1* for the single AFS partition. If you are using a different disk (a SCSI disk, for example), then substitute *sc0* for *hd1* everywhere in the setup instructions for the server machine.

1.1. Perform a tape installation of IBM/4.3 on the RT.

Follow the instructions for installing an RT from streaming tape, found in Chapter 3 of this article, with one significant change:

Change the value for "optional system components" from the default to "afs."

Note that making this change means you will not be loading source. Unless you are using a SCSI disk for the AFS partition, The second hard disk (hd1) should still be formatted.

After you've installed the base system, power down the machine, and format hd1 with the following procedure:

(1) Insert the sautil diskette into the floppy drive and turn on the power. If it is already on, press **<Ctrl>-<Alt>-<Pause>**. (Press and hold the <Ctrl> key and the <Alt> key. Then press the <Pause> key.)

(2) When the Standalone Utility Menu appears, respond to the *Choice?* prompt by typing **2 <Enter>** to select the "format" option. **Note:** If you do not respond to the Standalone Utility Menu within 30 seconds, the system will time out and the boot prompt will appear. If this occurs, press <Ctrl>-C to return to the Standalone Utility Menu.

You will now format a hard disk partition for IBM/4.3.

(3) Respond to the prompt:

Device to format?

by typing:

hd(1,2) <Enter>

(4) When prompted to verify the format, type **yes <Enter>**.

(5) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burnin is selected, you will be prompted for the "number of passes"; press **<Enter>** to select the default value.

(6) When asked whether to change the default parameters, type **no <Enter>**.

(7) When asked whether to preserve data already on the disk, type **no <Enter>**. If you have an old unix file system already on the disk and you want to preserve it, type **yes <Enter>**.

(8) When asked if the format process should start, type **yes <Enter>**.

For severe burnin, the system will take several hours to complete the format process.

(9) When the format is finished, press **<Enter>** to return to the main menu.

(10) You can bring the machine up to multi-user mode at this point by selecting the *boot* option. Type "**1 <Enter> <Enter>**". **The kernel will be loaded from the hard disk, and the machine will come up.**

1.2. Create minidisk partitions for the disks.

(1) Use the /etc/minidisk command to create partitions on hd1. The following lines describe how to create a partition suitable for serving AFS from. You may have to substitute another name if the disk you decide to use is different. The following steps show how to create a single partition, hd1g, that spans all of hd1. Login as root. Then type:

 # /etc/minidisk /dev/rhd1c <Enter>
> init <Enter>

Type **y <Enter>** in response to *confirm (all data will be lost)*.

> standard single <Enter>
> quit <Enter>

(2) Reboot the system so that the new minidisk tables are read by the kernel.

1.3. Set up the host name and address

(1) If you already have a network of workstations set up at your site, then install your local versions of /etc/hosts and /etc/hosts. equivalent on the system.

(2) If you replaced your host tables, make sure there is still an entry for vice1 in them. If there is not, then create one.

(3) Run the tailor command, and change the hostname to vice1. (See *tailor*(8).)

(4) You may have to make some site specific changes to your rc.config file at this time. For example, if you are using a class B network with subnets, then you will have to supply a netmask to ifconfig via the net_flags variable. Thus:

net_flags = "netmask 0xffffff00"

or something like it will need to be placed in /etc/rc.config.

1.4. Convert the RT into an AFS server and client.

(1) Login as root and type the root password.

(2) Type the following command to initialize the AFS partition:

 # newfs /dev/hd1g <Enter>

Add the following line to the end of /etc/fstab:

/dev/hd1g /vicepa ufs rw 1 2

Reboot the machine, and bring it up single-user. To do this, type:

 # reboot <Enter>

When the boot prompt appears, type:

hd(0,0)vmunix <Enter>

If a "root device?" prompt appears, type:

hd0 <Enter>

(3) Type the following command to adjust the minfree parameter on the /usr partition to 0 percent. This allows a reasonably-sized AFS cache and the AFS server binaries to both fit in a normal-sized /usr partition:

 # tunefs -m 0 /dev/rhd0g <Enter>

(4) Type the following commands to mount all of the file systems:

 # cp /dev/null /etc/mtab <Enter>
 # mount -vf / <Enter>
 # mount -atv ufs <Enter>

Ignore the error message you get that results from a missing /vicepa directory. That directory will be created when contents of the IBM Andrew File System tape are loaded onto the hard disk.

(5) Insert the IBM Andrew File System (AFS) tape into the streaming tape drive.

(6) Issue the following commands to extract the server binaries from the tape and install them on the RT:

 # cd / <Enter>
 # ftc /dev/rst0 - | tar xvfp - <Enter>

(7) Edit the /etc/fstab file and add the following line to it:

vice1 /andrew afs rw 0 0

(8) Type the following commands to add the admin account to your password data base:

 # cat /etc/passwd.adm > > /etc/passwd <Enter>
 # vipw <Enter>

This will start the *vi*(1) editor, allowing you to edit the password file.

(9) Simply exit by typing:

:q! <Enter>

The password data base will be updated automatically.

(10) Type the following commands to initialize the Andrew File System user data base:

 # cd /vice/db <Enter>
 # /vice/bin/pwd2pdb -p /etc/passwd -g /usr/admin/groups > vice.pdb <Enter>
 # /vice/bin/pcfgen vice.pdb <Enter>
 # echo 00000000 > /vice/vol/maxvolid <Enter>

Note that in the previous command, it is important that the string of zeroes contain eight (8) zeroes.

(11) Reboot the machine again; when it comes up the AFS server and client code will be running. Type:

 # reboot <Enter>

1.5. Create the root volume.

(1) Login as root and type the following command:

 # /vice/bin/vol-create /vicepa root <Enter>

Note that vol-create is used to create only the first AFS volume. Afterwards the createvol command is used.

(2) Build the volume data base with the following command:

 # /vice/bin/bldvldb.sh <Enter>

(3) Reboot the machine to force the client to find the new root volume. Type:

 # reboot

1.6. Create andrew and usr volumes.

(1) Login as root and type the following commands:

 # /vice/bin/createvol andrew vice1 /vicepa <Enter>
 # /vice/bin/createvol usr vice1 /vicepa <Enter>

1.7. Make the RT an NFS server.

Note: You may skip this section if you are not going to install any workstations with minimal configurations. Add the following line to the /etc/exports file (create the file if it is not already there):

/ host1 host2 host3 ...

The hosts listed after the initial "/" are the names of all of the minimal workstations that are to use this server. You may leave this list null (Just put a "/" in the file) if you don't

know all of the hostnames in advance. This makes it possible for any machine on the network to be served the NFS root partition.

This completes the server installation procedure.

2. Set up the AFS Client/Master Machine

Disk Partitioning

The RT needs the following partitions:

- A standard root (hd0a) partition
- A standard /usr (hd0g) partition
- A 60 Megabyte scratch partition (we used hd1g, which was most of hd1)
- A 2 Megabyte partition for the minimal root partition "/minroot" (we used hd1a)

The minidisk partitions for the root disk are created as part of the procedure for installing the base IBM/4.3 system. A good way to create partitions for the 60 Megabyte scratch partition is to use the "standard source" command in the minidisk program. This will be described shortly.

2.1. Perform a tape installation of IBM/4.3 on the RT.

Follow the instructions for installing an RT from streaming tape, found in Chapter 3 of this article, with one significant change:

Change the value for "optional system components" from the default ("all") to "afs."

Note that making this change means you will not be loading source. The second hard disk (hd1) should still be formatted however. After you've installed the base system, power down the machine, and format hd1 with the following procedure:

(1) Insert the sautil diskette into the floppy drive and turn on the power. If it is already on, press **<Ctrl>-<Alt>-<Pause>**. (Press and hold the <Ctrl> key and the <Alt> key. Then press the <Pause> key.)

(2) When the Standalone Utility Menu appears, respond to the *Choice?* prompt by typing **2 <Enter>** to select the "format" option. **Note:** If you do not respond to the Standalone Utility Menu within 30 seconds, the system will time out and the boot prompt will appear. If this occurs, press <Ctrl>-C to return to the Standalone Utility Menu.

You will now format a hard disk partition for IBM/4.3.

(3) Respond to the prompt:

Device to format?

by typing:

hd(1,2) <Enter>

(4) When prompted to verify the format, type **yes <Enter>**.

(5) Press **<Enter>** to select the default *severe burnin* format pattern.

The default (severe burnin) pattern is more thorough, but takes more time. If you cannot afford the time for severe burnin, we recommend the zero pattern. Note: If severe burnin is selected, you will be prompted for the "number of passes"; press **<Enter>** to select the default value.

(6) When asked whether to change the default parameters, type **no <Enter>**.

(7) When asked whether to preserve data already on the disk, type **no <Enter>**. If you have an old unix file system already on the disk and you want to preserve it, type **yes <Enter>**.

(8) When asked if the format process should start, type **yes <Enter>**.

For severe burnin, the system will take several hours to complete the format process.

(9) When the format is finished, press **<Enter>** to return to the main menu.

(10) You can bring the machine up to multi-user mode at this point by selecting the *boot* option. Type "**1 <Enter> <Enter>**". The kernel will be loaded from the hard disk, and the machine will come up.

2.2. Create minidisk partitions for the disks.

(1) Use the /etc/minidisk command to create partitions on all of the disks other than hd0. The following lines describe how to create partitions suitable for creation of the 60 Megabyte scratch partition and the 2 Megabyte minimal root partition using hd1 (a 70 MB disk). You may have to substitute another name for hd1 if you decide to use a different disk. Here's how to create partitions hd1a and hd1g which can be used for the "minimal root" and scratch partitions. First login as root. Then type:

```
# /etc/minidisk /dev/rhd1c <Enter>
> init <Enter>
```

Type "y <Enter>" in response to "confirm (all data will be lost)".

```
> standard source <Enter>
> quit <Enter>
```

(2) Reboot the system so that the new minidisk tables are read by the kernel.

2.3. Set up the host name and address

(1) Install your local versions of */etc/passwd, /etc/passwd.dir, /etc/passwd.pag, /etc/group, /etc/hosts, and /etc/hosts.equiv* on the system. The password file must have an entry for *operator*, who must be in the *operator* group. There must also be an entry for *admin* in the password file. *Admin* should be in the *wheel* group.

(2) Create an entry for *master* in the */etc/hosts* file. It must have a network address of *46.0.0.0* or the default network address for network installation of the reduced and minimal workstations later will not work. Also make sure there is an entry for *vice1* in */etc/hosts*.

(3) Run the tailor command, and change the hostname to master. (See *tailor*(8).)

2.4. Convert the RT into an AFS client.

(1) Log in as root, edit the /etc/fstab file, and add the following line to it:

```
vice1 /andrew afs rw 0 0
```

(2) Now mount the Andrew File System by typing:

```
mount -atv afs
```

(3) Type the following commands to give the admin account full privileges to the andrew and usr volumes, which will have no quotas:

```
# login admin <Enter>
```

```
Password: install <Enter>
% cd /andrew <Enter>
% /usr/andrew/bin/fs sa . admin all <Enter>
% /usr/andrew/bin/fs mkmount andrew andrew <Enter>
% /usr/andrew/bin/fs sv andrew -a 0 <Enter>
% /usr/andrew/bin/fs sa andrew admin all <Enter>
% /usr/andrew/bin/fs mkmount usr usr <Enter>
% /usr/andrew/bin/fs sv usr -a 0 <Enter>
% /usr/andrew/bin/fs sa usr admin all <Enter>
```

If after making the mount points (with *fs mkmount*), you get errors trying to change the access list (with *fs sa*), and the quota (with *fs sv*), and the mount points look like symbolic links, or are not there, then something went wrong with the actual volume creation on the server. See the Andrew File System documentation for hints on debugging techniques.

2.5. Load the andrew volume.

(1) Type the following commands:

```
% /usr/andrew/bin/loadafs /usr/andrew /andrew "" <Enter>
```

If you changed your superuser password from what was in the original password file (a null password), then you will have to type it instead of the empty double quotes.

(2) Become the superuser again with the su command.

(3) Type the following commands:

```
# rm -rf /usr/andrew <Enter>
# ln -s /andrew/andrew /usr/andrew <Enter>
```

(4) You may at this point load the Andrew Toolkit binaries into the Andrew Volume. Since the Andrew Toolkit is rather large, it requires that you have more than 70 Megabytes of disk space to serve AFS from. If you are not going to load the IBM Andrew Toolkit tape, skip the next two steps.

(5) Insert the IBM Andrew Toolkit tape in the streaming tape drive.

(6) Type the following commands:

```
# newfs /dev/hd1g <Enter>
# mount /dev/hd1g /mnt <Enter>
# mkdir /mnt/usr /mnt/usr/andrew <Enter>
# cd /mnt/usr/andrew <Enter>
# /usr/ibm/ftc /dev/rst0 - | tar xfp - X11fonts bin dlib doc
                             etc fonts help include lib man <Enter>
# exit <Enter>
% /usr/andrew/bin/loadafs /mnt/usr/andrew /andrew "" <Enter>
% cd / <Enter>
% su <Enter>
# umount /dev/hd1g <Enter>
```

Again, if you have changed your superuser password from what was in the original password file (a null password), then you will have to type it instead of the empty double quotes.

2.6. Load the usr volume.

(1) Type the following commands:

```
# newfs /dev/hd1g <Enter>
# mount /dev/hd1g /mnt <Enter>
# mkdir /mnt/usr <Enter>
# cd /mnt/usr <Enter>
```

(2) Insert the ROOT/USER tape into the streaming tape drive.

(3) Type the following commands:

```
# mt -f /dev/rst0 rewind <Enter>
# mt -f /dev/nrst0 fsf <Enter>
# /usr/ibm/ftc /dev/rst0 - | restore rvf - <Enter>
# rm restoresymtable <Enter>
# ln -s /andrew/andrew /mnt/usr/andrew <Enter>
# rmdir src <Enter>
# ln -s /andrew/src src <Enter>
# cd /mnt <Enter>
```

(4) Follow the instructions in Chapter 3 for installing the other tapes and diskettes (if you have them) with one exception: prepend a "/mnt" to all pathnames that you cd to before loading tapes. This applies to the following tapes and diskettes:

- the X11 tape (only the binaries should be loaded at this point).
- the 6152 Academic System 5-1/4 inch diskette (only if you are going to net-install to 6152 workstations.)
- the Professional Pascal diskette

(5) If you loaded the 6152 Academic System 5-1/4 inch diskette in the previous step, type the following commands:

```
# cd /mnt <Enter>
# mv vmunix.atr /vmunix.atr <Enter>
```

(6) Type the following commands to install the /mnt filesystem into the usr volume:

```
# exit <Enter>
% /usr/andrew/bin/loadafs /mnt/usr /andrew "" <Enter>
% /sys/dist/make.site <Enter>
```

Again, if you have changed your superuser password from what was in the original password file (a null password), then you will have to type it instead of the empty double quotes.

3. Load source into the Andrew File System.

This section describes how to load the IBM 4.3 source code onto the AFS server. If you do not want to serve source to all of your workstations, or do not have enough disk space, you may skip this section.

3.1. Load utility source.

Login to the AFS server machine as root and type the following commands:

```
# cd /vice/bin <Enter>
# ./createvol src vice1 /vicepa <Enter>
```

Now logout, and log in as admin on the master machine. Respond to the "password:" prompt with "**install** <Enter>". Insert the tape labeled *source* into the tape drive, and

type the following commands:

```
% cd /andrew <Enter>
% /usr/andrew/bin/fs mkmount src src <Enter>
% /usr/andrew/bin/fs sa src admin all <Enter>
% /usr/andrew/bin/fs sv src -a 0 <Enter>
% cd src <Enter>
% /usr/ibm/ftc /dev/rst0 - | /etc/restore rvf - <Enter>
% rm restoresymtable <Enter>
```

3.2. Load Andrew Toolkit Source.

If you are not already logged in as admin, then do so. Insert the IBM Andrew Toolkit tape into the tape drive, and type the following commands:

```
in Chapter 3 % cd /andrew/andrew <Enter>
% /usr/ibm/ftc /dev/rst0 - | tar xfp - makefile bc2 examples
                             ibmdoc ibmhelp ibmman overhead <Enter>
```

3.3. Load Andrew File System Source.

This section describes loading only the user-level code for the Andrew File System. The kernel code, which is on the IBM Andrew File System tape, is also on the root/usr tape, which has already been loaded into the Andrew File System.

If you are not already logged in as admin, then do so. Insert the IBM Andrew File System tape into the tape drive, and type the following commands:

```
% cd /andrew/andrew <Enter>
% mt -f /dev/nrst0 fsf <Enter>
% /usr/ibm/ftc /dev/rst0 - | tar xfp - server/src <Enter>
```

3.4. Load X11 Source.

Note: This section assumes you have already created and made a mount point for the utility source AFS volume. If you followed the above steps describing how to load utility source into the Andrew File System, then the volume will be have been already set up.

If you are not already logged in as admin, then do so. Insert the X11 tape into the tape drive, and type the following commands:

```
% cd /andrew <Enter>
% usr/ibm/ftc /dev/rst0 - | tar xfp - src <Enter>
```

4. Create the minimal root partition.

(1) Become superuser with the su command.

(2) Type the following commands:

```
# newfs /dev/hd1a <Enter>
# mkdir /minroot <Enter>
# mount /dev/hd1a /minroot <Enter>
```

(3) Edit the /etc/fstab file and add the following line:

```
/dev/hd1a /minroot ufs rw 1 3
```

(4) Type the following commands:

```
# cd /sys/dist <Enter>
```

```
# ./make.minimal  <Enter>
```

This command will take approximately 5 minutes to run. If it does not print any error messages, all went well.

4.1. Set up symbolic links into AFS

Type the following commands to set up symbolic links in /usr to the AFS usr volume:

```
# cd /usr  <Enter>
# rm -rf man doc sys lib/learn dict pub src lib/font  <Enter>
# ln -s /andrew/usr/man /andrew/usr/doc /andrew/usr/sys .  <Enter>
# ln -s /andrew/usr/dict /andrew/usr/pub .  <Enter>
# ln -s /andrew/usr/lib/learn lib/learn  <Enter>
# ln -s /andrew/usr/lib/font lib/font  <Enter>
# ln -s /andrew/src /usr/src  <Enter>
# ln -s /andrew/usr/include/X11 /usr/include  <Enter>
# ln -s /andrew/usr/bin/X11 /usr/bin/X11  <Enter>
# ln -s /andrew/usr/lib/libX11.a /usr/lib/libX11.a  <Enter>
# ln -s /andrew/usr/lib/libXtk11.a /usr/lib/libXtk11.a  <Enter>
# ln -s /andrew/usr/lib/X11 /usr/lib/X11  <Enter>
```

This completes the master machine installation procedure.

Appendix E. Building an NFS Master/Server Machine

This appendix assumes that the reader has systems level knowledge of the UNIX operating system.

This appendix describes how to set up an RT that can be used for network installation of nfs minimal and nfs reduced configured RT and 6152 systems.

When you have completed these instructions, the RT will contain the following:

- the IBM 4.3 base system root and /usr partitions.
- a "minimal root" partition (only if you will be installing nfs reduced systems)

The root and /usr file systems will be served through NFS to any nfs minimal systems installed from the master machine. Any nfs reduced systems that are installed from the master machine will only be served the /usr file system.

Disk Partitioning

The RT needs the following partitions:

- A standard root (hd0a) partition
- A standard /usr (hd0g) partition
- A 2 Megabyte partition for the minimal root partition "/minroot". This partition is only required if you will be installing nfs minimal systems. /dev/hd1a is a good choice for the /minroot file system. However, if you only have a single large disk, say a hd114e disk, in your master machine, you can use the last couple of megabytes on hd0 at the expense of the /dev/hd0g partition.

The minidisk partitions for the root disk are created as part of the procedure for installing the base IBM/4.3 system. The procedure for creating a minidisk entry for the minimal partition will be described shortly.

The remainder of this appendix provides the installation steps required.

1. **Perform a tape or network installation of IBM/4.3 on the RT.** Follow the instructions for installing an RT from streaming tape, or through the network found in Chapter 3 of this article. If you perform a network installation, be sure to install a "full" system, that is, a system with local root and /usr file systems. This is the default with a 70 Megabyte (or larger) disk for drive 0. If you are going to be installing nfs minimal systems from this master machine, you will have to create a minidisk entry for the /minroot partition. If this partition is going to reside on hd0, the following modifications to the standard installation instructions in chapter 3 must be made:

   After creating the minidisk entries for hd0 by typing **standard root** **<Enter>**, and before typing **quit** **<Enter>**, perform the following steps:

   (1) Type **list** **<Enter>** to display the current minidisk entries. Now delete the *hd0g* entry by typing **delete hd0g** **<Enter>**. In response to the query: *confirm minidisk deletion (all data will be lost) [y/n]*, type **y** **<Enter>**.

   (2) Note the size (in blocks) that the hd0g minidisk entry used to be. Subtract 4000 from that number. This will be the size of the new hd0g minidisk entry. Now type **create** **<Enter>**. You will be prompted for *name, iodn, size,* and *type* fields. Enter the same values that the hd0g entry used to have, substituting the new (smaller) size for the original one.

(3) Now create a *hd0f* minidisk entry by typing **create** **<Enter>**. Again, you will be prompted for *name, iodn, size,* and *type* values. The *name* field should be **hd0f**. The *iodn* field should be one plus the number of the hd0g iodn field. The *size* and value fields should both be zero. (When a size of *zero* is entered, it is interpreted as meaning "use up all of the remaining space").

(4) When you are done, pick up where you left off with the original installation instructions. Note that this will cause the hd0g partition (the /usr file system) to be two megabytes smaller than it would have been. Keep this in mind when you decide which optional components to install.

If you don't want to use hd0 for the /minroot partition, you can use hd1. The "a" partition is a good choice. Here's a slight modification to the installation instructions in section 3 that allows you to use hd1a for the /minroot partition. when you get to the section where hd1 gets formatted and the minidisk tables get created, go ahead and perform those steps even if you are not installing utility source.

After the machine has been loaded with the base system, and is in multi-user mode, you need to perform the following steps:

(1) Make the machine an NFS server. To do this, add the following lines to */etc/exports*:

```
/ slave
/usr slave
```

If the machine that you will be net-installing will not be called *slave*, then substitute whatever hostname it will have. If you will be installing more than one nfs minimal or nfs reduced machine, then list all of the hostnames on the same line after the file system name. You may also omit all of the hostnames, causing the server to consider itself authorized to serve to any host. If you don't plan on installing any nfs minimal systems, then you may omit the **/ slave** line from the */etc/exports* file.

(2) Create the /minroot file system. This step is only required if you will be installing nfs minimal systems. nfs reduced systems do not require the /minroot partition on the master machine. To do this, log in as root, newfs the partition that you reserved for the /minroot file system, and mount it. If you are going to use hd1a, then type

```
# newfs /dev/hd1a <Enter>
# mkdir /minroot <Enter>
# mount /dev/hd1a /minroot <Enter>
```

Add the following line to /etc/fstab:

```
# /dev/hd1a /minroot ufs rw 1 2
```

Type the following to create the /minroot file system:

```
# cd /sys/dist <Enter>
# ./make.minimal -nfs <Enter>
```

If you are using some partition other than hd1a for the /minroot file system, then substitute its name for hd1a.

(3) Create links into the /site directory. Since the nfs reduced and nfs minimal systems that are being served by this master machine will be sharing the same /usr file system, some way of allowing the various machines to have their own private copies of some directories in /usr that need to be written to. Some examples of this type of directory are */usr/spool, /usr/adm,* and */usr/tmp*. The way that this is arranged is to replace the directories in /usr on the master machine with symbolic links pointing to directories in /site. On the master machine, /site will be a symbolic link to /usr/site. All of the files and subdirectories in /usr that get replaced with symbolic links will first get copied into /usr/site. On the nfs minimal and nfs reduced machines, /site will be a real directory

on the local hard disk. The following commands will setup the symbolic links and the /usr/site directory on the master machine. Before executing them, make sure that /site and /usr/site do not already exist. To make the links, type:

```
# cd /sys/dist <Enter>
# ./make.site -m -l /usr/site -u /usr <Enter>
```

(4) Add lines of the form "**hostname root**" to ~operator/.rhosts, where *hostname* is the name of a machine the you will be net-installing.

(5) If you will be installing any 6152 machines, insert the 6152 5-1/4 inch feature diskette in the drive, and type the following command:

```
# dd if=/dev/rfd0 bs=30b | uncompress -v | tar xvfp - <Enter>
```

This completes the special installation steps.

This page intentionally left blank.