

Starting OS-9

Booting OS-9

Before using OS-9 on your computer, you must *boot* the system. Booting is also called a *cold start* or *bootstrapping*. It involves the computer reading a portion of the system disk (or tape) into memory.

If your system is a standard disk-based computer, the system disk contains all the modules that make up OS-9. The system disk usually contains other files and directories frequently used during normal operations. This includes a directory for each user, a shared commands directory, and files used by the system. A description of the directories commonly supplied with Professional OS-9 is provided at the end of this chapter.

Two files, which are called *startup* and *OS9Boot* by convention, need to be discussed here. *startup* is a shell procedure file that is processed immediately after the system starts running. *startup* may contain any legal OS-9 command or program. *OS9Boot* contains the OS-9 system modules that are read into memory. The chapter on OS-9 system management contains information on changing the *startup* and *OS9Boot* files.



The boot procedure varies depending on the requirements of the specific hardware. The manufacturer supplies detailed instructions outlining the boot procedure for the specific system involved. You should follow the instructions as specified.

If the system fails to boot, recheck the hardware setup instructions, especially if you made any modifications to your computer. Make sure the disk (or tape) was inserted correctly, and try the boot sequence again. If the boot sequence fails several times, contact your supplier.

When the system boots correctly, a welcoming message is displayed followed by the `setime` prompt. The `setime` utility starts the system clock and allows OS-9 to keep track of the date and time of the creation of new files. The clock must be running for multi-tasking to take place.

The clock may be started by the Init module (refer to the chapter on OS-9 system management for more information). If it is not started and you have a system with a battery-backed clock, type the following command to start the system clock:

```
$ setime -s
```

Otherwise, execute `setime` by typing:

```
$ setime
```

`setime` prompts with the following:

```
yy/mm/dd hh:mm:ss [am/pm]
Time ?
```

At the prompt, enter the year, month, day, hour, minutes, seconds, and optionally am or pm. Unless am or pm is specified, `setime` uses the 24 hour clock. For example, 15:20 is the same as 3:20 pm. The input is one or two digit numbers with a space, colon, semicolon, comma, or slash used as a field delimiter. If a semicolon is used, the entire date string must be within quotes. For example, to set the time on May 14, 1991 at 1:24 pm, type:

```
91/5/14/1/24/pm      or   91 05 14 1 24 pm      or   91,5,14,13,24      or
91:5:14:13:24       or   91/5/14/13/24       or   "91;5;14;13;24"
```

To find out if the system clock is running or if the date and time was set correctly, use the `date` command. For example:

```
$ date
July 2, 1990 Monday 1:25:26pm
```

Once the time and date have been properly set, the system displays the following prompt:

```
$
```

The `$` prompt means the operating system is active and waiting for you to enter a command line. This prompt is the default system prompt. This manual uses the `$` prompt for all examples. For information on changing the shell prompt, refer to the chapter on the shell.

+ **NOTE:** The following sections are specifically intended for systems distributed with floppy disk system disks. These sections are also of general interest in terms of formatting and backing up floppy disks. If you have a hard disk or are booting from a media other than a disk, refer to the OS-9 system management chapter.

Backing Up the System Disk

Before experimenting with OS-9, you should make a backup of your master system disk. The backup procedure involves making an exact copy of a disk. If for any reason your system disk becomes damaged, it may become unreadable. For this reason, it is important to have another copy stored safely away.

Before you can backup your system disk, you need a properly formatted disk. New disks cannot be read from or written to until they have been formatted. The `format` utility initializes new disks for reading and writing. `backup`, the OS-9 utility that makes copies of disks, *requires* the backup disk to be the same size and format as the original disk.

The following section describes the steps to be taken to backup a disk on a typical OS-9 system that boots from a floppy drive (usually called /d0).

NOTE: Before formatting your first disk, it is strongly recommended that you read the entire section on formatting disks.

+ **NOTE:** A list of the naming conventions OS-9 uses is located in the chapter on the shell.

Formatting a Disk

The format of OS-9 system disks vary by the type of disk drive and by manufacturer. Usually, the format is set to be the maximum capacity of the disk drive.

You can place several parameters on the command line with the `format` command:

- sd for single density disks
- dd for double density disks
- ss for single sided disks
- ds for double sided disks

Refer to your hardware documentation for the maximum capacity of your drives. Refer also to the label of your system disk for the proper format of your backup copy. Consult the `format` utility description in the **OS-9 Utilities** section for other available parameters.

Multiple Drive Format

If your system has two disk drives, place the system disk in the first drive and the new disk in the second drive. The second drive is usually called /d1. At the \$ prompt, type `format`, the drive name of the new disk, any desired options and press the <return> key to enter the command line:

```
$ format /d1
```

This command line specifies that the disk in the second drive will be formatted as a double-sided, double-density disk. If your disk is different, your options will be different.

Single Drive Format

If your system has only one disk drive, you will need to load the `format` utility into memory. The `load` utility puts a copy of a program into the memory of the computer. Once `format` has been loaded into memory, you can remove your system disk from the drive. OS-9 can execute the copy of `format` that resides in memory. Any OS-9 utility can be loaded and executed in this fashion.

To load the `format` utility into memory, type the following command at the \$ prompt:

```
load format
```

When `format` has been loaded, remove the system disk from the drive. Place the disk to format into the drive. At the \$ prompt, type:

```
format /d0
```

This command line formats the disk.

Continuing the Formatting Process with Either a Single Drive or a Multiple Drive

In the case of both single and multiple drive systems, `format` displays the specific disk format settings, followed by a prompt:

```
Formatting device: <drive name>  
proceed?
```

NOTE: <drive name> is replaced by the name of the device on which you are trying to format. For example, /d0.

If the drive name in the prompt is not the name of the drive with the blank disk, type `q` to quit, or your only system disk may be erased.

If the drive name and parameters in the prompt are correct, type `y` for yes. If you type `y` at the prompt, there will be a pause while the disk is being formatted. `format` then prompts for the name of the disk:

```
volume name:
```

After you have entered the volume name, `format` prints:

verifying media, building bitmap...

During the final phase of the process, the hexadecimal number of each track is displayed as each track is verified to see if any sectors are bad. If any bad sectors are found, an error message is displayed along with the number of the bad sector. The number of good sectors, the number of unusable sectors, and the total number of verified sectors is also displayed.

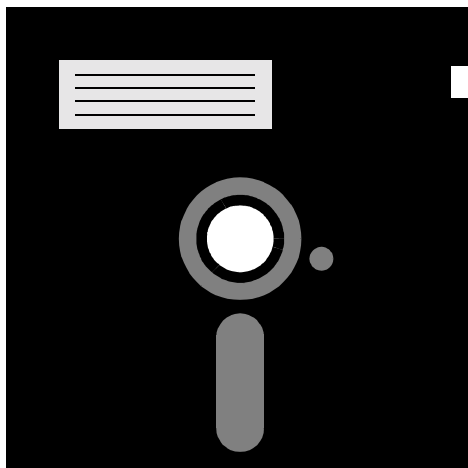


WARNING: Never backup a system disk to a disk that has any bad sectors reported by format.

The Backup Procedure

After a disk is formatted, you can run **backup**. The **backup** utility makes an exact copy of the OS-9 system disk. There are other ways to make a copy of a disk, but this method is the least complicated. The backup process involves copying everything from your system disk to a formatted disk. During the backup procedure, the system disk is referred to as the **source disk**. The backup disk is called the **destination disk**.

NOTE: This procedure makes copies of any disk, not just the system disk.



← Write Protect Tab

NOTE: You may wish to **write protect** your source disk with a write protect tab when using the **backup** procedure. This prevents any accidental confusion in exchanging the source and destination disks.

backup makes two passes. The first pass reads a portion of the source disk into a buffer in memory and writes it to the destination disk. The second pass verifies that everything was copied to the new disk correctly.

If an error occurs on the first pass, something is wrong with the source disk or the drive it is in.

If an error occurs during the second pass, the problem is with the destination disk. If **backup** repeatedly fails on the second pass, reformat the disk to make sure it has no bad sectors. If the disk reformats correctly, try the backup procedure again.

Multiple Drive Backup

If your system has two disk drives, place the source disk in the first drive (/d0) and the destination disk in the second drive (/d1). At the \$ prompt, type **backup** and press the <return> key.

The system assumes you want to backup the disk in /d0. It responds to **backup** with the following prompt:

ready to BACKUP /D0 to /D1?

If you have placed the correct disks in the correct drives, type **y** for yes. Otherwise, type **q** for quit. If you type **y**, the system copies all information on the disk in /d0 on to the disk in /d1 and returns the \$ prompt.

Single Drive Backup

If your system has only one drive, the **backup** utility needs to be loaded into memory. Make sure your system disk is in /d0 and type the following command:

load backup

After you have loaded **backup**, you may proceed with the backup procedure. Type the following command:

backup /d0 -b=100k

This tells the system that you are performing a single drive backup and that you want to use a 100K buffer for the backup. If your system will allow you to use a larger buffer, increase this number. The larger the buffer, the fewer swaps you will have to make. The system responds with the following prompt:

ready to BACKUP /D0 to /D0?

Type **y** if you are ready to perform the backup. Otherwise, type **q** for quit. If you type **y**, the system begins a series of prompts to complete the backup procedure. This consists of swapping the source and destination disks in the disk drive as prompted by the system.

The first prompt is:

ready destination, hit a key

At this prompt, remove the source disk from the drive and insert the destination disk. Once this is done, press any key to continue the backup procedure. The next system prompt is:

ready source, hit a key

At this prompt, remove the destination disk from the drive and insert the source disk. Once this is done, press any key to continue the backup procedure. The exchanging of disks continues until the backup procedure is completed.

+ When you have backed up the system disk, store the original disk in a safe place and use the duplicate as your working system disk.

Directories Contained on the System Disk

The following is a list of directories commonly distributed with Professional OS-9. They are all contained in the primary directory (the root directory) of your system:

- BOOTOBJS** Contains the system modules for bootstrap files, system-specific files, etc.
- C** Contains Cstart source code and an example of trap handlers for user education.
- CMDS** Contains all the system utilities such as **backup**, **load**, **settime**, etc. Many of the utilities are discussed in the following chapters. The **OS-9 Utilities** section contains descriptions of each utility distributed with Professional OS-9.
- DEFS** Contains several files of symbolic definitions that are useful when using programming languages.
- IO** Contains the device descriptor source for system customization. For more information on changing device descriptors, refer to the chapter on OS-9 system management.
- LIB** Contains system library files.
- MACROS** Contains general macros used in driver development, etc.
- SYS** Contains system files including:
 - Errmsg** Contains text for descriptions of error messages. An appendix listing the error messages is included with this manual set.
 - password** Contains a sample password file for timesharing systems. The password file contains information such as the user name, password, initial process, etc. for each user. For more information on the password file, refer to the chapter on the shell in this manual and the **login** utility in the **OS-9 Utilities** section.
 - termcap** Contains descriptions of your terminal characteristics. For more information on the termcap file, refer to the chapter on OS-9 system management.
- SYSMODS** Contains the source for **SysGo** and **init** for system customization. For more information on **SysGo** and **init**, see the **OS-9 Technical Manual**.

End of Chapter 2