

CHAPTER

2

How Hardware and Software Work Together

In this chapter, you will learn:

- How an operating system manages hardware
- How system resources help hardware and software communicate
- The steps involved in booting your computer

Computer systems contain both hardware and software, and computer technicians must understand how they interact. Although the physical hardware is the visible part of a computer system, the software is the intelligence of the system that enables the hardware components to work. After reading this chapter, you should have a general understanding of how hardware and software work together, and what happens

when a PC is first turned on. You will also learn about error messages that can occur during the boot. You can look at this chapter as your crash course on operating systems from the perspective of a hardware technician!

How an Operating System Manages Hardware

An **operating system (OS)** is software that controls a computer. It manages hardware, runs applications, provides an interface for users, and stores, retrieves, and manipulates files. In general, an operating system acts as the middleman between applications and hardware (see Figure 2-1).

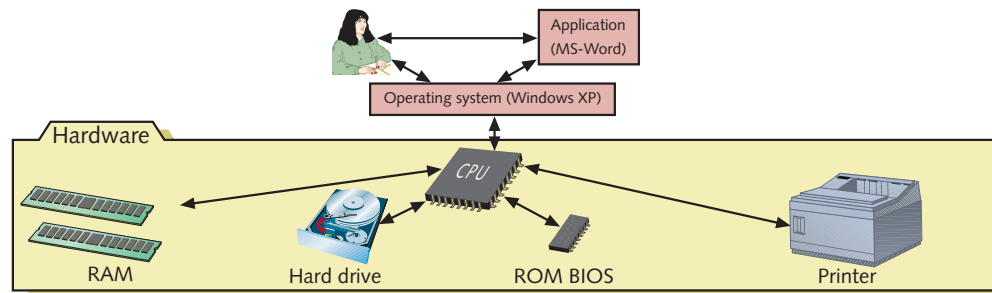


Figure 2-1 Users and applications depend on the OS to relate to all hardware components

Several applications might be installed on a computer to meet various user needs, but it only needs one operating system. The most popular operating systems for personal computers today are Microsoft Windows XP, Windows 2000, and Windows 98/Me. Other Microsoft operating systems for PCs that are outdated or becoming outdated are Windows NT, Windows 95, and DOS. There are other OSs not made by Microsoft, including Linux and the Mac OS. When you learn about hardware, it is sometimes important to know how an OS installs devices and how to use the OS to help in troubleshooting a failed device. In this book, for these purposes, we will be using Windows XP or Windows 98, as these are the OSs you are most likely to be called on to support.

An operating system is responsible for communicating with hardware, but the OS does not relate directly to the hardware. Rather, the OS uses device drivers or the BIOS to interface with hardware. Figure 2-2 shows these relationships. Therefore, most PC software falls into three categories:

- Device drivers or the BIOS
- Operating system
- Application software

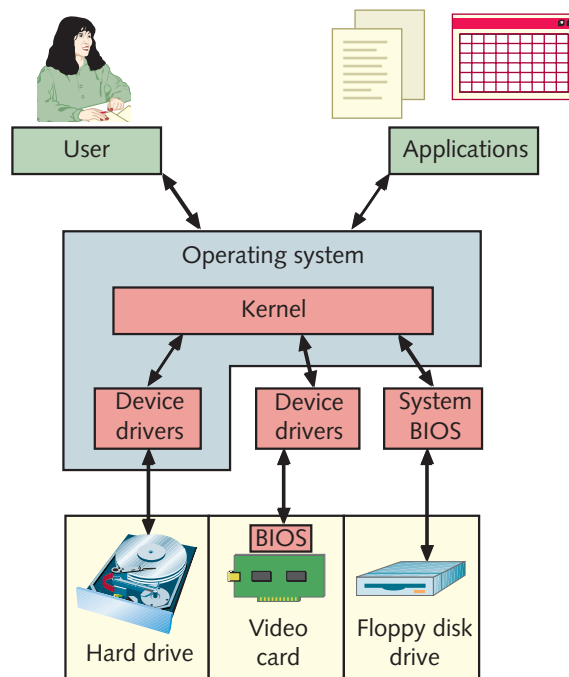


Figure 2-2 An OS relates to hardware by way of BIOS and device drivers

Device drivers are small programs stored on the hard drive that tell the computer how to communicate with a specific hardware device such as a printer, network card, or modem. Recall from Chapter 1 that the basic input/output system (BIOS) on the motherboard is hard-coded or permanently coded into a computer chip called the ROM BIOS chip or firmware chip. BIOS programs fall into three categories: programs to control I/O devices (called **system BIOS**), programs to control the startup of a computer (called **startup BIOS**), and a program to change the setup information stored in CMOS (called **CMOS setup**). Next we look at how an OS uses device drivers and the BIOS to manage hardware.

How an OS Uses Device Drivers

A+
CORE
1.8

Device drivers are software designed to interface with specific hardware devices. They are stored on the hard drive and installed when the OS is first installed or when new hardware is added to a system. The OS provides some device drivers, and the manufacturer of the specific hardware device with which they are designed to interface provides others. In either case, unlike BIOS, device drivers are usually written for a particular OS and might need to be rewritten for use with another.

When you purchase a printer, DVD drive, Zip drive, digital camera, scanner, or other hardware device, bundled with the device is a set of floppy disks or CDs that

contain the device drivers (see Figure 2-3). You must install these device drivers under the operating system so it will have the necessary software to control the device. In most cases, you install the device and then install the device drivers. There are a few exceptions, such as a digital camera using a USB port to download pictures. In this case, most often you install the software to drive the digital camera before you plug in the camera. See the device documentation to learn what to do first. Later chapters cover device driver installations.



Figure 2-3 A device such as this CD-ROM drive comes packaged with its device drivers stored on a floppy disk or other media. Alternately, you can use device drivers built into the OS.



Device drivers come from a number of sources. Some come with and are part of the operating system, some come with hardware devices when they are purchased, and some are provided for downloading over the Internet from a device manufacturer's Web site.

There are two kinds of device drivers: 16-bit real-mode drivers and 32-bit protected-mode drivers. Windows 95 and Windows 98 support both, but Windows Me and Windows NT/2000/XP use only 32-bit drivers. Windows 9x and Windows 2000/XP provide hundreds of 32-bit drivers for many different kinds of devices, and device manufacturers also provide their own 16- or 32-bit drivers, which come bundled with the device or can be downloaded from the device manufacturer's Web site.

Before installing a new hardware device on a Windows 2000/XP system, always check the **hardware compatibility list (HCL)** to determine if a driver will work under Windows 2000/XP.

A+
CORE
1.8

Go to the Microsoft Web site and search for your device:

www.microsoft.com/whdc/hcl/search.msp

If the device does not install properly or produces errors, check the manufacturer's Web site for a driver that the manufacturer says is compatible with Windows 2000/XP.

Windows 2000/XP and Windows 9x keep information about 32-bit drivers in the Windows registry, a database of hardware and software settings, Windows configuration settings, user parameters, and application settings.

Sometimes, to address bugs, make improvements, or add features, manufacturers release device drivers that are more recent than those included with Windows or bundled with the device. Whenever possible, it is best to use the latest driver available for a device provided by the device manufacturer. You can usually download these updated drivers from the manufacturer's Web site. You will learn how to install, update, and troubleshoot drivers in later chapters.

2

APPLYING CONCEPTS

Suppose you have just borrowed an HP 995c Deskjet printer from a friend, but you forgot to borrow the CD with the printer drivers on it. You could go to the Hewlett-Packard Web site (www.hp.com), download the drivers to a folder on your PC, and install the driver under Windows. Figure 2-4 shows you a Web page from the site listing downloadable drivers for inkjet printers. Search the HP site and find the driver for your borrowed HP 995c printer.

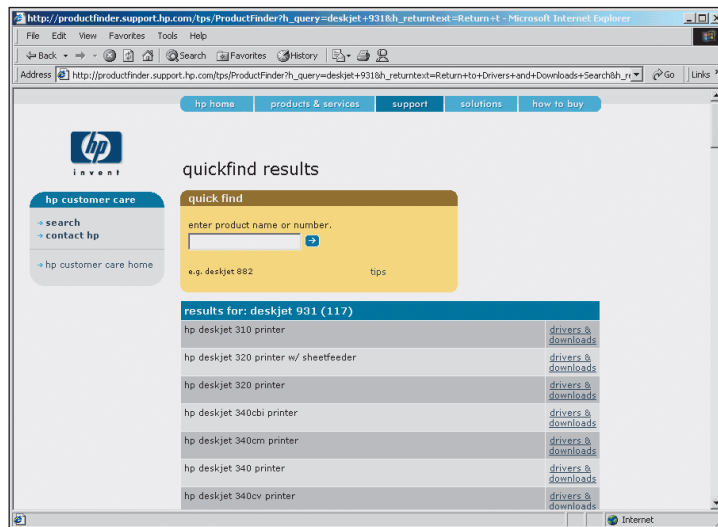


Figure 2-4

Download the latest device drivers from a manufacturer's Web site

How an OS Uses System BIOS to Manage Devices

A+
CORE
1.8

The OS communicates with simple devices, such as floppy drives or keyboards, through system BIOS. In addition, system BIOS can be used to access the hard drive. In some cases, an OS has a choice of using system BIOS or device drivers to access a device. Most often it uses device drivers because they are faster. The trend today is to use device drivers rather than the BIOS to manage devices.



There is a good way to determine whether the BIOS or a device driver is controlling a device. If the device is configured using CMOS setup, most likely system BIOS controls it. If the device is configured using the OS, most likely a driver controls it. Sometimes you can use the Windows System Information or Device Manager utilities to find out the name of a driver controlling a device.

For example, in Figure 2-5, the setup main menu for an Award BIOS system lets you configure, or set, the system date and time, the Supervisor Password (power-on password), floppy disk drives, the hard drive, and the keyboard. Figure 2-6 shows another setup window for this same BIOS that can configure serial ports, an infrared port, and a parallel port. System BIOS can control all these devices. On the other hand, there is no setup window in this BIOS to control the DVD drive or Zip drive installed on this system. The BIOS is not aware of these devices; this means they are controlled by device drivers.



CMOS setup windows are accessed during startup. A system displays a message at the bottom of the screen saying something like, “Press Del to enter setup.” Pressing the indicated key launches a program stored on the ROM BIOS microchip to change the contents of CMOS RAM. This BIOS setup program provides windows like those in Figures 2-5 and 2-6.

Recall that the system BIOS is stored in ROM. Because access to RAM is faster than access to ROM, at startup a system might copy the system BIOS from ROM to RAM in order to improve performance. This practice is called **shadowing ROM**, or just **shadow RAM**. Because the system BIOS is not used often, if CMOS setup gives you the option, you might want to disable shadow RAM in order to conserve RAM.



Sometimes a system becomes unstable (crashes at unexpected times). If your CMOS setup gives you the option, one thing you can try to stabilize the system is to disable shadow RAM.

An OS uses BIOS or device drivers to manage hardware devices. The BIOS or driver communicates with a device by way of system resources on the motherboard. We next look at these resources and how they work.

A+
CORE
1.8

2

AwardBIOS Setup Utility		
Main	Advanced	Power Boot Exit
System Time	[11:42:09]	Item Specific Help <Enter> to select field; <+>, <-> to change value
System Date	[04/04/2000]	
Legacy Diskette A	[1.44M, 3.5 in.]	
Legacy Diskette B	[None]	
Floppy 3 Mode Support	[Disabled]	
► Primary Master	[Auto]	
► Primary Slave	[Auto]	
► Secondary Master	[Auto]	
► Secondary Slave	[Auto]	
► Keyboard Features		
Language	[English]	
Supervisor Password	[Disabled]	
User Password	[Disabled]	
Halt On	[All Errors]	
Installed Memory	128MB	
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults ESC Exit ← Select Menu Enter Select Sub Menu F10 Save and Exit		

Figure 2-5

Use the BIOS setup main menu for Award BIOS to configure some of the devices controlled by system BIOS

Award BIOS Setup Utility		
Advanced		
I/O Device Configuration		Item Specific Help
Onboard FDC Swap A & B	[No Swap]	<Enter> to select if switch drive letter assignments or not.
Floppy Disk Access Control	[R/W]	
Onboard Serial Port 1:	[3F8H/IRQ4]	
Onboard Serial Port 2:	[2F8H/IRQ3]	
UART2 Use Infrared	[Disabled]	
Onboard Parallel Port:	[378H/IRQ7]	
Parallel Port Mode:	[ECP + EPP]	
ECP DMA Select:	[3]	
F1 Help ↑↓ Select Item -/+ Change Values F5 Setup Defaults ESC Exit ← Select Menu Enter Select Sub Menu F10 Save and Exit		

Figure 2-6

Use this Award BIOS setup window to configure several I/O devices, including the serial, parallel, and infrared ports

System Resources

A+
CORE
1.4

A **system resource** is a tool used by either hardware or software to communicate with the other. When BIOS or a driver wants to send data to a device (such as when you save a file to the hard drive), or when the device needs attention (such as when you press a key on the keyboard), the device or software uses system resources to communicate. There are four types of system resources: memory addresses, I/O addresses, interrupt request numbers (IRQs), and direct memory access (DMA) channels. Table 2-1 lists these system resources used by software and hardware, and defines each.

System Resource	Definition
IRQ	A line of a motherboard bus that a hardware device can use to signal the CPU that the device needs attention. Some lines have a higher priority for attention than others. Each IRQ line is assigned a number (0 to 15) to identify it.
I/O addresses	Numbers assigned to hardware devices that software uses to send a command to a device. Each device “listens” for these numbers and responds to the ones assigned to it. I/O addresses are communicated on the address bus.
Memory addresses	Numbers assigned to physical memory located either in RAM or ROM chips. Software can access this memory by using these addresses. Memory addresses are communicated on the address bus.
DMA channel	A number designating a channel on which the device can pass data to memory without involving the CPU. Think of a DMA channel as a shortcut for data moving to and from the device and memory.

Table 2-1 System resources used by software and hardware

As Table 2-1 explains, all four resources are used for communication between hardware and software. **Hardware devices signal the CPU for attention using an IRQ. Software addresses a device by one of its I/O addresses. Software looks at memory as a hardware device and addresses it with memory addresses, and DMA channels pass data back and forth between a hardware device and memory.**

All four system resources depend on certain lines on a bus on the motherboard (see Figure 2-7). A bus such as the system bus has three components: the data bus carries data, the address bus communicates addresses (both memory addresses and I/O addresses), and the control bus controls communication (IRQs and DMA channels are controlled by this portion of the bus). Let’s turn our attention to a more detailed description of the four resources and how they work.

A+
CORE
1.4

2

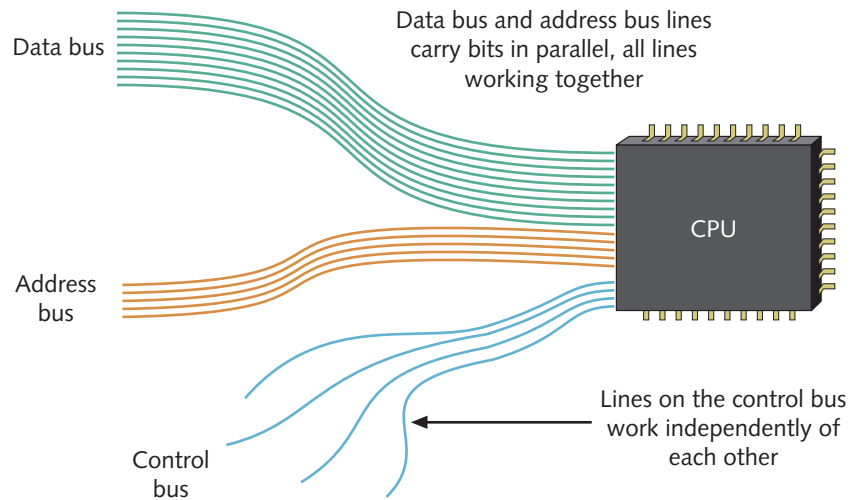


Figure 2-7 A bus consists of a data bus, an address bus, and a control bus

Interrupt Request Number (IRQ)

When a hardware device needs the CPU to do something—for instance, when the keyboard needs the CPU to process a keystroke after a key has been pressed—the device needs a way to get the CPU’s attention, and the CPU must know what to do once it turns its attention to the device. These interruptions to the CPU are called **hardware interrupts**, and the device initiates an interrupt by placing voltage on the designated **IRQ (interrupt request) line** assigned to it. This voltage on the line serves as a signal to the CPU that the device has a request that needs processing. Often, a hardware device that needs attention from the CPU is referred to as “needing servicing.” Interrupts initiate many processes that the CPU carries out, and these processes are said to be “interrupt-driven.”

Table 2-2 lists common uses for the sixteen IRQs. I/O addresses also listed in the table are discussed in the next section.

IRQ	I/O Address	Device
0	0040-005F	System timer
1	0060-006F	Keyboard controller
2	00A0-00AF	Access to IRQs above 7
3	02F8-02FF	COM2 (covered in Chapter 8)

Table 2-2 (continued)

A+
CORE
1.4

IRQ	I/O Address	Device
3	02E8-02EF	COM4 (covered in Chapter 8)
4	03F8-03FF	COM1 (covered in Chapter 8)
4	03E8-03EF	COM3 (covered in Chapter 8)
5	0278-027F	Sound card or parallel port LPT2 (covered in Chapter 8)
6	03F0-03F7	Floppy drive controller
7	0378-037F	Printer parallel port LPT1 (covered in Chapter 8)
8	0070-007F	Real-time clock
9-10		Available
11		SCSI or available
12	0238-023F	Motherboard mouse
13	00F8-00FF	Math coprocessor
14	01F0-01F7	IDE hard drive (covered in Chapter 7)
15	0170-017F	Secondary IDE hard drive or available (covered in Chapter 7)

Table 2-2 IRQs and I/O addresses for devices**A+ EXAM TIP**

The A+ Core exam expects you to have the IRQ assignments in Table 2-2 memorized.

In Table 2-2, notice the COM and LPT assignments. COM1 and COM2 are preconfigured assignments that can be made to serial devices such as modems, and LPT1 and LPT2 are preconfigured assignments that can be made to parallel devices such as printers. For example, rather than being assigned an IRQ and some I/O addresses, the modem uses the assignments previously made to COM2, which makes it easier to configure the modem and to avoid conflicts with other devices that also need an IRQ and some I/O addresses. You will learn more about COM and LPT assignments in Chapter 8.

On motherboards, part of the chip set called the interrupt controller manages the IRQs for the CPU. The CPU actually doesn't know which IRQ is "up" because the interrupt controller manages that. If more than one IRQ is up at the same time, the interrupt controller selects the IRQ that has the lowest value to process first. For example, if a user presses a key on the keyboard at the same time that she moves the mouse configured to use COM1, the keystroke is processed before the mouse action, because the keyboard is using IRQ 1 and the mouse on COM1 is using IRQ 4. Think of the interrupt controller as the "inside man" with the CPU. All devices wait outside the door for the controller to let the CPU know what they need.

The interrupt controller on early motherboards was designed to handle only eight different IRQs. IRQ 2 was reserved because it was intended to be used as part of a link to mainframe computers, and IRQs 0 and 1 are always in use by the system clock and keyboard. Therefore, only five IRQs were available for devices, and each

A+
CORE
1.4

device had to have its own IRQ. This made it difficult for more than five devices to be connected to a PC at any one time. In order to accommodate the need for more devices, a second group of IRQs was later added (IRQs 8 through 15), and a second interrupt controller was added to manage these new IRQs.



A+ EXAM TIP

The A+ Core exam expects you to know the IRQ priority levels.

This second controller did not have access to the CPU, so it had to communicate with the CPU through the first controller (see Figure 2-8). To signal the first controller, the second controller used one of the first controller's IRQ values (IRQ 2). These last eight IRQs plug into the system using IRQ 2. Because of this, the IRQ priority level became: 0, 1, (8, 9, 10, 11, 12, 13, 14, 15), 3, 4, 5, 6, 7.

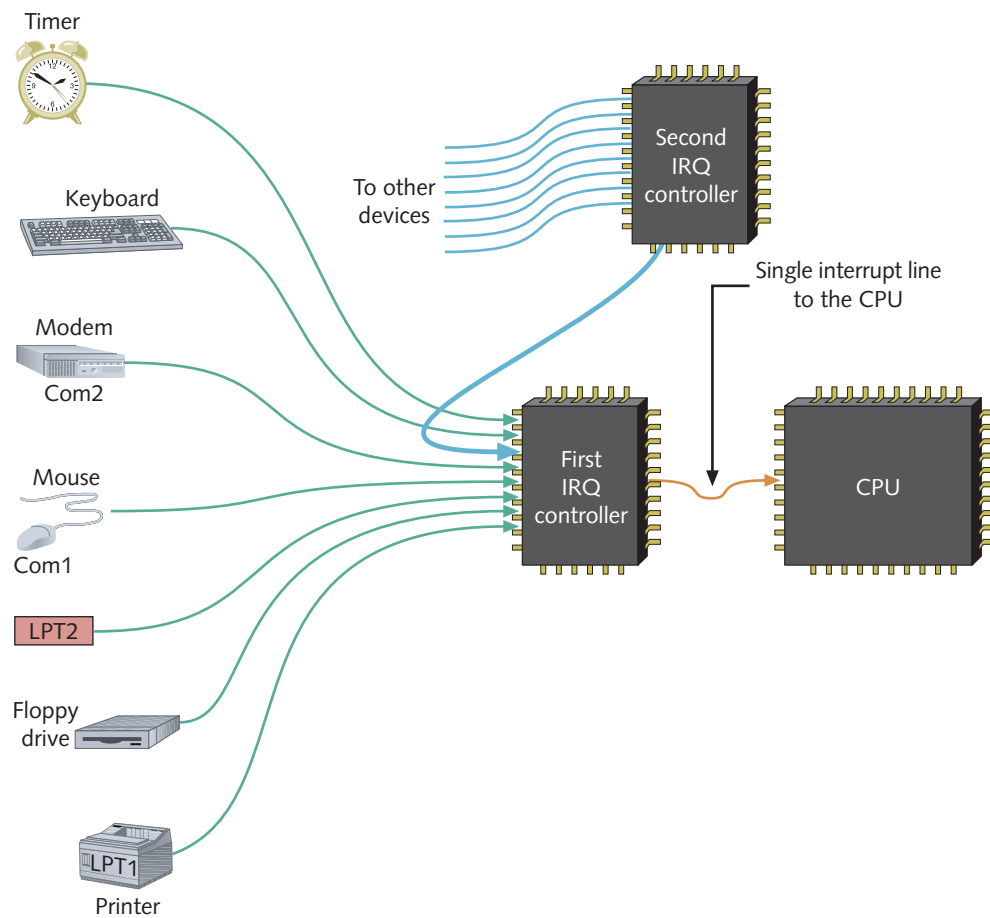


Figure 2-8

The second IRQ controller uses IRQ 2 to signal the first IRQ controller

APPLYING CONCEPTS

To see how the IRQs are assigned on your computer, use MSD for DOS and Device Manager for Windows 2000/XP and Windows 9x. (Windows NT does not have Device Manager.) For Windows XP, click Start, right-click My Computer, and select Properties on the shortcut menu. For Windows 2000, right-click My Computer on the desktop and select Properties on the shortcut menu. The System Properties dialog box appears. Click the Hardware tab and then click the Device Manager button. (For Windows XP, see Figure 2-9.) On the menu, click View, and then click Resources by Type, if necessary. Click the plus sign next to Interrupt request (IRQ) to open the list of assigned IRQs. Notice in the figure that IRQs 9 and 11 are each being shared by two devices.



NOTE

Sharing IRQs is not possible with ISA devices on the ISA bus. However, newer buses are designed to allow more than one device to share an IRQ. In Chapter 4, you will see how the PCI and USB buses do this.

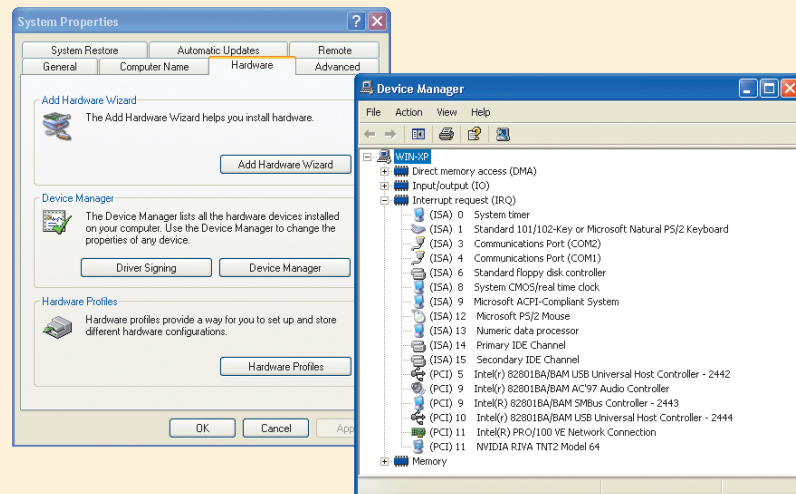


Figure 2-9 Use Device Manager to see how your system is using IRQs and other system resources

To see current assignments in Windows 9x, click Start, point to Settings, click Control Panel, and double-click System. Click the Device Manager tab, select Computer, and then click Properties. Figure 2-10 shows the Computer Properties dialog box. Notice that IRQ 2 is assigned to the programmable interrupt controller because it is being used to manage IRQs 8 through 15.

A+
CORE
1.4

2

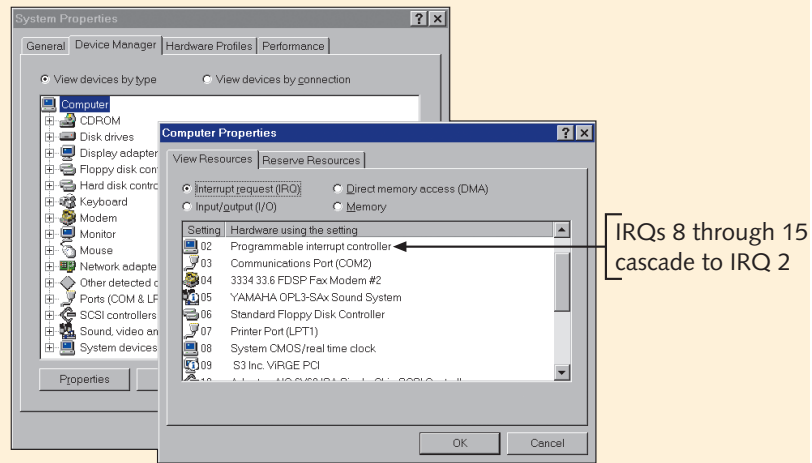


Figure 2-10 Windows 9x Device Manager shows current assignments for system resources

With interrupts, the hardware device or the software initiates communication by sending a signal to the CPU, but a device can be serviced in another way, called polling. With **polling**, software that is constantly running has the CPU periodically check the hardware device to see if it needs service. Not very many devices use polling as the method of communication; most hardware devices use interrupts. A joystick is one example of a device that does use polling. Software written to manage a joystick has the CPU check the joystick periodically to see if the device has data to communicate, which is why a joystick does not need an IRQ to work.

Memory Addresses

An operating system relates to memory as a long list of cells that it can use to hold data and instructions, somewhat like a one-dimensional spreadsheet. Each memory location or cell is assigned a number beginning with zero. These number assignments are made when the OS is first loaded and are called **memory addresses**. Think of a memory address as a seat number in a theater (see Figure 2-11). Each seat is assigned a number regardless of whether someone is sitting in it. The person sitting in a seat can be data or instructions, and the OS does not refer to the person by name but only by the seat number. For example, the OS might say, "I want to print the data in memory addresses 500 through 650."

A+
CORE
1.4

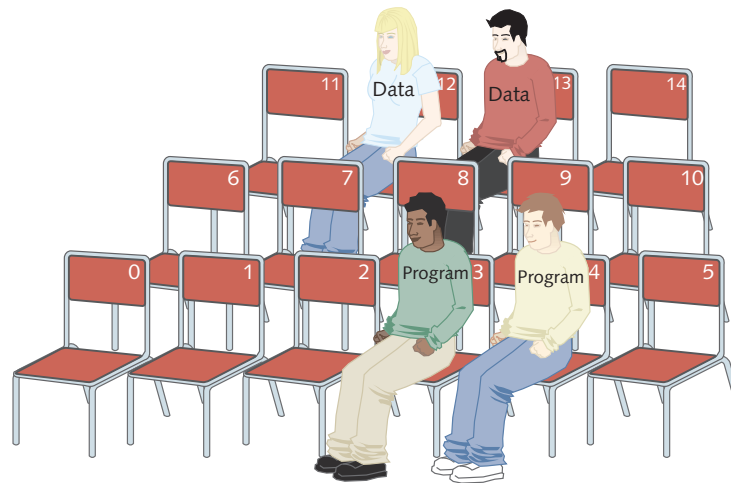


Figure 2-11 Memory addresses are assigned to each location in memory, and these locations can store data or instructions

These addresses are most often displayed on the screen as hexadecimal (base 16 or hex) numbers in segment:offset form (for example, C800:5, which in hex is C8005 and in decimal is 819,205).



NOTE

Windows offers a calculator that can quickly convert numbers in binary, digital, and hexadecimal. Enter a number in one number system, and then click another number system to make the conversion. To access the calculator in Windows NT/2000/XP or Windows 9x, click Start, Programs, Accessories, and then Calculator.

I/O Addresses



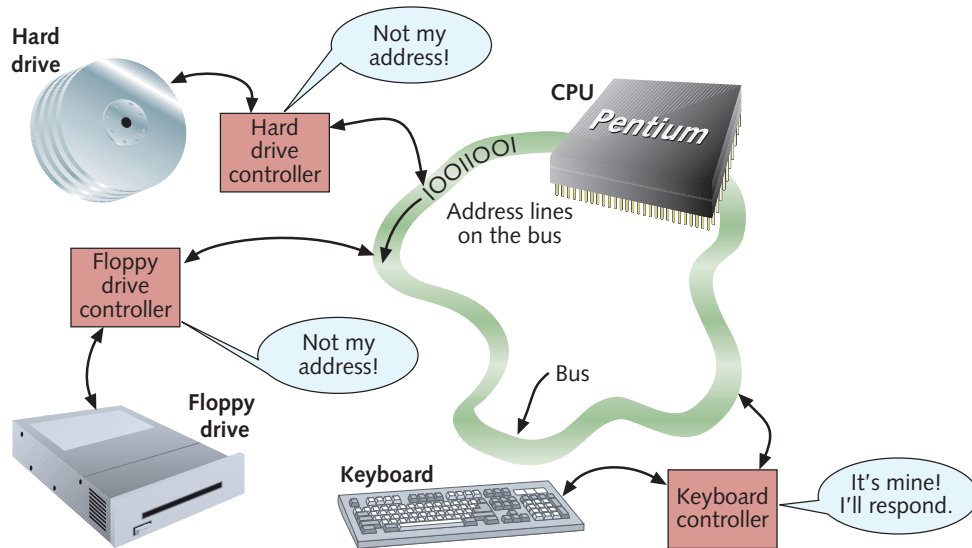
A+ EXAM TIP

Older 16-bit device drivers required a specified range of memory addresses to work, but newer 32-bit drivers don't care what memory addresses they use. Therefore, when considering the system resources used by devices, memory addresses are no longer considered. This is why the A+ Core exam covers only three system resources: IRQs, I/O addresses, and DMA channels.

Another system resource made available to hardware devices is input/output addresses, or I/O addresses. **I/O addresses, or port addresses, sometimes simply called ports, are numbers the CPU can use to access hardware devices, in much the same way it uses memory addresses to access physical memory. The address bus on the motherboard sometimes carries memory addresses and sometimes carries I/O addresses.** If the address bus has been set to carry I/O addresses, then each device "listens" to this bus (see Figure 2-12). If the address belongs to it, then it responds; otherwise, it ignores the request for information. In short, the CPU "knows" a hardware device as a group of I/O addresses. If it wants to know the status of a printer or a floppy drive, for example, it places a particular I/O address on the address bus on the motherboard.

A+
CORE
1.4

2



- ① CPU turns up signal on bus that says, "I/O addresses are on the address line"
- ② All I/O device controllers listen for their addresses
- ③ CPU transmits an I/O address
- ④ The device that "owns" the address responds

Figure 2-12 I/O address lines on a bus work much like an old telephone party line; all devices "hear" the addresses, but only one responds



Refer back to Table 2-2 for a listing of a few common assignments for I/O addresses. Because these addresses are hex numbers, you sometimes see them written with 0x first, such as 0x0040, or with the h last, like this: 0040h.

✓ A+ EXAM TIP

The A+ Core exam expects you to memorize the first I/O address in the I/O address range for the more significant devices. See Table 2-2.

Because IBM made many address assignments when it manufactured the first PC in the late 1970s, common devices such as a hard drive, a floppy drive, or a keyboard use a range of predetermined I/O addresses that never change. Their BIOS is simply programmed to use these standard addresses and standard IRQs. Legacy devices (devices that use older technologies) were designed to use more than one group of addresses and IRQ, depending on how jumpers or DIP switches were set on the device. Newer devices, called Plug and Play devices, can use any I/O addresses or IRQ assigned to them during the boot process. You will learn more about this in Chapter 8.

DMA Channels

A+
CORE
1.4

Another system resource used by hardware and software is a **direct memory access (DMA) channel**, a shortcut method that lets an I/O device send data directly to memory, **bypassing the CPU. A chip on the motherboard contains the DMA logic and manages the process.** Earlier computers had four channels numbered 0, 1, 2, and 3. Later, channels 5, 6, and 7 were added. DMA channel 4 is used as IRQ 2 was used, to connect to the higher IRQs. In Figure 2-13, note that DMA channel 4 cascades into the lower DMA channels.

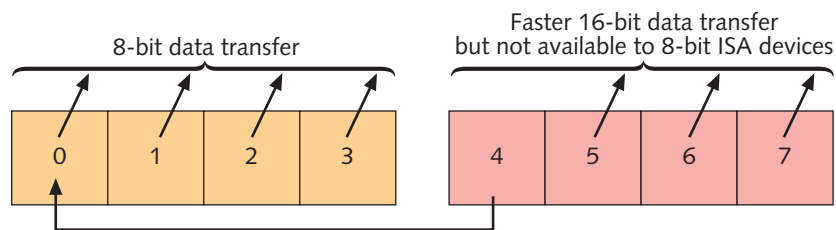


Figure 2-13 DMA channel 4 is not available for I/O use because it is used to cascade into the lower four DMA channels

Some devices, such as a printer, are designed to use DMA channels, and others, such as the mouse, are not. Those that use the channels might be able to use only a certain channel, say channel 3, and no other. Alternately, the BIOS might have the option of changing a DMA channel number to avoid conflicts with other devices. Conflicts occur when more than one device uses the same channel. DMA channels are not as popular as they once were, because their design makes them slower than newer methods. However, slower devices such as floppy drives, sound cards, and tape drives may still use DMA channels.

OS Tools to Examine a System

You have learned about many hardware devices, OS components, and system resources in this chapter. When installing new components or troubleshooting a system, it is important to know how to use OS tools to examine the system. This section discusses several of these tools.

APPLYING CONCEPTS

2

Device Manager

Device Manager under Windows 2000/XP and Windows 9x is the primary tool used to manage hardware devices. (Recall that Windows NT does not have a Device Manager.)

To access Device Manager using Windows XP: Click Start, right-click My Computer, and then select Properties on the shortcut menu. The System Properties dialog box appears. Click the Hardware tab and then click Device Manager.

To access Device Manager using Windows 2000: Right-click the My Computer icon on the desktop, select Properties on the shortcut menu, click the Hardware tab, and then click the Device Manager button.

To access Device Manager using Windows 9x: Right-click the My Computer icon on the desktop, select Properties on the shortcut menu, and then click the Device Manager tab.

The Device Manager dialog box for Windows XP is shown in Figure 2-14. Click a plus sign to expand the view of an item, and click a minus sign to collapse the view. To find out more information about a device, right-click the device and select Properties on the shortcut menu. You can see the Properties dialog box for the video card in Figure 2-14.

When a device is giving problems, check the Properties dialog box of that device for information you can also update the driver for a device, enable or disable a device, change a system resource assigned to a device, and uninstall a device. The steps for performing these tasks are covered in later chapters.

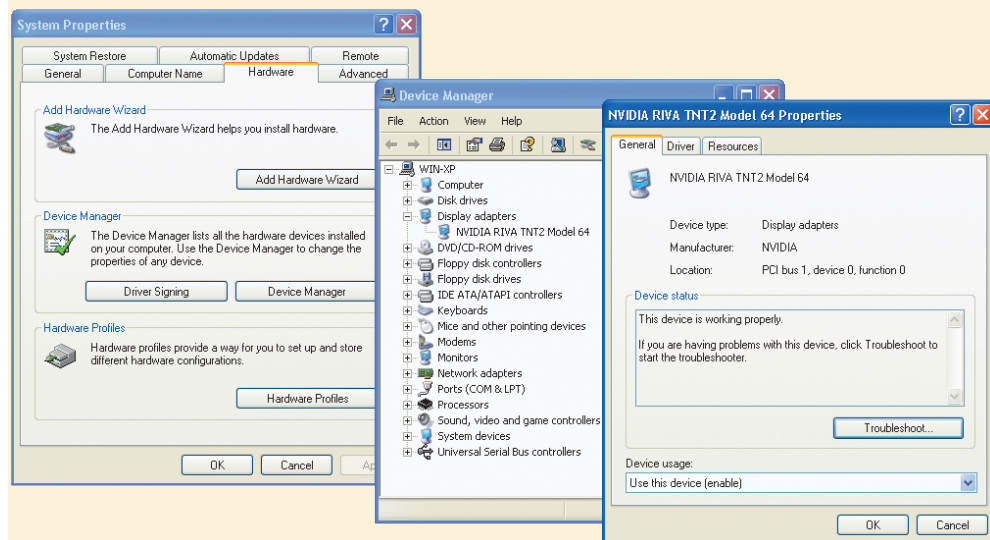


Figure 2-14 Device Manager gives you information about devices

Earlier in the chapter you saw how to use Device Manager to view system resources. You can get a printed report of system information using Device Manager, which can be useful to

document the status of a system. To print the report using Windows XP, click the printer icon on the Device Manager toolbar. There are three options for the report: System summary, Selected class or device, and All devices and system summary.

System Information

The System Information utility gives information similar to that given by Device Manager plus more. For example, it tells you the BIOS version you are using, the directory where the OS is installed, how system resources are used, information about drivers and their status, and much information about software installed on the system that is not included in Device Manager.

To run System Information using Windows 2000/XP or Windows 9x: Click Start, and then click Run. In the Run dialog box, enter `Msiinfo32.exe`, and then click OK. The System Information dialog box appears (see Figure 2-15).

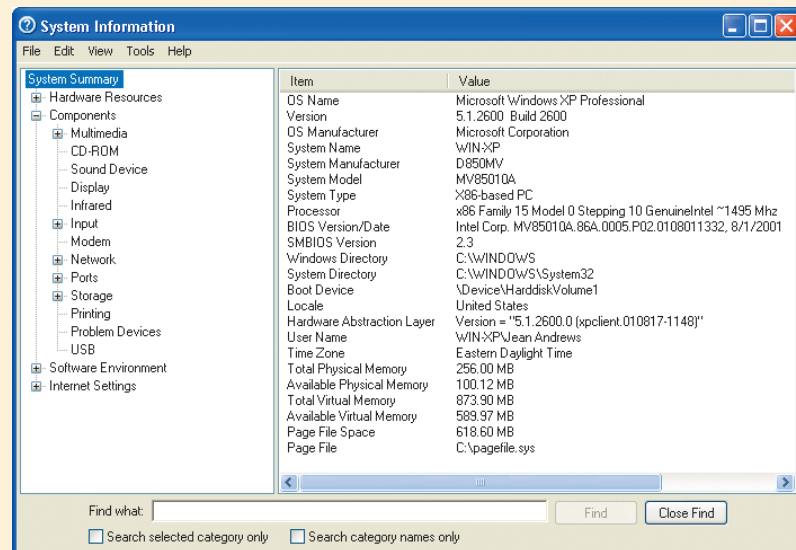


Figure 2-15 System Information gives information about your system that can be useful when troubleshooting

System Information can be useful when a system is having trouble starting. Use it to get a list of drivers that loaded successfully. If you have saved the System Information report when the system was starting successfully, comparing the two reports can help identify the problem device.

Microsoft Diagnostic Utility (MSD)

DOS and Windows 9x offered the Microsoft Diagnostic Utility (MSD), a utility useful for viewing information about the system, including information about memory, video, ports, device drivers, and system *resources*.

To load MSD using Windows 9x: Click Start, click Run, enter MSD.EXE in the Run dialog box, and then click OK. The MSD window appears (see Figure 2-16). You will practice using MSD in a project at the end of this chapter.

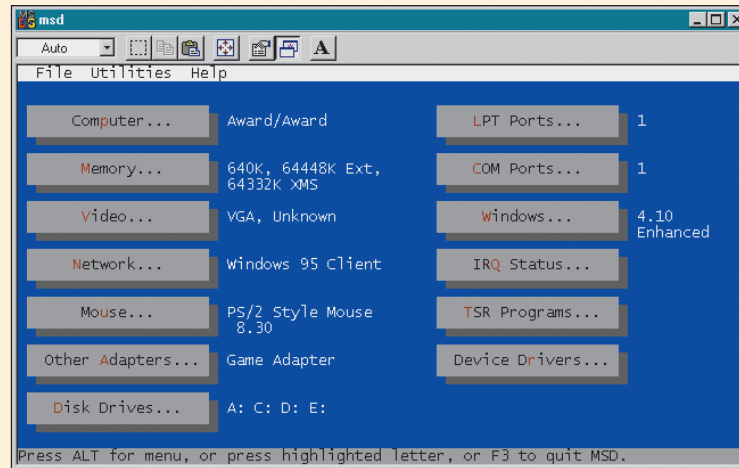


Figure 2-16 MSD opening screen

Booting Up Your Computer

The term **booting** comes from the phrase “lifting yourself up by your bootstraps” and refers to the computer bringing itself up to an operable state without user intervention. Booting refers to either a “soft boot” or “hard boot.” A **hard boot**, or **cold boot**, involves turning on the power with the on/off switch. A **soft boot**, or **warm boot**, involves using the operating system to reboot. For Windows NT/2000/XP and Windows 9x, one way to soft boot is to click Start, click Shut Down, select Restart from the Shut Down menu, and then click OK. For DOS, pressing the three keys Ctrl, Alt, and Del at the same time performs a soft boot.

A hard boot is more stressful on your machine than a soft boot because of the initial power surge through the equipment. Also, a soft boot is faster. Always use the soft boot to restart unless the soft boot method doesn’t work. If you must power down, avoid turning off the power switch and immediately turning it back on without a pause, because this can damage the machine. Some PCs have a reset button on the front of the case. Pressing the reset button starts the boot process at an earlier point than does the operating-system method, and is therefore a little slower, but it might work when the operating-system method fails. For newer motherboards, pressing the reset button is the same as powering off and on, except that there is no stress to the system caused by the initial power surge.

In the next section, you will learn what happens when the PC is first turned on and the startup BIOS takes control and then loads the OS.

Startup BIOS Controls the Beginning of the Boot

A+
CORE
1.1
2.1
4.4

A successful boot process depends on the hardware, the BIOS, and the operating system all performing without errors. If errors occur, they might or might not stall or lock up the boot. Errors are communicated as beeps or as messages onscreen. Appendix A, “Error Messages and Their Meanings,” lists some examples of these messages. The functions performed during the boot can be divided into four parts, as shown in the following list. Startup BIOS is in control for the first step and the beginning of the second step, where control is turned over to the OS.

Here is a brief overview of all four parts before we look at the first two parts in detail. (The last steps depend on the OS being used and are not covered in this book.)

- *Step 1: Startup BIOS runs the power-on self test (POST) and assigns system resources.* The ROM BIOS startup program surveys hardware resources and needs, and assigns system resources to meet those needs (see Figure 2-17). The ROM BIOS startup program begins the startup process by reading configuration information stored in DIP switches, jumpers, and the CMOS chip, and then comparing that information to the hardware—the CPU, video card, disk drive, hard drive, and so on. Some hardware devices have BIOSs of their own that request resources from startup BIOS, which attempts to assign these system resources as needed.
- *Step 2: The ROM BIOS startup program searches for and loads an OS.* Most often the OS is loaded from logical drive C on the hard drive. Configuration information on the CMOS chip tells startup BIOS where to look for the OS. Most new BIOSs support loading the OS from the hard drive, a floppy disk, a CD, or a Zip drive. The BIOS turns to that device, reads the beginning files of the OS, copies them into memory, and then turns control over to the OS. This part of the loading process works the same for any operating system; only the OS files being loaded change.
- *Step 3: The OS configures the system and completes its own loading.* The OS checks some of the same things that startup BIOS checked, such as available memory and whether that memory is reliable. Then the OS loads the software to control a mouse, CD-ROM, scanner, and other peripheral devices. These devices generally have device drivers stored on the hard drive. The Windows desktop is loaded.

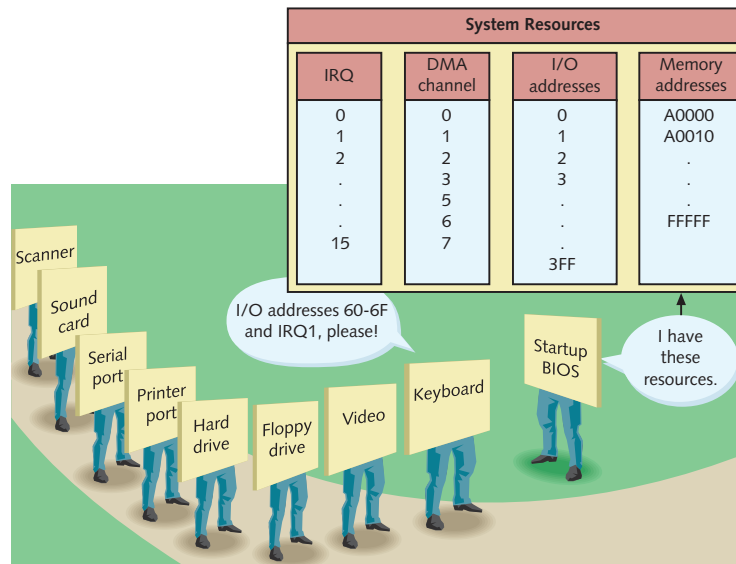


Figure 2-17 Boot Step 1: ROM BIOS startup program surveys hardware resources and needs and assigns system resources to satisfy those needs

- *Step 4: Application software is loaded and executed.* Sometimes an OS is configured to automatically launch application software as part of the boot. When you tell the OS to execute an application, the OS first must find the application software on the hard drive, CD-ROM, or other secondary storage device, copy the software into memory, and then turn control over to it. Finally, you can command the application software, which makes requests to the OS, which, in turn, uses the system resources, system BIOS, and device drivers to interface with and control the hardware. At this point, the user is in control.

Let's look a bit more closely at the first step, beginning with the POST.

POST and Assignment of System Resources

When you turn on the power to a PC, the CPU begins the boot by initializing itself and then turning to the ROM BIOS for instructions. The ROM BIOS then performs POST. Listed below are the key steps in this process.

- When the power is first turned on, the system clock begins to generate clock pulses.
- The CPU begins working and initializes itself (resetting its internal values).
- The CPU turns to memory address FFFF0h, which is the memory address always assigned to the first instruction in the ROM BIOS startup program.
- This instruction directs the CPU to run the POST tests.
- POST first checks the BIOS program operating it and then tests CMOS RAM.

- A test determines that there has not been a battery failure.
- Hardware interrupts are disabled (this means that pressing a key on the keyboard or using another input device at this point will not affect anything).
- Tests are run on the CPU, and it is further initialized.
- A check determines if this is a cold boot. If so, the first 16K of RAM is tested. Hardware devices installed on the computer are inventoried and compared to configuration information.
- Video, memory, keyboard, floppy disk drives, hard drives, ports, and other hardware devices are tested and configured, and IRQ, I/O addresses, and DMA assignments are made. The OS will later complete this process.
- Some devices are set up to go into “sleep mode” to conserve electricity.
- The DMA controller is checked.
- The interrupt controller is checked.
- CMOS setup (a BIOS program to change CMOS configuration data) is run if requested.
- BIOS begins its search for an OS.

During POST, before the CPU has checked the video system, errors encountered up to this point are communicated by beeps. Short and long beeps indicate an error; the coding for the beeps depends on the BIOS. After POST checks and verifies the video controller card (note that POST does not check to see if a monitor is present or working), POST can use the monitor to display its progress. After checking video, POST checks RAM by writing and reading data. A running count of RAM is displayed on the monitor during this phase.

Next, the keyboard is checked. With some BIOSs, if you press and hold any keys at this point, an error occurs. Secondary storage, including floppy disk drives and hard drives, is also checked. The hardware that POST finds is checked against the data stored in the CMOS chip, jumpers, and/or DIP switches to determine if they agree.

System resources (IRQ, I/O addresses, DMA channel, or memory addresses) required by a device are assigned. Think of the process as a dialog: The startup BIOS recognizes that a hardware device is present. The BIOS asks the device, “What resources do you need?” The device says, “I need this IRQ, these I/O addresses, this DMA channel, and this many memory addresses.” For legacy hardware, a device is the sole owner of these resources, and problems occur when more than one device attempts to use the same resource. Today, more flexible Plug and Play devices simply say, “I need one IRQ, some I/O addresses, and this many memory addresses for my BIOS. Please tell me the resources I can use.” Also, for devices using an ISA slot, the resources are assigned to the device. For devices using PCI slots, the resources are assigned to the slot, and the PCI controller can sometimes share these resources among more than one slot.

BIOS first enables the devices that are not Plug and Play, and then tries to make the Plug and Play devices use the leftover resources. BIOS then turns this information over to Windows when it loads, which completes the assignment of resources.

How the BIOS Finds and Loads the OS

Once POST and the first pass at assignment of resources is complete, the next step is to load an OS. Startup BIOS looks to CMOS setup to find out which device is set to be the boot device. Most often the OS is loaded from logical drive C on the hard drive. See Figure 2-18. The minimum information required on the hard drive to load an OS is listed below. You can see some of these items labeled in Figure 2-19.

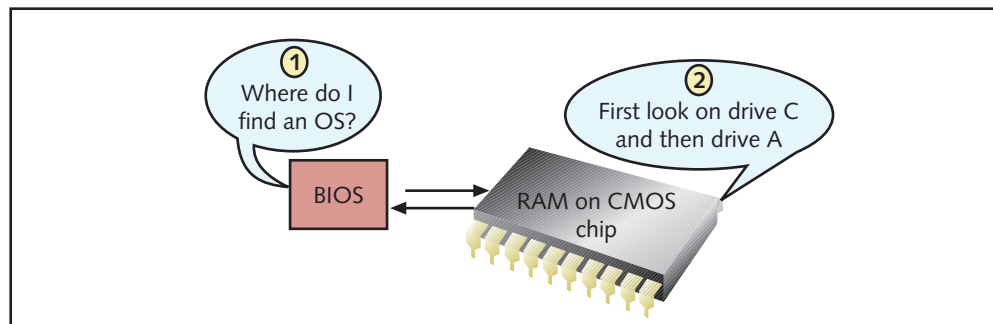


Figure 2-18 BIOS uses CMOS RAM to know where to look to find an OS

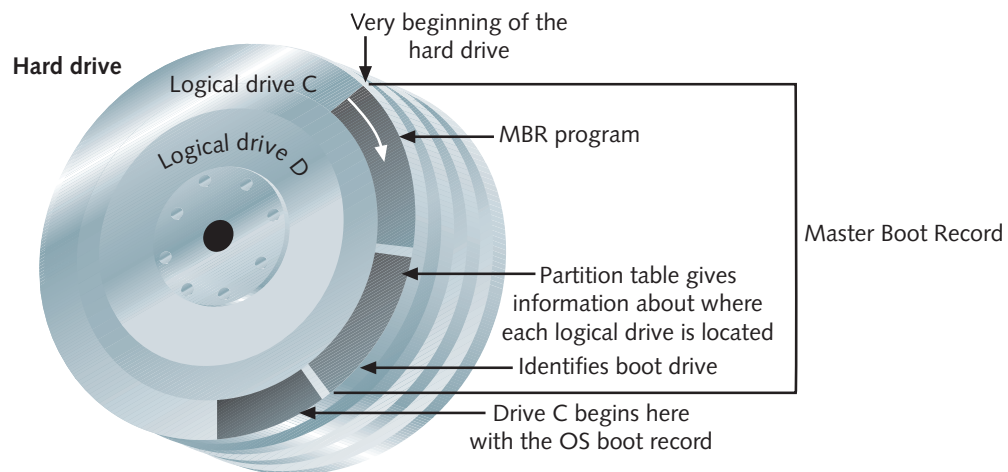


Figure 2-19 A hard drive might contain more than one logical drive; the partition table at the beginning of the drive contains information about the location of each logical drive, indicates which drive is the boot drive, and holds the master boot program that begins the process of loading an operating system

- A small segment (512 bytes) of information at the very beginning of the hard drive is called the Master Boot Record (MBR); it contains two items. The first item is the master boot program, which is needed to locate the beginning of the OS on the drive.
- The second item in the MBR is a table that contains a map to the logical drives on the hard drive and indicates which drive is the boot drive. This table is called the **partition table**.
- At the beginning of the boot drive (usually drive C) is the OS boot record, which loads the first program file of the OS. For Windows NT/2000/XP, that program is Ntldr, and for Windows 9x, that program is Io.sys.
- The boot loader program for the OS (Ntldr or Io.sys) begins the process of loading the OS into memory.

Often a hard drive is divided or partitioned into more than one logical drive—for example, drive C and drive D—as shown previously in Figure 2-19. Whether a hard drive has one or several logical drives, it always contains a single partition table, located at the very beginning of the drive, which tells BIOS how many partitions the drive has and how each partition is divided into one or more logical drives, which partition contains the drive to be used for booting (called the active partition), and where each logical drive begins and ends. All this is covered in more detail in Chapter 7.

The master boot program at the beginning of the table is used to start the boot process from the hard drive. One logical drive on the hard drive is designated as the boot drive, and the OS is stored on it. At the beginning of this logical drive is the OS boot record, which knows the names of the files that contain the core programs of the OS.

The process for BIOS to load the OS begins with BIOS looking to CMOS setup to find out which secondary storage device should contain the OS. Setup might instruct the BIOS to first look to drive C, and, if no OS is found there, then to try drive A; or the order might be A then C. If BIOS looks first to drive A and does not find a disk in the drive, it turns to drive C. If it first looks to drive A and finds a disk in the drive, but the disk does not contain the OS, then the following error message or a similar one is displayed:

Non-system disk or disk error, press any key

You must replace the disk with one that contains the OS or simply remove the disk to force the BIOS to continue to drive C to find the OS.

Troubleshooting a Failed Boot

As a PC hardware technician, you need to know how to troubleshoot problems with a failed boot when these problems are caused by hardware. (Troubleshooting a failed boot caused by problems with the OS is beyond the scope of this book.) If an error message indicates there is a problem with the hard drive, use a bootable floppy disk to boot the system. A bootable disk is a floppy disk that has enough of an OS installed to boot the system. Windows 9x uses a single bootable disk, but Windows

2000 uses a set of four disks. How you create these disks is specific to each OS. A project at the end of this chapter shows you how to create a Windows 9x startup disk. Although Windows XP allows you to create a bootable disk, you can do little with it except verify that the system boots without the use of the hard drive. In Chapter 7, you will learn how to troubleshoot a hardware problem with the hard drive subsystem, regardless of the OS installed.



For a listing of error messages and beep codes that occur during the boot and what to do about them, see Appendix A, "Error Messages and Their Meanings."

CHAPTER SUMMARY

- ▶ An operating system (OS) is software that controls a computer. It manages hardware, runs applications, provides an interface for users, and stores, retrieves, and manipulates files.
- ▶ Some types of software are BIOS, device drivers, operating systems (OSs), and application software.
- ▶ Application software relates to the OS, which relates to BIOS and device drivers to control hardware.
- ▶ ROM BIOS programs are used to start the computer (startup BIOS), manage simple devices (system BIOS), and make changes to CMOS (CMOS setup).
- ▶ Windows 2000/XP does not claim backward compatibility with older hardware devices. To know if a device will work under a certain OS, check the HCL (hardware compatibility list).
- ▶ Four system resources that aid the communication between hardware and software are I/O addresses, IRQs, DMA channels, and memory addresses.
- ▶ An IRQ is a line on a bus that a device uses to alert the CPU that it needs servicing.
- ▶ A DMA channel provides a shortcut for a device to send data directly to memory, bypassing the CPU.
- ▶ Memory addresses are numbers assigned to RAM and ROM so that the CPU can access both.
- ▶ The CPU sends a device's I/O address over the address bus when it wants to initiate communication with the device.
- ▶ COM and LPT are preconfigured assignments of system resources that a device can use. For example, COM1 is IRQ 4 and I/O addresses 03F8 through 03FF.

- ▶ A hardware interrupt is initiated by a hardware device sending an IRQ to the CPU.
- ▶ To know how IRQs are assigned on your computer, use Device Manager.
- ▶ Software manages memory by means of memory addresses that point to locations in RAM. The number of memory addresses is partly limited by the number of wires on the bus devoted to these addresses.
- ▶ RAM and ROM BIOS on the motherboard and other circuit boards need memory addresses assigned to the BIOS so the CPU can access these programs.
- ▶ When a PC is first turned on, the startup BIOS is in control. It later loads the OS and then turns control over to it.
- ▶ Error messages during the boot are communicated by beeps or by messages displayed on the screen.
- ▶ Startup BIOS performs a power-on self test (POST) that surveys and tests hardware, examines setup information, and assigns system resources to the hardware. Startup BIOS then begins the process of loading the OS.
- ▶ When the OS loads from a hard drive, the first program BIOS executes is the Master Boot Record (MBR) program, which executes the OS boot record program. This OS boot loader, in turn, loads the first program file of the OS. Windows NT/2000/XP attempts to find Ntldr, and Windows 9x and DOS attempt to find Io.sys on the hard drive.

KEY TERMS

For explanations of key terms, see the Glossary near the end of the book.

booting
CMOS setup
cold boot
device driver
direct memory access (DMA) channel
hard boot
hardware compatibility list (HCL)
hardware interrupt

I/O addresses
interrupt request (IRQ) line
memory addresses
operating system (OS)
partition table
polling
port addresses
power-on self test (POST)

shadow RAM
shadowing ROM
soft boot
startup BIOS
system BIOS
system resource
warm boot

REVIEWING THE BASICS

2

1. List the four major functions of an operating system.
2. List three well-known OSs.
3. List two OSs written for personal computers that are not made by Microsoft.
4. What are the three general types of programs stored on the ROM BIOS chip of a motherboard?
5. Name one device that is usually connected to a PC after its device drivers are installed.
6. Where is the most convenient and best place to go to obtain an update for a device driver?
7. What is the name of the list of devices that are supported by Windows 2000/XP?
8. When a device is *not* configured in CMOS setup, does the OS relate to the device by way of the BIOS or device drivers?
9. What are two Windows utilities you can use to view the name of a driver controlling a device?
10. The practice of copying ROM instructions into RAM in order to improve performance is called ____.
11. What is the IRQ of the system timer?
12. What IRQ does COM2 use?
13. Which DMA channel is used to cascade into the lower four DMA channels?
14. List four system resources that software uses to manage hardware.
15. What Windows 9x utility allows you to see the IRQ assignments made to devices?
16. How is a hardware interrupt initiated?
17. If memory addresses are used by the CPU to access memory, then what are I/O addresses used for?
18. What is the I/O address range for the keyboard?
19. Why are DMA channels not as popular as they once were with high-speed devices?
20. Name a device that uses polling in order to be serviced by the CPU.
21. What is the name of the first 512 bytes on a hard drive?
22. What is the name of the table at the beginning of a hard drive that tells where logical drives are stored on the hard drive?

23. What is the name of the Windows 2000/XP file stored on a hard drive that begins the process of loading the OS?
24. What is the purpose of Io.sys in Windows 98?
25. How many floppy disks does Windows 2000 require to boot the system when you cannot boot from the hard drive?

THINKING CRITICALLY

1. Name one system resource that a video card most likely will not need.
2. Is a mouse more likely to be controlled by a device driver or by system BIOS?
3. Name one device that is likely to be controlled by system BIOS.
4. If your printer is giving you trouble, what is the best way to obtain an update for the device driver?
5. When you boot your PC, you hear six beeps and the system halts. You cannot find any documentation to tell you what the six beeps mean. What do you do next?



HANDS-ON PROJECTS

PROJECT 2-1: Observing the Boot Process

1. If your computer has a reset button, press it and then watch what happens. If your computer does not have a reset button, turn off the computer, wait a few seconds, and then turn it back on. Write down every beep, light on/off, and message on the screen that you notice. Compare your notes to those of others to verify that you are not overlooking something.
2. Answer these questions from observing the boot:
 - a. What type of video card is the computer using?
 - b. Who is the BIOS vendor, and what version of the BIOS is the computer using?
 - c. As the computer boots, memory is counted. Observe the memory count and record the amount of memory detected. What number system is used to count this memory?

3. Unplug the keyboard and reboot. What is different about the boot? Write down your observations.
4. Plug the keyboard back in, unplug the monitor, and reboot. After you reboot, plug the monitor in. Did the computer know the monitor was missing?
5. Put a floppy disk that is not bootable in drive A and reboot. Write down what you observe. If the PC booted to the desktop as usual, why didn't it look to the floppy disk to load the OS?

PROJECT 2-2: Using the Windows 2000/XP System Information Utility

Windows 2000/XP has a System Information utility that gives you detailed information about your system. Using Windows 2000/XP, do the following to run the System Information utility and gather information about your system:

1. Click **Start**, click **Run**, and then type **Minfo32.exe** in the Run dialog box. Click **OK**. The System Information dialog box appears.
2. Browse through the different levels of information in this window and answer the following questions:
 - a. What OS and OS version are you using?
 - b. What is your CPU speed?
 - c. What is your BIOS manufacturer and version?
 - d. How much RAM is installed on your video card? Explain how you got this information.
 - e. What is the name of the driver file that manages your parallel port? Your serial port?
 - f. How is IRQ 10 used on your system? IRQ 4?
 - g. Which DMA channels are used on your system and how are they used?

PROJECT 2-3: Using a Freeware Diagnostic Utility

You can download many freeware diagnostic utilities from the Internet and use them to examine, troubleshoot, and benchmark a system. Do the following to download and use one utility to examine your system:

1. Go to the CNT Networks Web site at www.cnet.com and download the latest version of Fresh Diagnose. Web sites change often, but at the time of this writing, you would click **Downloads**, then **Utilities & Drivers**, and then **Fresh Diagnose 6.0**. Save the utility to a folder on your hard drive named **Downloads**.
2. Double-click the file to execute the program and install the software. When given the opportunity, choose to create a shortcut to the software on your desktop.
3. Click the shortcut to run the Fresh Diagnose program.
4. Browse through the Fresh Diagnose menus and answer the same questions listed in Project 2-2 for the Windows 2000/XP System Information utility.
5. Compare the two programs, Fresh Diagnose and System Information, by answering the following questions:
 - a. Which product is easier to use and why?
 - b. Which product gives more information about your system?
 - c. What is one advantage that System Information has over Fresh Diagnose?
 - d. What is one advantage that Fresh Diagnose has over System Information?
 - e. Which product do you prefer and why?

PROJECT 2-4: Using Microsoft Diagnostics with Windows 9x

DOS and Windows offer the Microsoft Diagnostics utility. This utility examines your system, displaying useful information about ports, devices, memory, and the like. You can find the MSD.EXE utility in the \TOOLS\OLDMSDOS directory on your Windows 9x installation CD. Using Windows Explorer, copy the utility to your hard drive and store it in a folder named \Tools.

For Windows 9x, boot your PC to an MS-DOS prompt in real mode. To boot into real mode, press **Ctrl** or **F8** as you start the system. The Windows Startup menu appears. Select **Command prompt only**. From the DOS prompt, execute this command: **C:\TOOLS\MSD**. You should see a screen similar to that in Figure 2-16.

Browse carefully through all menu options of this utility, and answer the following questions about your system:

1. List the following information or print the appropriate MSD screen that contains it: manufacturer; version number; and date of your system BIOS, video BIOS, and mouse device driver.
2. What kind of video card is installed?
3. How much memory is currently installed on this PC?
4. What version of the OS is the PC running?
5. What CPU is the PC using?

Exit MSD. Save the information you noted to compare with the information that you will obtain from MSD in Windows 2000.

You need Windows 2000 installed on a PC to do the rest of this project:

1. Copy the **MSD.exe** program to a folder on your Windows 2000 PC named **\Tools**.
2. From within Windows 2000, open a command prompt. (Click **Start**, **Programs**, **Accessories**, and **Command Prompt**.)
3. From the command prompt, start MSD using this command: **\Tools\MSD**.
4. Browse through all menu options, and answer the same questions about your system as you did for Windows 9x.

PROJECT 2-5: Using Device Manager

Using Device Manager under Windows 2000/XP or Windows 9x, answer the following questions about your computer. To access Device Manager using Windows 2000/XP, open the System Properties dialog box, click the **Hardware** tab, and then click **Device Manager**. For Windows 9x, right-click the **My Computer** icon on the desktop, select **Properties** on the shortcut menu, and click the **Device Manager** tab.

1. Does your computer have a network card installed? If so, what is the name of the card?
2. What three settings can you change under Device Manager?
3. What are all the hardware devices that Device Manager recognizes as present?

PROJECT 2-6: Examining CMOS Setup

Boot your PC and look for directions on the screen that tell you how to access CMOS setup on your PC, such as “Press Del to enter setup.” Access CMOS setup and answer these questions:

1. What keystroke(s) did you use to access CMOS setup?
2. What BIOS does your motherboard use? (Include brand and version.)
3. List the different CMOS setup windows that you can access from the CMOS main menu window.
4. Access the window that gives information about serial ports. What is the name of that window?
5. What I/O addresses and IRQ does the first serial port use?
6. What I/O addresses and IRQ does the first parallel port use?
7. What are the system date and system time reported by CMOS setup?
8. What keys do you press to exit CMOS setup without saving any changes?

Exit CMOS setup without saving any changes you might have accidentally made. Your PC should then boot to the Windows desktop.

PROJECT 2-7: Creating and Using a Windows 9x Startup Disk

Complete the following steps to create and use a startup disk for Windows 9x:

1. Click **Start**, point to **Settings**, and then click **Control Panel**.
2. In the Control Panel window, double-click the **Add/Remove Programs** icon.
3. Click the **Startup Disk** tab, and then click the **Create Disk** button (see Figure 2-20).
4. Windows might need the Windows CD to create the disk. Insert the CD if it is requested. The startup disk will then be created.
5. Label this bootable disk “Windows 98 Startup Disk” and keep it for future projects.

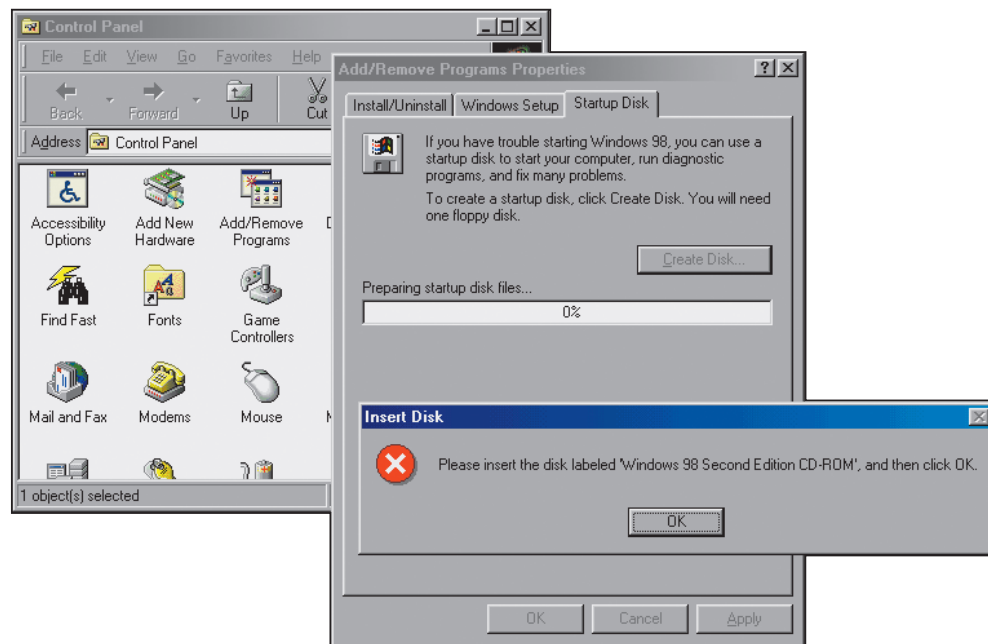


Figure 2-20 Windows might use the Windows CD to create a startup disk

Complete the following steps to use the Windows 9x startup disk:

1. Boot the Windows 98 computer with the startup disk in the floppy disk drive. What message do you see on the screen during the boot?
2. If you have access to a Windows 2000 or Windows XP computer, use the startup disk to boot the computer. Does the boot process proceed differently on this computer?

