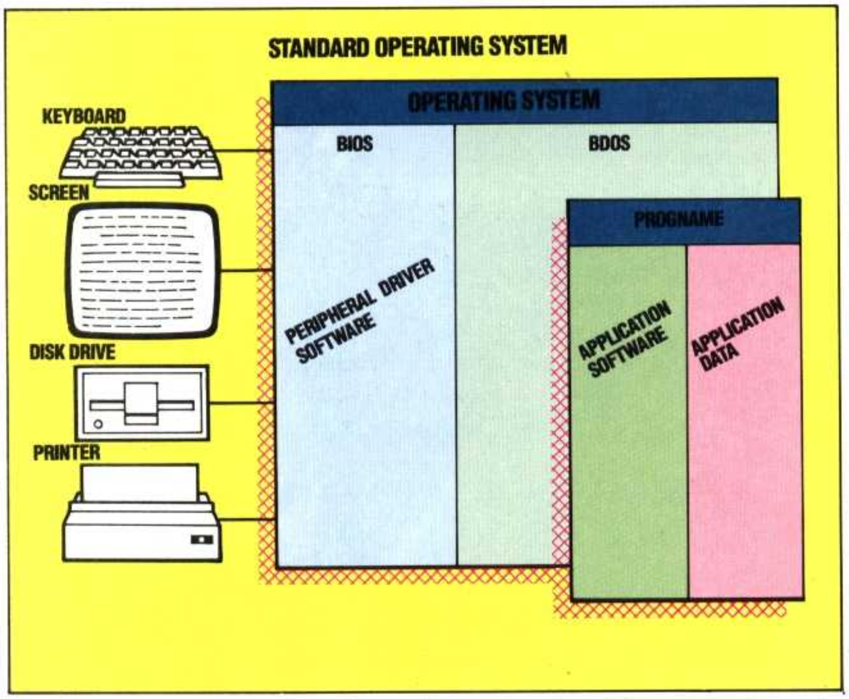
# COMPLETE CONTROL

**In the last instalment of this series on integrated software we looked at the most common approach, that of producing all-in-one programs covering all the functions that you need. However, this is not the best system, as such programs are huge and wasteful of memory. Now we look at a more versatile method.**



**Operating Under Orders**
With the traditional operating system the program currently running has complete charge. Its logic determines what appears on the screen, when the disk drive is to be accessed and how to interrogate the keyboard. Its general instructions are passed to the operating system, which manages the detailed driving of the particular hardware in use. The program's execution is paramount, and the operating system's subordination is taken for granted

The alternative approach to integrated software is based on a completely different principle. This relies on the computer's operating system to provide the basic facilities of integration, and individual programs written to work with that system will automatically fit and work together.

Creating such an operating system has been no easy task, since it requires the computer's hardware and software to be more sophisticated than in traditional designs. Apple has led the field with its custom-designed Lisa and Macintosh computers, although several other companies, notably Microsoft, are preparing systems to run on other popular computers such as the IBM PC.

Programs for these new operating systems are very different from programs for traditional systems. A large part of most programs deals with the user interface – the routines that receive commands and information from the user and present the results. Opinions differ on how programs should be operated, so nearly every package has its own unique operating procedures and needs to be learnt from scratch.

An integrated operating system provides a built-in set of user interface routines for every application program to use. When the program wants to display a list of options on the screen for the user to choose from, there's a ready-made routine to do it in the operating system. The advantage of this, of course, is that all the programs written to work with the operating system will have roughly the same operating procedures. Once you've learnt one program on the system, you're well on the way to using all of the others available!

One particular user interface provided for these programs is the mouse. This is a pointing device used to choose options from the screen via a corresponding cursor. An alternative is the 'touch-screen', in which a matrix of light beams responds to the touch of a finger. The display is divided into separate 'windows', each containing a different option or task. Technically, such a user interface demands a fast processor, lots of memory and very high-resolution graphics. But it is worth these extra costs because the system is generally applicable to almost any program available, it is very easy to learn and it provides the simplest possible way for the user to be able to see and switch between several applications at a time.

## OPERATION CONTROL

It is important to appreciate the way this system integrates programs. The program and user are never in direct contact – everything has to be done through the operating system and the operating system is in control the whole time. In effect, each application program becomes an extension of the operating system, and the computer is a single integrated 'environment'.

This brings us onto the second major difference in the way such systems function. In a traditional system, communication between program and operating system is very much one-way. The program asks for a specific task to be carried out and the operating system subsequently does it.

In an integrated system, the operating system is in control and make demands of the program. For example, the operating system may send a message to the program that says 'Could you redraw your display because the user has just moved it to the other side of the screen' or 'Hold everything, the user has moved the mouse to a different application' or 'Here's some data for you taken from a spreadsheet.' In other words, the program has to be able to respond to the requests and demands of the operating system as well as the other way round.