Once you have this degree of co-operation between all the software on a machine, it is easy to build an integrated environment. Each program has its own window on the screen. When the user puts the mouse inside the window and chooses an option, the operating system notifies that particular program and the relevant operation is carried out.

For example, if the user moves to the corner of the window and selects the option to pick up that window and move it to a new position, routines in the operating system carry out the task and then, if necessary, inform the program of the changes so that it can amend its display appropriately. If the user takes the mouse to a different window, the original program is temporarily dormant and the operating system starts working with the new program – switching between applications is as simple as moving the mouse.

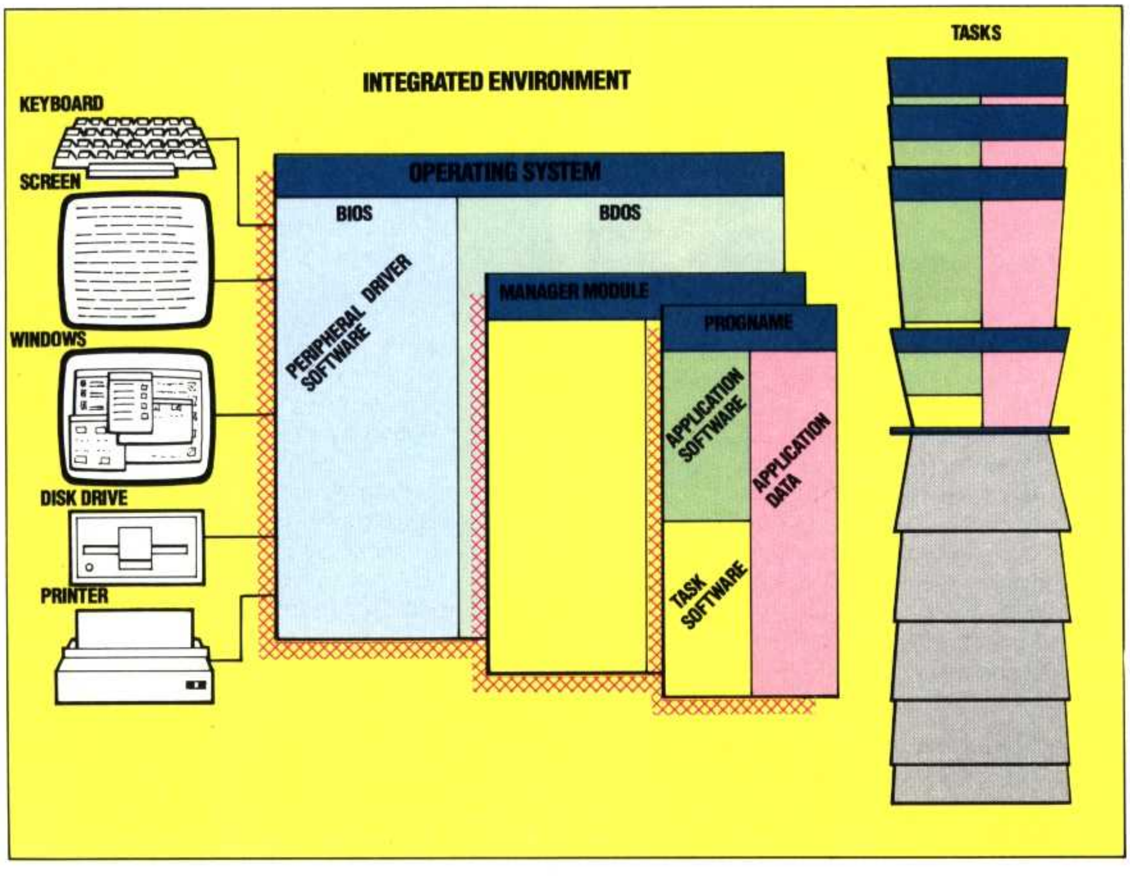
Like large all-in-one programs, such systems suggest that all the programs and information on the screen at any one time are in memory and available for use. To facilitate this, many systems have massive memories – one Megabyte on the Apple Lisa, for example, and 512 Kbytes on the Macintosh. Even then, it is usually necessary for the operating system occasionally to swap information and programs on and off disks to accommodate everything. To make the system acceptably fast, it is generally necessary for it to operate on a hard disk.

In order for data to be exchanged easily between programs, the operating system has a built-in set of formats and routines for transferring data. When you 'export' some data from one program and ask to 'import' it to another, the operating system will suspend the first program and start the second, then ask the current application to read in and process information coming from another program. These pathways can be set up automatically so that when you change information in a spreadsheet, for example, a graph of the same spreadsheet will automatically change also. The two programs don't run at the same time – the operating system merely juggles between the two of them as it needs to.

A slightly more sophisticated concept is demonstrated by Apple's Lisa, where information can be 'cut' to a clipboard window from any program and then 'pasted' into any other. Formatting information is carried with the data so that a graph produced with the business graphics software will be transferred as a graph into another program.

This then is the most sensible way to create integrated software. It enables you to mix and match any programs on the system, switch between them and move information between them easily. The drawback is that it requires sophisticated hardware that for the moment is quite expensive, and there is very little software available for you to integrate.

However, any technological innovation of this scale will take time to become commonplace. The mouse and windows interface was, for example, developed by Xerox's research teams over 10 years ago but it's taken until now for such a system to appear in the shops!



KEVIN JONES

Combined Operations
In an integrated system the operating system is enhanced by the addition of a manager module, which treats all current programs and data as 'tasks' to be scheduled and processed, and handles the underlying detailed operating system as simple system support software. This module moves tasks in and out of main memory and on and off disk according to user's requests and current task's needs. It is equipped to pass information to and from applications in standard forms, and so enables the transfer of data among the tasks. In effect the manager is a high-priority task itself, and its relationship with the other tasks is symbiotic rather than servile