

Gerald's Column *by Gerald Fitton*

One suggestion for future packages for the RISC OS machine is that software programmers should co-operate with applications specialists when writing software. I am not convinced that this is the best way forward and, in this article I suggest an alternative strategy. Finally I offer an opportunity to write a useful BASIC program.

Background

On the Archive-on-Line List there has been some discussion about the need for new RISC OS software. In last month's Archive I argued against the need for a new 'Killer Spreadsheet' package and two other contributors countered my arguments. They believed that something similar to Microsoft Excel or Lotus is needed.

At a more general level (on A-o-L) one correspondent wrote:

“Much software, particularly that of a specialist nature, suffers from being technically good but practically lacking. This is because the writer is a programmer rather than a specialist. Real quality could be imparted to new RISC OS software if the prospective writers were able to collaborate with professionals within the subject area for the program.”

I would add that, even with good software:

A !Paint program will not turn anyone into a Vincent van Gough
A Word Processor will not turn anyone into the Poet Laureate
A Spreadsheet will not turn anyone into a Maths Wrangler

The time it takes to learn, say Accountancy, is much longer than the time it takes to learn how to use an accountancy package. The package will not create a brilliant accountant out of a non-numerate user overnight! Even with the best packages it is essential to know the non computer aspects of a specialist subject—the computer package is a tool and should not be allowed to corrupt good practice. At the college where I used to work, the Lecturers of the Secretarial Skills course learned this fact the hard way!

Much of the better software I have used has been written by someone who is primarily a specialist and has the secondary skill of programming. The speciality takes many years to learn whereas learning to write a program is a relatively easier process!

Nowadays it is harder to write a program than it used to be. Nevertheless I would like to see a return of the application specialist to programming.

BBC BASIC

In the days of BBC BASIC, programming was much easier to learn than it is now. From my own personal experience I know that students found simple mathematical procedures such as the solution of simultaneous equations easy to program in BBC BASIC. How many present day computer studies students would be able to do that now?

Bob Ardler once wrote to me that there were a few golden years for young budding mathematicians to write programming tools (in BBC BASIC) which did wonderful mathematical things for them. Bob regretted the passing of the 'easy to program' machine.

Of course BBC BASIC is still with us but I know of no major RISC OS package which is written in BASIC. It still has its uses as we shall see towards the end of this article.

WordWise

There will be some of you who used the relatively simple WordWise programming language to do marvellous things with that Word Processor package. Paul Beverley produced many booklets of WordWise programs but even amateur writers could and did write programs which were widely distributed.

The WIMP

Now we have a GUI WIMP and multi-tasking, multi-threading, distributed computing, etc, etc. It has made programming much harder! I do not dispute that these programming features do make the use of the computer much easier but it does make it harder for, say, a musician, to write a piece of musical software or for a mathematician/astronomer to write a simple algorithm for finding the date/time/place of the next eclipse.

Like Bob I do mourn the passing of the 'easy to program' BBC BASIC in favour of the GUI WIMP but I know that such progress was, and is inevitable.

The Way Forward

What I see as the way forward is for some brilliant programmer to bring back the simplicity which was present in the BBC BASIC (but not in the same form).

Please can we have a Graphic User Interface with Windows, Icons, Mouse, Pointer, multi-tasking, and all these other etceteras of the most up-to-date operating system but with a user interface which can allow musicians, artists, mathematicians, writers, etc to write simple 'Scripts' which interface with the OS and so bring back the expert into the writing?

Let me make my point perfectly clear.

The suggestion on Archive-on-Line was that a pool of mentors, specialists in their fields, should volunteer to help programmers include all those features which the specialist knows are essential but which the programmer may not be aware of. I wonder how that list of specialist mentors is progressing.

My suggestion is this:

Instead of asking programmers to consult a specialist I would turn the whole concept around. I would ask programmers to produce an Operating System and a Language which will allow specialists to write 'simple' Script files which will integrate with all the magical WIMP features of a modern OS and do what the specialist asks it to.

The Current Compromise

There are some excellent compromise features built into many packages such as:

- Applets for Ovation Pro
- Command Files for PipeDream
- Macros for Schema
- Custom functions for PipeDream & Fireworkz

Also it is possible to write Acorn Obey and Command (Exec) files which can be used to control the operation of the computer.

All these are what I have loosely referred to as 'Script Files' which allow the user to do some programming. I know that the examples I've chosen are from a very limited field but I'm sure those of you who are specialists in other fields will know of other features of other packages such as musical and drawing packages.

One problem with these 'Script Files' is that they run more slowly than an efficient program. For example I once used my specialist mathematical skills to write a small program in BBC BASIC which would check whether a (very) large number was a prime number or not. When I used the same skills to construct a custom function using a PipeDream spreadsheet it took twenty times longer to execute. In machine code it would have been calculated in a flash!

Also, 'Script Files' use much more RAM!

There is an answer to the slow speed at which these 'Script Files' are executed and that is that we need even faster computers. If you don't accept that this is an answer then have a look at the history of computers and wonder why Ovation Pro could not have existed on the BBC B computer. Was it lack of memory? Was it lack of processor speed? Was it that everybody believed WYSIWYG impossible? PipeDream had its origins in the days when WYSIWYG was impossible. Fireworkz is WYSIWYG but is slower than PipeDream.

There are other excellent compromise tools which help a specialist such as:

- The Midi interface for musical composers
- The Scanner for artists
- The OCR for feature writers
- The Digital Camera interface

Technology Alone is Not Enough

I am sure that everyone with a RISC machine will remember that the Archimedes was running a WIMP when MS-DOS IBM machines had only a command line interface. Those with longer memories will know that the Apple had a WIMP before the Archimedes. I nearly bought an Apple as my first real computer but I was persuaded by a friend to buy a non-WIMP BBC B machine because of its greater flexibility. Now Windows is the definitive operating system; the innovative WIMP of Apple did not automatically guarantee their position as market leaders.

My point here is that although (in my opinion) the Operating System of 'our favourite machine' does need a big step forward, such a leap as that which I suggest does not necessarily guarantee success in the market place. Nevertheless I believe quite firmly that the next big step in computers will be in the Operating System. The new OS will allow more use of 'Script Files' so that specialists can program the machine—as they did in BBC BASIC and in the WordWise programming language. It will need a faster machine with faster (and more) RAM and it may need multiple monitors for its best use.

I know that many believe that the future of the Personal Computer lies with the Internet. I don't. I think that the Internet is a specialist application. I believe that it will be serviced by specialist 'machines' which look like a TV with an infra red or cordless keyboard.

There are many other applications which do not need a general purpose computer.

One of my neighbours has disposed of her Windows machine in favour of a Word Processor machine with a built in ink jet printer. She does not need a DTP, database nor spreadsheet. She certainly does not require a games nor a specialist graphics machine. She may want to get on the Internet but will not buy a general purpose computer for that purpose.

The future of the Personal Computer is not with the Internet but for personal and individual use as a working tool. To fulfil this role it must be capable of being programmed by the individual specialist—not only by an expert in programming.

RISC OS or Language

I have received many emails on this subject and I can not quote from all of them. I have chosen one I have received from Bob Ardler because of its lucidity.

Bob's response to my remark that the RISC OS WIMP environment has made programming harder is, "You're right . . . that fact is a disgrace to RISC OS."

He points out that:

"On another widely used and not very lovable OS you have C++Builder, where you open a window, drag in your window furniture and write your scripts in C++. The compiler generates the windows part of the code for you. It's kind of DrWimp and template editor and more besides. You don't have to be a programmer to get started."

Bob goes on to refer to the language Python. About Python he says, "It comes with RISC OS 4 and is a mouth-watering, love-at-first-sight language."

Bob also comments about C and C++. He says:

"What bothers me with RISC OS is the bottleneck caused by C and C++. The languages (especially C++) keep changing so that the Acorn and EasiC/C++ compilers are dead or dying and the nearest thing to up-to-date is the GNU Gcc suite. Gcc is pretty much for programmers and people who understand paths and command lines. The file structure is half a dozen obscene violations of RISC OS decency, and even some programmers have trouble with the swarm of libraries needed to make C/C++ actually do stuff.

“What this means is:

- 1 You don't have a language which kids and amateurs can teach themselves, thus maintaining the corps of RISC OS programmers. If they do play with C++ it will be on C++Builder and you won't lure them back.
- 2 Most translators are written in C/C++, so without a pleasant C/C++ you won't be able to port to RISC OS; whether it's script languages or one of the many other wonderful languages out there.
- 3 The language-poverty in RISC OS is dire. The ports are few, underfeatured, unusable (or barely usable), understandable mainly by the Unix-reared, they work only on RISC OS Version [n-1] and leak memory because they were written in C.
- 4 Fewer and fewer students will do fewer and fewer exercises on a RISC OS machine.

“Gerald, you see it as an OS rather than a Language thing. To me it's about Language.”

Upon reflection I have to agree with Bob that it is at least as much about providing an easy-to-program Language as it is about the Operating System.

If you want to contact Bob you can do so on Bob Ardler <ardler@argonet.co.uk>.

Filters

During recent months I have received much correspondence suggesting that PipeDream and Fireworkz should be upgraded to increase the number of packages or formats which are automatically Loaded and Saved.

We don't believe that changing the PipeDream code each time there is an upgrade or improvement in HTML or RTF is the best way of dealing with this requirement. PipeDream is a mature program and is fairly 'stable'. Those things which change more rapidly should be isolated from the main program so that we do not have to keep issuing an upgrade each time these other formats are changed.

What we are doing is to encourage third parties (that means you rather than us) to develop filters for Loading 'foreign' formats into PipeDream and Fireworkz and for Saving PipeDream and Fireworkz files in these 'foreign' formats.

Generally the filters we have received and distributed use a BASIC program to convert the files. I believe that using single tasking BASIC is often the simplest way of producing a filter program. One filter for which there is a demand is a filter for converting PipeDream files to HTML. This could be written in single tasking BASIC. If you want to have a go then we will give you what help we can by way of explaining the PipeDream constructs.

Of course the third party writing the filter (you) would be the owner of the copyright of the filter. If you wish then you could sell it and we would help with its promotion. However, up to now those who have created filters have made them available either without charge or for a nominal payment. Please contact me if you would like to undertake such a project.

Finally

Thanks for your many emails and letters. Thanks also to those who have sent me a self addressed sticky label and return postage with their letters. Again my thanks to those who have included an example illustrating their comments or difficulties as an email attachment or on floppy disc. My address is that of Abacus Training which you will find in the Fact File at the back of Archive.