

An Overview of OS-9

What Is an Operating System?

An operating system is the master supervisor of the resources and functions of a computer system. Computer resources consist of memory, CPU time, and input/output devices such as terminals, disk drives, and printers.

OS-9 is a sophisticated operating system for microcomputers. OS-9's basic functions are to:

- Provide an interface between the computer and the user.
- Manage the input/output (I/O) operations of the system.
- Provide for the loading and execution of programs.
- Create and manage a system of directories and files.
- Manage timesharing and multi-tasking.
- Allocate memory for various purposes.

Using OS-9 as Your Operating System

The most visible function of the operating system is its role as an interface between you and the complex internal hardware and software functions of the system. OS-9 was designed to make its powerful features easy to use, even by persons with limited technical knowledge.

Because an operating system provides only part of the overall software necessary to make the computer useful, *application programs* such as word processors and accounting packages tend to be the most frequently used programs. They are not part of the operating system, but they rely heavily on services such as input and output provided by the operating system. Most application programs are written by users or obtained from commercial software suppliers.

Similarly, programming languages are tools used to create application programs. These rely heavily on and are closely related to the operating system.

To help make OS-9 easy to use, a set of over 70 programs called *utilities* are included. Utilities are not part of the basic operating system. Instead, they are actually small application programs that provide essential housekeeping, management, customization, and maintenance functions. Some utilities, such as the μ MACS text editor, are useful, general-purpose application programs.

Using OS-9's Functions

OS-9's many capabilities and functions can be used in two basic ways.

The first method uses the utility command set and the `shell` command interpreter program. This allows you to type OS-9 commands directly on your keyboard. These commands are translated into the more complex internal system calls actually required to carry out the desired operations. The OS-9 utilities are described in detail in the **OS-9 Utilities** section.

The second method uses system calls. System calls are requests made to OS-9 within programs written in assembler or a high-level language. These system calls are available to load programs into memory; create new tasks; create or delete files; read, write, open, or close files; and so on. All OS-9 programming languages have statements that cause the program to use OS-9 system calls, often in a hidden manner. System calls are largely of interest to advanced programmers and are discussed in detail in the **OS-9 Technical Manual**.

Storing Information

Another basic function of any operating system is storing information. Without some way to store and organize your programs, data, and text, working on a computer would be extremely complicated.

+

OS-9 stores information in files and directories located on mass-storage devices such as floppy disks. OS-9 provides easy access methods for updating, storing, and retrieving files and directories through standard utilities.

OS-9 organizes all files into organizational structures called *directories*. A directory is actually a special file containing the names and locations of each file it contains. Directories can contain files and subdirectories. In turn, these subdirectories may contain other files and subdirectories. This is called a *tree structure*, or *hierarchical*, organization for file storage.

For more information, refer to the chapter on the OS-9 file system.

Multi-tasking and Multi-user Features

OS-9 is a *multi-tasking* and *multi-user* operating system.

Multi-tasking, or *multi-processing*, allows the computer to run many different programs at the same time. By rapidly switching from one program to the next, many times per second, programs appear to be running at the same time.

Each program running on the system is called a *task*, or *process*. OS-9 allows you to have one or more tasks running in the background, while a task is running in the foreground.

A foreground process is a task that requires your interaction. For example, if you are editing a file, it is a foreground process because you are actively using it. A program that prompts you for information is also a foreground process because you need to respond to it.

A background process is a task that does not require your attention. For example, if you are printing a text file, you do not have to supervise the printing process. Therefore you can have the file printing in the background while you edit another file. This frees the computer from the limitation of doing only one thing at a time.

+

A *foreground process* requires your interaction.

A *background process* does not require your attention.

OS-9's multi-tasking capabilities make it possible for efficient memory use, CPU time, and I/O operations to be shared by all programs without conflict.



Typical Multi-tasking Usage:

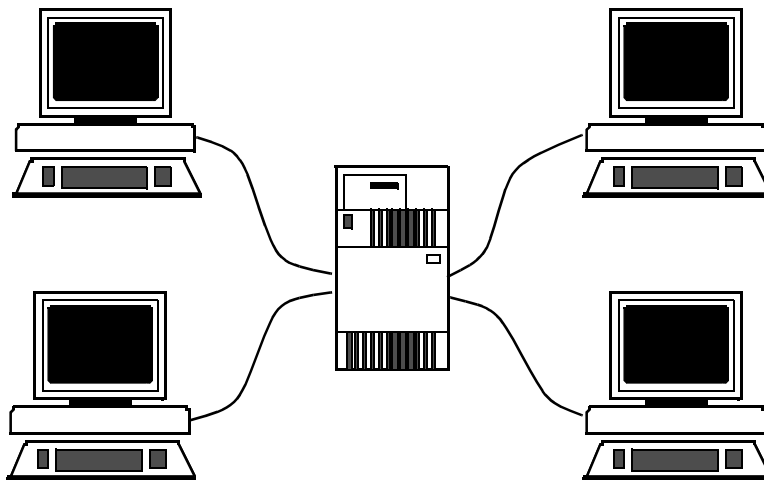
Editing a file (foreground process)

Listing a file to a printer (background process)

Sorting and merging datafiles (background process)

Multi-user, or *timesharing*, operation is a natural extension of the system's basic multi-tasking functions. It allows several people to use the computer simultaneously. OS-9 provides additional security-related timesharing functions to control access to the system and privacy within the system.

**Typical Multi-user System Configuration:
Four terminals on one OS-9 Computer**



The multi-tasking and multi-user capabilities tremendously increase OS-9's versatility. OS-9 is often used as a single-user/multi-tasking system on small computers. It is also used as a multi-user/multi-tasking system on larger computer systems. In either case, there is no difference in OS-9 itself, the application software, or how either works.

The Memory Module and Modular Software

A unique feature of OS-9 is its support of modular software techniques based on *memory modules*. The use of memory modules can:

- Provide more efficient use of available disk and memory storage.
- Make the system run faster.
- Simplify programming jobs.
- Make it easy to customize and adapt OS-9 itself.

All OS-9 programs are kept in the form of one or more *program modules* containing pure program code. They do not contain variable storage; OS-9 assigns variable storage in a separate block of memory at run-time. Each module has a unique name and can be loaded into memory or stored on disk or tape. OS-9 automatically keeps track of the names and locations of all modules present in memory.

An important characteristic of memory modules is the sharing of one module by several tasks or users at the same time. For example, if four users want to run BASIC at the same time, only one copy of the BASIC program module will be loaded into memory. Other operating systems would typically load four exact copies of BASIC into memory, thus requiring 300% more memory. The shared module system is completely automatic and usually transparent to the user.

Another advantage of memory modules is that frequently used functions can share common *library* modules. For example, a standard OS-9 module called *Math* provides basic floating point arithmetic operations for virtually all programming languages and programs. Again, this eliminates the need for each program to include its own math package. It also means that if you add a hardware floating point processor to your system, you only need to replace this one module and all your other software will automatically be converted without modification. In addition, large and complex programs can be split up into smaller, testable modules.

End of Chapter 1