Operating Systems

When a brand-new computer comes off the factory assembly line, it can do nothing. The hardware needs software to make it work. Are we talking about applications software such as word processing or spreadsheet software? Partly.

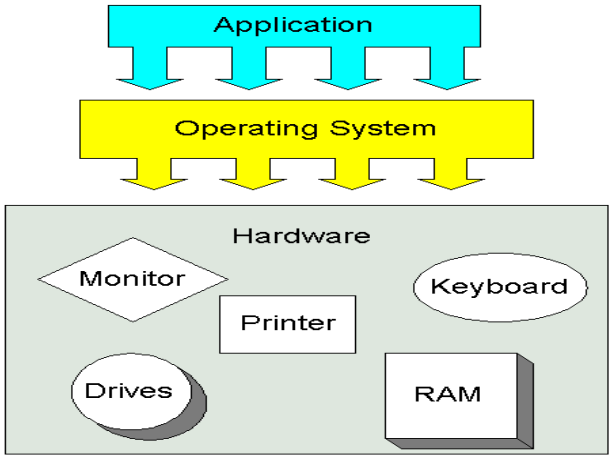


Figure 1: The Operating System in a Hierarchy

But an applications software package does not communicate directly with the hardware. As shown in Figure 1, between the applications software and the hardware is a software interface - an operating system. An operating system is a set of programs that lies between applications software and the computer hardware. Conceptually the operating system software is an intermediary between the hardware and the applications software. Incidentally, the term system software is sometimes used interchangeably with operating system, but system software means all programs related to coordinating computer operations. System software does include the operating system, but it also includes the BIOS software, drivers, and service programs, which we will discuss briefly in this chapter (see Figure 2).

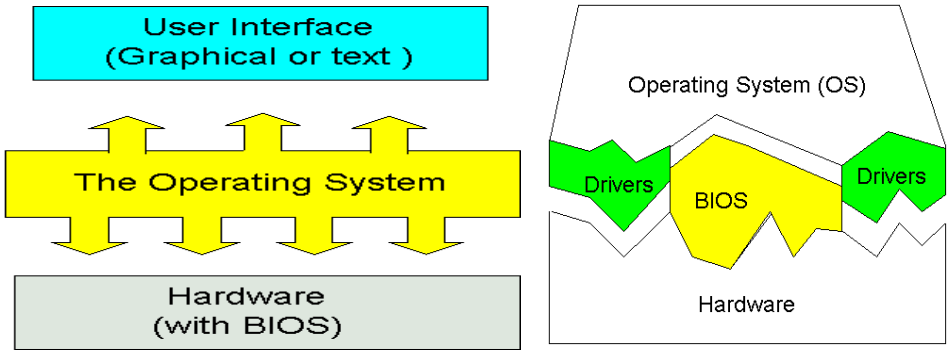


Figure 2: System Software

Note that we said that an operating system is a set of programs. The most important program in the operating system, the program that manages the operating system, is the supervisor program, most of which remains in memory and is thus referred to as resident. The supervisor controls the entire operating system and loads into memory other operating system programs (called non-resident) from disk storage only as needed.

An operating system has three main functions: (1) manage the computer's resources, such as the central processing unit, memory, disk drives, and printers, (2) establish a user interface, and (3) execute and provide services for applications software. Keep in mind, however, that much of the work of an operating system is hidden from the user; many necessary tasks are performed behind the scenes. In particular, the first listed function, managing the computer's resources, is taken care of without the user being aware of the details. Furthermore, all input and output operations, although invoked by an applications program, are actually carried out by the operating system. Although much of the operating system functions are hidden from view, you will know when you are using an applications software package, and this requires that you invoke-call into action-the operating system. Thus, we both establish a user interface and execute software.

Operating systems for mainframe and other large computers are even more complex because they must keep track of several programs from several users all running in the same time frame. Although some personal computer operating systems-most often found in business or learning environments-can support multiple programs and users, most are concerned only with a single user. We begin by focusing on the interaction between a single user and a personal computer operating system.

Operating Systems for Personal Computers:

If you peruse software offerings at a retail store, you will generally find the software grouped according to the computer, probably IBM (that is, IBM compatible) or Macintosh, on which the software can be used. But the distinction is actually finer than the differences among computers: Applications software-word processing, spreadsheets, games, whatever-are really distinguished by the operating system on which the software can run.

Generally, an application program can run on just one operating system. just as you cannot place a Nissan engine in a Ford truck, you cannot take a version of WordPerfect designed to run on an IBM machine and run it on an Apple Macintosh. The reason is that IBM personal computers and others like them have Intel-compatible microprocessors and usually use Microsoft's operating system, called MS-DOS (for Microsoft disk operating system) on older computers, and Windows95 or Windows98 on more modern computers. Computers that have come out since the year 2000 often come with Windows ME (Millennium Edition), or Windows2000. Macintoshes use an entirely different operating system, called the Macintosh operating system, which is produced by Apple. Over 75 percent of personal computers use a version of Windows as their operating systems. Macintosh comprises about 15 percent of the market, with other operating systems such as Linux comprising the rest.

Users do not set out to buy operating systems; they want computers and the applications software to make them useful. However, since the operating system determines what software is available for a given computer, many users observe the high volume of software available for MS-DOS machines and make their computer purchases accordingly. Others prefer the user-friendly style of the Macintosh operating system and choose Macs for that reason.

Although operating systems differ, many of their basic functions are similar. We will show some of the basic functions of operating systems.

MS-DOS:

Most users today have a computer with a hard disk drive. When the computer is turned on, the operating system will be loaded from the hard drive into the computer's memory, thus

making it available for use. The process of loading the operating system into memory is called bootstrapping, or booting the system. The word booting is used because, figuratively speaking, the operating system pulls itself up by its own bootstraps. When the computer is switched on, a small program (in ROM-read-only memory) automatically pulls up the basic components of the operating system from the hard disk. From now on, we will refer to MS-DOS by its commonly used abbreviated name, DOS, pronounced to rhyme with boss.

The net observable result of booting DOS is that the characters C> (or possibly C:\>) appear on the screen. The C refers to the disk drive; the > is a prompt, a signal that the system is prompting you to do something. At this point you must give some instruction to the computer. Perhaps all you need to do is key certain letters to make the application software take the lead. But it could be more complicated than that because C> is actually a signal for direct communication between the user and the operating system.

Although the prompt is the only visible result of booting the system, DOS also provides the basic software that coordinates the computer's hardware components and a set of programs that lets you perform the many computer system tasks you need to do. To execute a given DOS program, a user must issue a command, a name that invokes a specific DOS program. Whole books have been written about DOS commands, but we will consider just a few that people use for ordinary activities. Some typical tasks you can do with DOS commands are prepare (format) new diskettes for use, list the files on a disk, copy files from one disk to another, and erase files from a disk.

Microsoft Windows:

Microsoft Windows started out as a shell. Windows uses a colourful graphics interface that, among other things, eases access to the operating system. The feature that makes Windows so easy to use is a graphical user interface (GUI-pronounced "goo-ee"), in which users work with on-screen pictures called icons and with menus rather than with keyed-in. They are called pull-down menus because they appear to pull down like a window shade from the original selection. Some menus, in contrast, called pop-up menus originate from a selection on the bottom of the screen. Furthermore, icons and menus encourage pointing and clicking with a mouse, an approach that can make computer use both fast and easy.

To enhance ease of use, Windows is usually set up so that the colourful Windows display is the first thing a user sees when the computer is turned on. DOS is still there, under Windows, but a user need never see C> during routine activities. The user points and clicks among a series of narrowing choices until arriving at the desired software.

Although the screen presentation and user interaction are the most visible evidence of change, Windows offers changes that are even more fundamental. To understand these changes more fully, it is helpful at this point to make a comparison between traditional operating systems for large computers and Windows.

In addition to adding a friendly GUI, Windows operating systems added another important feature to DOS - multi-tasking. Multi-tasking occurs when the computer has several programs executing at one time. PCs that ran under DOS could only run one program at a time. Windows-based computers can have multiple programs (e.g. a browser, a word processor, and several Instant Messaging instances) running at the same time. When programs are executing at the same time, they are said to be executing concurrently.

As we learned, personal computers have only one CPU that handles just one instruction at a time. Computers using the MS-DOS operating system without a shell are limited not only to just one user at a time but also to just one program at a time. If, for example, a user were

using a word processing program to write a financial report and wanted to access some spreadsheet figures, he or she would have to perform a series of arcane steps: exit the word processing program, enter and use and then exit the spreadsheet program, and then re-enter the word processing program to complete the report. This is wasteful in two ways: (1) the CPU is often idle because only one program is executing at a time, and (2) the user is required to move inconveniently from program to program.

Multi-tasking allows several programs to be active at the same time, although at an instant in time the CPU is doing only one instruction for one of the active programs. The Operating System manages which instructions to send to the CPU. Since computers are so fast, the operating system can switch the program that gets to execute on the CPU so quickly, the user can not tell. This is what allows your computer to be "listening" for incoming instant messages, for instance, while you use a word processor to write a paper.

How The Operating System Works:

When you turn on your computer, it's nice to think that you're in control. There's the trusty mouse, which you can move anywhere on the screen, summoning up your music library or internet browser at the slightest whim. Although it's easy to feel like a director in front of your desktop or laptop, there's a lot going on inside, and the real person behind the curtain handling the necessary tasks is the operating system.

Microsoft windows powers most of the computers we use for work or personal use. Macintosh computers come pre-loaded with macOS. Linux and UNIX operating systems are popular for digital content servers, but many distributions or distros, have become increasingly popular for everyday use. Regardless of your choice, without an operating system, you're not going to get anything done.

Other devices have their own operating systems. Google's Android and Apple's iOS are the most common smartphone OSes as of the 2020s, although some manufacturers have developed their own, mostly based on the Android operating system. Apple ships iPads with iPadOS, Apple watches with watch OS and Apple TV uses tvOS. And there are all kinds of other devices that have their own operating systems — think Internet of Things devices, smart TVs and the systems that run car infotent system. And that doesn't even include the complex system needed. in self-driving cars.

The purpose of an operating system is to organize and control hardware and software so that the device it lives in behaves in a flexible but predictable way. In this article, we'll tell you what a piece of software must do to be called an operating system, show you how the operating system in your desktop computer works and give you some examples of how to take control of the other operating systems around you.

What Is an Operating System?

A Windows 11 logo is seen on a smartphone screen with a Microsoft website in the background. Windows is probably the most common operating system.

Not all computers have operating systems. The computer that controls the microwave oven in your kitchen, for example, doesn't need an operating system. It has one set of tasks to perform, very straightforward input to expect (a numbered keypad and a few pre-set buttons) and simple, never-changing hardware to control. For a machine like this, an elaborate operating system would be unnecessary baggage, driving up the development and manufacturing costs significantly and adding complexity where none is required. Instead, the computer in a microwave oven simply runs a single hard-wired program called an embedded system all the time.

For other devices, an operating system creates the ability to:

Serve a variety of purposes.

Interact with users in more complicated ways.

Keep up with needs that change over time.

All desktop computers have operating systems. The most common are the Windows family of operating systems developed by Microsoft, the Macintosh operating systems developed by Apple and the UNIX family of operating systems developed by a whole history of individuals, corporations and collaborators. There are hundreds of other operating systems available for special-purpose applications, including specializations for mainframes, robotics, manufacturing, real-time control systems and so on.

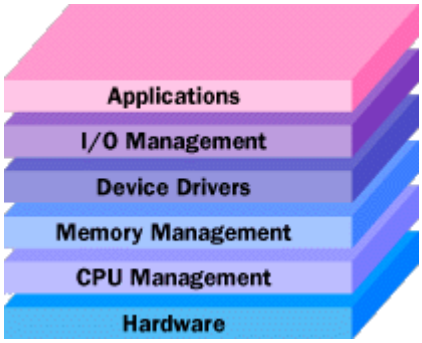
In any device that has an operating system, there's usually a way to make changes to how the device works. This is far from a happy accident; one of the reasons operating systems use portable code rather than permanent physical circuits is so that they can be changed or modified without having to scrap the whole device.

For a desktop computer user, this means you can add a new security update, system patch, new application or even an entirely new operating system rather than junk your computer and start again with a new one when you need to make a change. As long as you understand how an operating system works and how to get at it, in many cases you can change some of the ways it behaves.

Operating System Functions

At the simplest level, an operating system does two things:

1. It manages the hardware and software resources of the system. In computers, tablets and smartphones these resources include the processors, memory, disk space and more.
2. It provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware.



The operating system controls every task your computer carries out and manages system resources to optimize performance.

The first task, managing the hardware and software resources, is very important, as various programs and input methods compete for the attention of the Central Processing Unit (CPU) and demand memory, storage and input/output (I/O) bandwidth for their own purposes. In this capacity, the operating system plays the role of the good parent, making sure that each application gets its necessary resources while playing nicely with all the other applications, as well as husbanding the limited capacity of the system to the greatest good of all the users and applications.

The second task, providing a consistent user interface, is especially important if there is more than one of a particular type of computer using the operating system, or if the hardware making up the computer is ever open to change. A consistent application programming interface (API) allows a software developer to write an application on one computer and have a high level of confidence that it will run on another computer of the same type, even if the amount of memory or the quantity of storage is different on the two machines.

Even when a particular computer is unique, an operating system ensures that applications continue to run when hardware upgrades and updates occur. This is because the operating system — not the application — is charged with managing the hardware and the distribution of its resources. One of the challenges facing developers is keeping their operating systems flexible enough to run hardware from the thousands of vendors manufacturing computer equipment. Today's systems can accommodate thousands of different printers, disk drives and special peripherals in any possible combination.