



the 'current input stream'. This is normally the keyboard, but other input streams are the RS423 interface and the current filing system — the latter is accessed via the use of the *EXEC command. The major OS routine that deals with input from all these sources is called OSRDCH, which stands for OS ReaD CHaracter.

2. Output And Display Routines: These routines handle output from the computer; in the BBC microcomputer there are a variety of output streams that can be selected as the current output stream, such as the television display, printer, RS423 interface and, using the *SPOOL command, the current filing system. As well as being responsible for printing or otherwise outputting data, the OS display routines are also involved with tasks such as the control of the 6845 display chip in the computer and the use of user defined characters, to name but two of the extra functions of these routines. The OS calls used are OSWRCH (for OS WRite CHaracter), OSASCII and OSNEWL.

3. Filing Systems: Any operating system must offer the user a means of saving the contents of computer memory to a more permanent medium. The section of the OS that deals with such transactions is called the 'currently selected filing system', and on the BBC we have the choice of tape, disk, Econet, ROM or telesoftware. The OS filing routines can make use of additional routines in ROM that instruct the OS to deal with the hardware associated with a particular filing system.

To enable simple interfacing of magnetic storage media via paged filing system ROMs, a number of standard OS routines are provided to deal with file-handling. These include routines for writing or reading whole files, getting or putting single bytes from an open file and reading or writing specified groups of bytes to or from a file. The seven operating system calls concerned with filing each use a vector to point to the relevant routine in the cassette filing system. If a paged filing system ROM is fitted, it can dovetail with existing BASIC or Assembly language file processing programs by simply changing these vectors to point to its own ROM routines. One ROM of this type is the DFS ROM that allows the BBC Micro to use floppy disk drives.

4. Interrupts: An interrupt is basically a signal generated in either software or hardware that tells the CPU to stop what it's currently doing and perform a task that requires immediate attention. After doing this, the CPU then resumes its task as if nothing had happened. In the BBC, there are a variety of interrupt-driven facilities that the user can gain access to via the OS.

In addition to these four main areas, there are two vitally important OS calls that deal with a lot of different functions of the machine. These are called OSBYTE and OSWORD, and are used to control the sound chip, the break key action, and similar things.

WHY USE OS CALLS?

In most cases, information about the internal arrangement of a computer's memory and hardware becomes available (legal injunctions permitting) within a few months of the release of the machine. If we know these details, why should we bother using operating system calls? Why not simply access the devices or memory directly?

Part of the answer to this question was mentioned at the beginning of this article: using OS calls offers you some protection from hardware or configuration changes made to the machine by the manufacturer. In a similar fashion, calling routines by using their addresses within the ROM will cause problems should the ROM be altered; if you access ROM routines through the appropriate OS calls then you will be protected from this problem.

The instructions below will, in a standard BBC Micro, write the value 200 to the BBC user port that is at address &FE60.

?&FE60=200

If a second processor is added to the computer, however, this routine will not send the value to the user port, but will write the value to a memory location in the second processor. By using the appropriate OS call we can get round this problem; the OS call will know how to write the value to the user port whether or not the second processor is connected. The OS call for this is:

***FX 151,96,200**

This call is one of several that can be used to access areas of the BBC, such as the user versatile interface adaptor (VIA), the system VIA and the 1MHz bus.

Furthermore, why re-invent the wheel? If a routine to perform a particular function already exists within the machine, then why bother to go directly to the ROM routines? The chances are that the OS call will be more efficient than your direct access of the various routines involved in the OS call.

The only really good reasons for directly accessing memory or hardware devices are if you require speed of access or if an OS call doesn't exist to do the job. If speed is vital then direct access is often faster than going through the OS routines. However, it is still rather tricky to access the machine directly in this way, and we should take care in doing so — always remembering that programs that work on one particular machine may not work on other machines with different OS or hardware connected. This kind of problem is associated more with the BBC OS and the number of changes that it has undergone in its life time, than with the Spectrum, which has had the same OS throughout its history. In the next instalment of this series, we will begin to look at the BBC OS routines in more detail. Our discussion of the Spectrum OS, and other machines, will follow later in the course.