

Archie RISC OS

Acorn's RISC OS is everything the Archimedes' original Desktop should have been but wasn't: a multi-tasking shell which gives users a working environment they had previously only dreamed of.

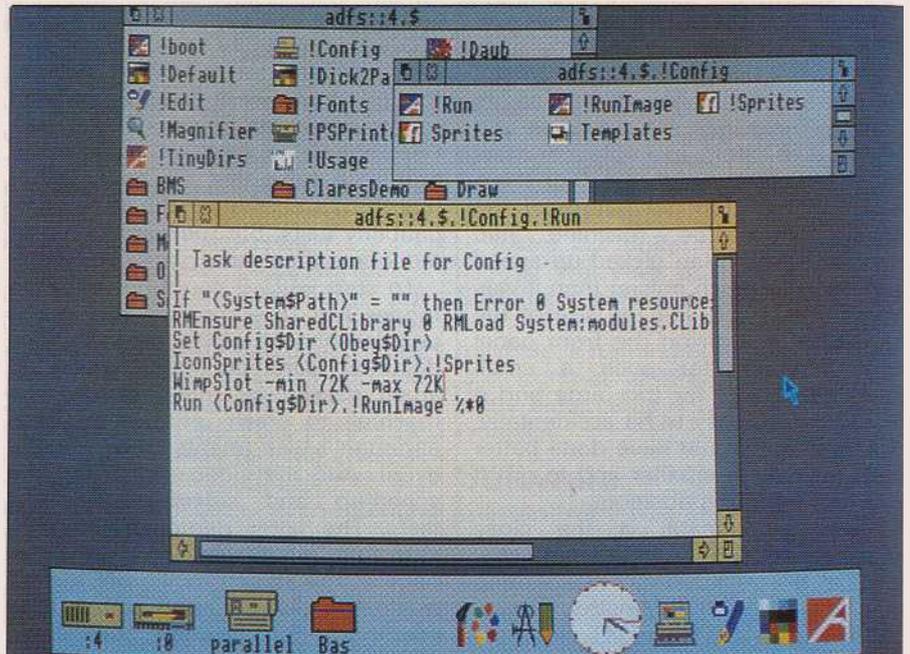
Dick Pountain examined its qualities and its quirks.

When Acorn launched the ARM-powered Archimedes in 1987, it came equipped with an operating system with the unlikely name of Arthur and half a desktop. Arthur was (and, indeed, still is) a development from the operating system of the BBC Micro and it presents a user interface which is every bit as hostile as that of MS-DOS, although to be fair it has some powerful features that make it bearable for experienced users.

The Desktop supplied with the original Archimedes was clearly unfinished; as part of the Welcome package it seemed as if it was meant as a taste of the future rather than a serious tool. Although it had working icons and windows there were many things you couldn't do with it. You could not, for example, pass parameters to a program launched by mouse clicking; many of the supplied applications did not return properly to the desktop after execution and few of them ran in a resizable window.

RISC OS, a completely new multi-tasking desktop shell for the Archimedes, is clearly what Acorn intended the Archimedes to have from the start, but presumably time ran out as it so often does in this business. It was worth the wait, however, as RISC OS completely transforms the usability of the Archimedes for ordinary users; and, with few exceptions, it is snappy enough not to irritate experienced users either.

Acorn loaned me an EPROM version of RISC OS for the purposes of this review. The finished form will be contained in a set of replacement ROMs, but this is not due to be released until April 1989. I found the EPROM version to be robust and bug free, and Acorn confirmed that it is more or less the final version and that the release date is governed by



The RISC OS windowing system owes much of its layout and design to previous windowing environments — with icons, windows, scroll bars and gadgets

the availability of ROMs rather than the software.

The Desktop

The RISC OS Desktop is still, strictly speaking, a shell over Arthur 2.00, but it lives in ROM and is automatically initialised so that you don't see all the resident modules being loaded as you did with the old desktop. It is no longer written in Basic as the old desktop was, and some of Arthur's directory handling functions have been enhanced to make the desktop metaphor work more smoothly (of which more later). It's still possible to exit from the Desktop to an Arthur command line by pressing the F12 function key, or even to set the machine to boot into Arthur. But normally the Archimedes will boot straight into the desktop, a

practically instantaneous operation.

The opening screen is blank apart from a strip at its foot called the 'icon bar' containing an icon for each device (hard, floppy or RAM disk, or Econet) in your system, together with the multi-coloured Palette icon and a strange icon bearing only 'A' for Archimedes which is the Task Manager icon. Clicking on a disk icon opens a directory window for that disk. The directory can be displayed as words, large icons or small icons: I much prefer the latter as they take up less space, so you see more to a window. They have the filename alongside rather than underneath.

Windows offer scroll bars at the right and bottom edges, a zoom box at the top right, a close box at the top left and a resize box at the bottom right corner. The only thing out

of the ordinary here is that the Archimedes' remarkable graphics speed allows you to resize windows with instant effect; in other words, the contents of the window are updated in real time as you drag the resize box, unlike the Mac or GEM where you drag a dotted 'skeleton' window that is only updated when you stop. If the effect makes you seasick you can alter it to Mac-style from the configuration menu.

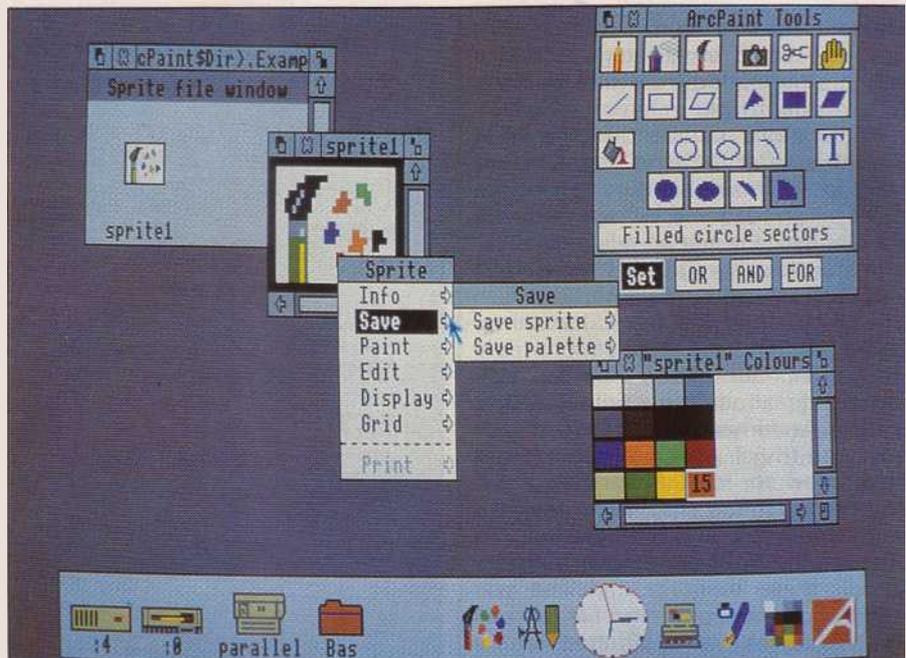
Every window or icon has its own menu that you pop-up by pressing the middle mouse button. The menu appears at the cursor position rather than pulling down from a menu bar; I believe that this is better ergonomics but avoiding Apple's predatory lawyers is a useful side-effect. The menu is context-sensitive and contains different operations in different places; for a directory icon it contains all the housekeeping operations such as Copy, Rename and Delete and creation of new sub-directories. It knows which icon you have selected and presents options related to that named selection, or to the whole directory if nothing is selected. It's a multi-level 'walking' menu, where sub-menus appear displaced to the right of their parent option (cf. Amiga Intuition or SideKick Plus).

Acorn has added a new twist which is that you access a sub-menu by simply sliding the cursor off to the right rather than by clicking on the parent option; options with sub-menus are marked with a right arrow so you just follow the arrow (see opposite). Often after following a chain of such sub-menus you will end up in a dialogue box which has an editable text field and an OK button to click. All very clever in theory, but in practice I have found it hard to get used to, for if your hand wavers in the vertical axis you end up popping the wrong sub-menu.

As an alternative to using the menu you can perform many operations by dragging. You can copy a file by just dragging its icon to another window or move it by holding down the Shift key while you drag.

The Desktop uses Arthur's file types to give different files appropriate run actions so that if you double click on an icon, it will usually do something. An executable program will load and run, a sprite file will display its sprite, and a data file from an application will invoke that application to display itself.

Some of the applications supplied on the new Welcome disk set have a subtler action, though. For example, when you click on ArcEdit, ArcDraw or ArcPaint, instead of running immediately, they cause the large version of their icon to appear on the



Acorn has opted for pop-up menus that appear when the middle mouse button is clicked. A sub-menu is selected simply by moving the mouse to the side



RISC OS also provides a number of desktop accessories including two clocks, a calculator, a puzzle and a magnifying glass for instant zooming

icon bar at the foot of the screen. This means that the program has installed itself in memory, like a resident utility or TSR under MS-DOS, and it can be invoked whenever you wish without loading it again from disk. (In fact such applications, unlike MS-DOS TSRs, continue running as tasks when they are installed on the icon bar, but more of that later.)

You can run an installed application either by clicking on the icon bar icon (as opposed to the disk icon, which remains in the directory window) or by simply dragging a data file's icon to it and 'dropping it in'. Unwanted resident applications can be killed by pointing to their icon bar

icon and selecting 'Remove' from the middle-button menu.

Only specially written applications have his property of automatically installing themselves on the icon bar, but the icon bar is also a handy place to 'park' applications and directories that you are using a lot, to save having to search through a window hierarchy each time you need them. RISC OS provides a general way to achieve this through a special application called '!TinyDirs'. When you click on !TinyDirs it creates an unnamed directory folder icon on the icon bar. Any other application or directory can now be dragged onto this icon and, immediately, its icon

appears on the icon bar. You can now execute it by clicking, though it will still be loaded from disk unlike fully resident task.

Applications installed like this sit at the left side of the icon bar, whereas tasks sit at the right. When there are more than can be shown on the screen, the icon bar scrolls sideways.

In that last paragraph I referred to 'applications and directories', but in one sense they are the same thing; under RISC OS applications are directories. One of the most frustrating things about the original Arthur on the Archimedes was that, to run any non-trivial program, you might also need to load a variety of resources such as font files, resident code modules and sprites, and to allocate the right amount of memory for all these resources. Unless you had the correct command file (and all the resources were situated where this command file expected them to be) then you could play all day without getting anything to run. This sort of messiness fits very poorly with the desktop metaphor where everything is supposed to be just a click away, and so RISC OS has been modified to hide it all from you.

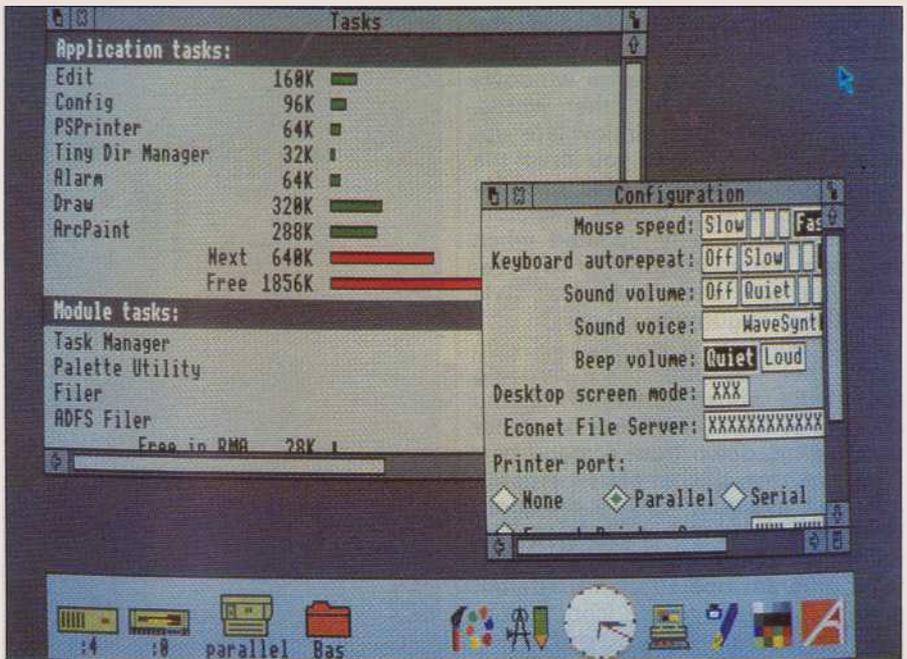
An application is a special kind of directory that contains all the resources needed to run it. When you double click a directory whose name begins with a '!' character, RISC OS looks inside for a file called !Run which contains a set of CLI commands to start up the application. To open an application directory and view its contents rather than executing it, you must hold down the Shift key while clicking.

Typically, !Run will check that the right modules are loaded using a new command called RMEnsure, which is followed by an action to perform if the module is not found: So, a line like:

```
RMEnsure FPEmulator 0.00
RMLoad System.Modules.
FPEmulator
```

will check that floating point emulator module version 0.00 is loaded, and load it if not. !Run also needs to allocate a chunk of runtime memory using the Wimpslot command, and then calls either the application code itself or perhaps a separate loader program. !Run files can be nested recursively to any level. You may also have a second file called !Boot inside the directory that is only run the very first time you click the application; useful for RMLoading modules and setting up Aliases which only need to be done once.

!Run and !Boot belong to a new file type called 'obey' files which differ from command files in that they



Multi-tasking is a complex process to follow and control. Acorn provides two utilities, !usage (see lower shot, p1531 and !tasks. The task manager allows all the current applications to be listed with a graphical representation of memory usage

can only contain Arthur CLI commands (more like DOS batch files). Obey files can find out what directory they were activated from by inspecting an operating system variable called Obey\$Dir, and they can use this to remember the path. For example:

```
run <Obey$Dir>.!Prog
```

Applications are assigned a default icon with APP on it; but if you draw your own icons and store them in a file called !Sprites, then put an IconSprites command in the obey file, they will be used instead. You need two, one big and one small.

Welcome applications

The Welcome programs supplied with RISC OS now extend over two disks/ and are much enhanced compared to the original Archimedes set. The most important one is ArcEdit, a general-purpose window-based text editor that at last provides a quick and convenient tool for creating and editing command files and similar jobs. Indeed, ArcEdit can serve as a simple word processor as it supports cut and paste, find and replace with wildcards, indents, variable margins, line spacing and fonts/ and an Undo feature. You can cut and paste between documents in different windows directly, without going through any intermediate Clipboard, and you can insert one file into another just by dragging its icon into an open editor window.

What ArcEdit does not have is proper wordwrap, which is rather bizarre given the other powerful fea-

tures. The option called Wrap on the menu causes the text to wrap to fit the window as you resize it, but it breaks words without hyphenation.

ArcEdit can create a number of different file types in addition to plain text; you can use it to make command files, obey files, binary data files and task windows. A task window is an Arthur command shell in its own scrolling, editable window, which you can use to run teletype-orientated programs like C compilers. It's often useful to be able to scroll back up and view, edit and save the compiler output. You can open several task windows and run multiple programs. ArcEdit knows all about Arthur file types and will edit any file you throw at it, displaying in an appropriate format; binary files are displayed as a mixture of hex bytes in square brackets and printable ASCII characters.

Two new drawing programs are supplied. ArcDraw is a powerful object-based drawing package, along the lines of MacDraw, whose features include arbitrary scaling and rotation of drawn screen objects by simply dragging on their 'handles'. It can also add text in multiple sizes and fonts to a drawing (including anti-aliased fonts) and import columns of text from ArcEdit to perform crude desktop publishing.

ArcPaint, on the other hand, is a sprite editor for creating purely bit-mapped images like icons. Like ArcDraw it offers a range of geometric drawing facilities for lines, rectangles, circles and arcs, and can add text, but only in one font. It permits some transformations such as hori-

zontal and vertical reflection and rotation through multiples of 90 degrees, and clipping to size by removing ' rows or columns of pixels. ArcPaint images can be imported into ArcDraw pictures but they may not then be scaled or rotated.

In addition to the two drawing programs several utilities are supplied. !TinyDirs has already been mentioned, and !Usage will be described later. !Config offers a preferences menu for things like mouse gearing and sound volume, replacing those *CONFIGURE commands you used to type at the CLI. !PSPrinter offers support and configuration for Postscript printers, and !MailMan is an electronic mailbox for those who are on an Econet.

Then there are the usual desk accessories: two clocks (analogue or digital alarm), a calculator, a magnifying glass and a puzzle. All of these are installable on the icon bar. Some things have gone — for example, the diary, the notepad and the font editor, while Maestro the music editor has been cleaned up, simplified and debugged.

Multi-tasking

RISC OS contains a simple but efficient task scheduler so that you can run more than one program at once. The mechanics of multi-tasking are quite straightforward as far as the user is concerned: you just keep clicking on the programs that you want to run, and they run. Programs that have been installed on the icon bar can be invoked repeatedly in different windows, so to edit several different files you just open several copies of the editor. Some older pre-RISC OS programs (for example, the Basic V editor) don't know about windows or tasking and just grab the whole screen and the whole CPU. In this case you can't launch any new tasks until you exit, but when you do all your previous tasks will come back to life as you left them.

The RISC OS scheduler is a 'cooperative' as opposed to a 'preemptive' or time-slicing scheduler like those used in OS/2 or Unix. The difference is this. With a cooperative scheduling algorithm, tasks each get a turn at the processor in a round-robin fashion; they take as much CPU time as they want then relinquish the CPU to the next task in the queue. This scheme works effectively only if each task takes a sensible slice of CPU time and remembers to let go, hence the name 'cooperative'. You can think of it as being analogous to an ideal anarchist society in which everyone is helpful and no-one is greedy. Of course the programs must be written by the programmer to be cooperative, by placing

an explicit instruction to 'pause and let the next task have a go' at a suitable point in the code.

A pre-emptive scheduler, on the other hand, gives each task a strictly rationed portion of CPU time, or timeslice; when this timeslice is up, the next task takes over regardless. Such schedulers are used in Unix and OS/2 and will normally use a timer interrupt to apportion time-slices.

In fact pre-emptive schedulers can be subtler than this, employing either user or dynamically variable priority levels (as OS/2 does) so you can tune the system to give each task what it needs.



'It's a shame that this impressive environment was not in place at the Archimedes launch.'



The advantage of a cooperative scheduler is that it is simple to implement, robust, and has very low overheads. The disadvantage is that it offers no control over the time apportioned to each task, and a hog can deprive all the other tasks of CPU time. RISC OS gets around this quite well by using the window manager event loop as a sort of pseudo timeslicer. Every application that is 'desktop aware' needs to call the window manager to display itself, and its main body must consist of an event loop in which the program idles, waiting for events from the keyboard or mouse; when an event is received the program jumps off somewhere to do the appropriate action. RISC OS puts an instruction to give up the CPU to the next task into the window manager itself, so that merely by calling the manager a program becomes well behaved and gives up the CPU at least once per event loop.

The net effect of this scheme is that if so many tasks are running the CPU time becomes scarce, then the program which has most events in effect gets the highest priority; which is almost always desirable since such a program tends to be the interactive application you're running in the foreground with the mouse, and you want the response time to be as fast as possible. If you run a CPU-intensive animated graphics task in a window and then open the editor

and start typing like hell, you can see the animation slow down or even stop while you type, but the typing response remains sharp.

Acorn supplies two utilities to help you manage tasks. One, an indicator called !Usage, shows CPU loading as a slowly unfolding bar chart in a small window; when the bar hits the top, the CPU is fully utilised.

The other utility is the Task Manager which lives permanently at the right-hand end of the icon bar. From this you can open a window which shows all the currently running tasks, and the memory they occupy measured by horizontal bar graphs. Other parts of this window show you the amount of free memory, the amount of memory that will be allocated to the next task, and all the various system tasks and memory areas such as the sprite, font and module caches, sound buffers, RAM disk and screen memory. You can use the mouse to alter many of these by grabbing the bar and sliding it.

All of this is made possible by the Archimedes hardware memory manager, the MEMC chip, which not only prevents tasks from encroaching on each other's memory areas but also allows memory areas to be resized without crashing the system disastrously. If a task should crash under RISC OS, it is usually possible for you to shut it down and reclaim all its memory without the other tasks being affected.

Conclusion

I find RISC OS a pleasure to work with, give or take the odd quirk. It is snappy enough that I never started to mutter 'Come on!' (or worse) through clenched teeth the way I confess to doing on the Mac, the Amiga or the Atari. The multi-tasking is so transparent and so obviously complementary to the desktop metaphor that you wonder how windowing systems were ever designed without it. The new tools like ArcDraw, ArcPaint and ArcEdit are good, with the promise of better still to come like the Acorn desktop publishing package and various third-party programs.

As far as compatibility goes, I found I could run all the old applications that I have under RISC OS (including the not-so-old Oak PDT), though sometimes you'll have to fiddle with configuration and directory paths for a while. Even applications which hog the CPU and screen will still return properly to the desktop.

It's a shame that this impressive environment was not in place at the Archimedes' launch, but it's still not too late for it to turn some heads.

Acorn is on (0223) 245200.