OPERATING SYSTEMS

Programs you buy or write, in order to operate properly, need additional instructions. They are found in the computer's *operating system.* The operating system is in charge of the fundamental system operations.

The operating system manages the hardware resources. For example, the operating system allocates memory to programs and system operations. It also can allocate processor time in situations where multiple programs are running.

The operating system maintains the system of files on disks and other means of secondary storage. Programs and files are organized into directories by the operating system.

The operating system also controls input and output operations. Keyboard input, mouse movements, displaying to the screen, and printing all involve the operating system.

Finally, the operating system loads programs and supervises their execution. When you issue a command to start a program, the operating system loads the program into memory and allows it to begin executing. The operating system regularly interrupts the program so that housekeeping chores like updating the system date and time can take place.

Some operating systems you may have seen or used are MS-DOS (the operating system written by Microsoft for PCs), OS/2, Windows, and the Macintosh operating system (MacOS).

The DOS prompt, shown in Figure 2-11 is actually a program that is part of the MS-DOS operating system.

Microsoft Windows, OS/2, and the Macintosh use graphical user interfaces as a control center from which programs are loaded. A *graphical user interface* is a system for interacting with the computer user through pictures. Graphical user interface is often abbreviated as GUI, pronounced "gooey."

The Macintosh operating system includes a program called the Finder (Figure 2-12) that organizes and loads programs and documents. The Macintosh

```
C:\>
```

F I G U R E   2 - 1 1
The DOS prompt is part of the MS-DOS operating system.

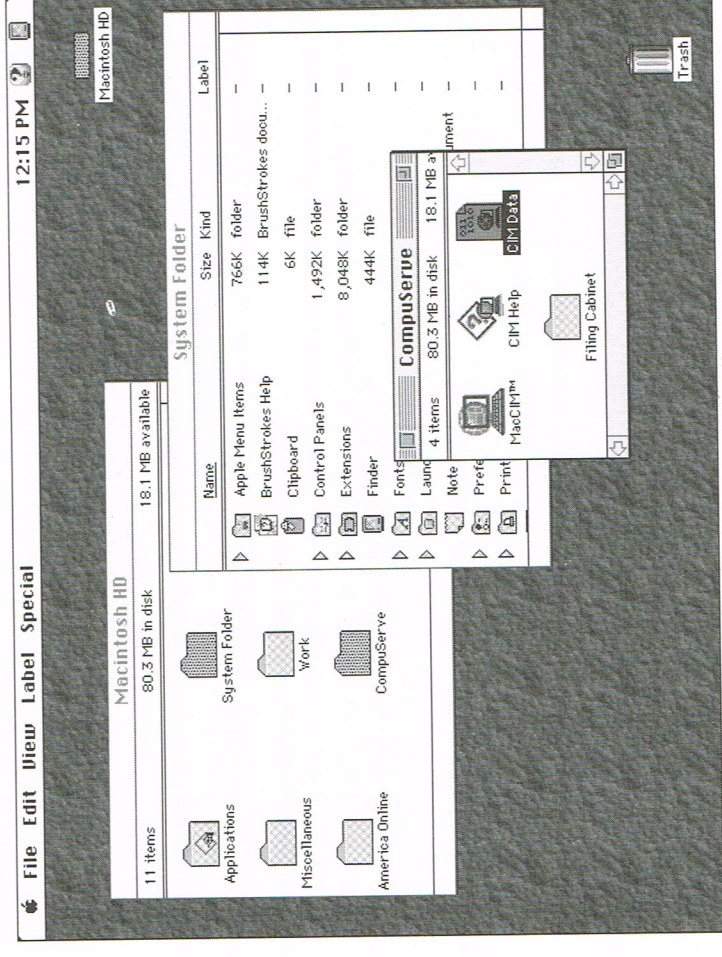**F I G U R E   2 - 1 2**
The Macintosh had the first widely-used graphical user interface.

the program is run. Programming languages such as Pascal and C++ use compilers rather than interpreters.

Figure 2-10 shows the steps involved in using a compiler. First, the source code is translated using the compiler to a file called an *object file.* An object file, however, is incomplete. A program called a *linker* is used to create an executable program. The linker combines the object file with other machine language necessary to create a program that can run without an interpreter. The linker produces an *executable file* which can be run as many times as desired without the need for translating again.

Although using a compiler involves more steps than using an interpreter, most C++ compilers automate the task and make it easy for the programmer to use. For example, most compilers allow you to compile and link in a single operation.

Programs you use regularly, such as word processors and games, are examples of programs written with a compiler. Compiled programs require less memory than interpreted programs because a compiled program does not require that an interpreter be loaded into memory. Compiled programs also run faster than interpreted programs because the translation has already been done. When a compiled program is run, the program is loaded into memory in the machine language the microprocessor needs.

F I G U R E   2 - 1 0
Compiling a program involves a compiler and a linker.

INTRODUCING C++

In this book you will learn to program in a compiled language called C++. You may have also heard of a language named C. The C++ language evolved from C. Most people call C++ a "better" C.

Why Is It Called C?

In this book you will learn to program in a compiled language called C++. You may have also heard of a language named C. The C++ language evolved from C. Most people call C++ a "better" C.

On the Net

Learn more about the history of programming languages at http://www. ProgramCPP.com. See topic 2.2.1.

The language C got its name because it is a descendent of a language called B. Both languages were developed at Bell Laboratories. There was no A language. The language B probably got its name because it was based on a language named BCPL.

Although C and C++ are high-level languages, they provide the programmer with much of the power and flexibility of a low-level language. In fact, C and C++ are widely used by professional programmers to write many of the applications with which you are familiar, such as your word processor and spreadsheet programs.

Categories of Software

There are two basic categories of software: application software and system software. Application software is the software that performs the tasks you want the computer to perform. For example, application software is what you use to produce a document or balance your checkbook. System software coordinates the interaction of the hardware devices, controls the input and output, and loads application software into memory so that you can run the programs you want to run. The system software required to run your computer is most often packaged together and called an operating system.