

The word processor is dead...

Long live the text processor — Emacs is versatile, hardware-friendly, cross-platform and free. Is it too good to be true? asks Chris Bidmead. Plus news from AIX.

Back in the late seventies when I first started messing about with computers a debate raged about the relative benefits of WYSIWYG word processing versus what was then thought of as “conventional text processing”. To the average PC user today, it seems unthinkable that a mere text processor could challenge a word processor like Microsoft Word. Text processors merely handle text. They don’t deal (at least not directly) with refinements such as italics, bold, fancy paragraph formatting, embedded graphs and pictures. And they don’t show you in advance what your pages are going to look like when they hit the printer.

I must confess I’d readily accepted the pro-word processor proposition, until a number of mailings from readers started me thinking about it again. The debate has been sparked off by Caldera’s advertised move to port WordPerfect to Linux.

Let me fill in some background here. In 1988, in a move that was to prove dangerously debilitating to the company, Borland introduced a new word processor to the market. Borland Sprint challenged established word processor products like WordStar, WordPerfect and Microsoft Word on a number of fronts.

Firstly, it was easily adaptable, so it could be made to function like any of them — remember, these were all character-based; Windows had not yet caught hold. This was because it included a “deep” macro language that made it simple for even casual programmers to remodel not just the key-stroke mapping, but also the behaviour of the software. In fact Sprint was really just a bundle of low-level text handling primitives held together by the macro language. The macros were compiled, not interpreted, which made it fast and kept it fast, no matter how many complex macros you felt inclined to add.

Another key difference from the established word processors was that Sprint

came with a separate formatting program. Yes, it made a stab at previewing the final page on the screen if you insisted, but primarily it kept text formatting separate. The formatter could do very complex page layout, picking up visible ASCII format commands that you left in the text, but you didn’t need to see the effects of these as you wrote (so went the theory) because you trusted the formatter to create the page according to pre-set rules you had laid down. A simple example: you didn’t need to check each page to make sure that paragraphs stayed attached to the headers that preceded them, because the formatter understood that certain types of headers shouldn’t be separated from the text that followed. Yes, I know that today’s word processors understand these kind of rules too, but that’s because they’ve absorbed the lesson.

Familiar friend

I used Sprint regularly until GUI word processors became dominant with the arrival of Windows 3.0. As a matter of fact, Sprint had arrived as a familiar friend, because it was based on a product called FinalWord, which I’d been using since the mid-eighties. FinalWord, in turn, was the offspring of PerfectWriter; a CP/M word processor on which I’d written a couple of my Doctor Who scripts back in 1980.

CP/M was very tight on memory, and all the other CP/M word processors I’d tried couldn’t cope with a whole script as a single file. And the separation between composing the script and formatting it for printing was a real boon. The convention for Doctor Who scripts was to confine the dialogue and stage directions to the right-hand half of the page, leaving the left section blank for camera directions. But composing the dialogue like this meant wasting half the screen. PerfectWriter let you forget about formatting while composing a scene.

Writers should never have to bother



The UK Windows NT-based systems vendor ServerWare sent out over 5,000 CDs to customers to demonstrate new applications. Unfortunately, the accompanying documentation in Microsoft Word format was infested with the so-called “Prank” macro virus, one of the first known viruses in the wild to be passed via data files. ServerWare quickly issued a fixed version of the CD, and thoughtfully included this unique Word document, which not only explains the virus, but contains its own anti-virus macros to get rid of it!

with formatting, in my opinion. (But writers are funny people. I remember discussing this with Lukas Heller, who did the screen play of The Dirty Dozen and many other films. Lukas came from the typewriter-and-cigarette generation of writers, and confessed that he amused himself while composing dialogue by choosing words that made the right hand column come out as even as possible...).

You are probably wondering why I am taking you on this trip down memory lane. Well, Borland Sprint, FinalWord and PerfectWriter may all be Norwegian blues, but they are the direct descendants of an awesome text processor that very much lives on. If you installed your Linux from any of the standard distributions like Linux-FT, Slackware or Caldera, you’ll either have it on your hard disk already, or it will be very close at hand. If you’re running AIX,

What's so great about Emacs?

● It's fast and economical on hardware. Because it uses plain ASCII (the X features are something we'll come to later), screens take no time to repaint and (for a slightly different reason) you can move from the top to the bottom of a huge text file in microseconds. It's also fast because it doesn't use a mouse (X features excepted), so your hands stay on the keyboard. This would be a mixed benefit without another important design feature: the Emacs cursor moves not just according to screen geometry, but also understands units of text. Simple keystroke combinations move it forward and back by word, sentence and paragraph. In the hands of an experienced Emacs user this makes navigating and editing very fast indeed.

● Emacs is totally configurable and extensible. I don't just mean that you can remap the keystrokes to match your favourite existing word or text processor — this is a trivial exercise, and one that will turn out to be a waste of time if, like me, Emacs quickly becomes your favourite text processor and you find yourself remapping everything back again in order to remain compliant with the standard. I mean you can change the behaviour of virtually every feature of Emacs, and add features of your own using Emacs Lisp macro language. The version of Emacs I'm using to write this — it comes with Caldera — has already been crazily extended to include Doctor, a variant of the old Eliza program that lets you

switch to another Emacs buffer and engage in dialogue with an electronic psychiatrist, and Spook, a random word generator that creates strings of words intended to alarm the US authorities supposedly monitoring internet email. More usefully, Emacs includes its own mail program and internet news readers, all written as Lisp extensions.

● Emacs is cross-platform. Yes, it's pretty complex to learn (although the built-in help pages and tutorial speed the process considerably), but once you've learnt it you have a set of skills that can pretty well be ported to any environment you can think of. Emacs is everywhere, and for a very good reason....

● Emacs is free. This isn't just a cheapskate attraction: Emacs is philosophically free. It's free of the political machinations of individual companies; it's free from being tied to the fortunes of any particular operating system. It's free (paradoxical thought this), of the onward march of new versions every 18 months that have become the marketing requirement of commercial software, and which cost users much more than just the upgrade fee. Like Linux itself, Emacs is issued under the GNU licence. This isn't just Open software, with that pompous, mendacious capital "O". Unix usefully distinguishes between upper and lower case. Like Linux, Emacs is not Open. It's *open*.

NeXTStep, UnixWare — you name it — a copy of this for your processor and operating system won't be more than a short FTP hop away. Likewise OS/2 and even DOS. As old hands will have realised, I'm talking about Emacs.

Word vs text

What's a word processor? The features that distinguish a word processor from a text processor seem to me to be all about preparing text for printing. I don't know about you, but I don't print anything much these days. Everything I write professionally goes straight down the telephone to people with their own definitive ideas about how the words I write should look in print. Yes, I have been sending stuff down in Microsoft Word format, but my occasional italics and my even rarer use of features like tables have probably hindered rather than helped the sub-editors at the other end.

More on Emacs and word processors next month.

AIX

At the recent IBM Technical Interchange at Disneyland, Paris, Mark Wieland of the Personal Power PC Group gave me a quick trip round the new version of AIX, version 4.1.3 which runs on a PowerSeries 850. I'll be rounding out the full picture of this machine and operating system once IBM manages to deliver the review machine they promised me in March 1985... ho-hum. But here are some introductory thoughts:

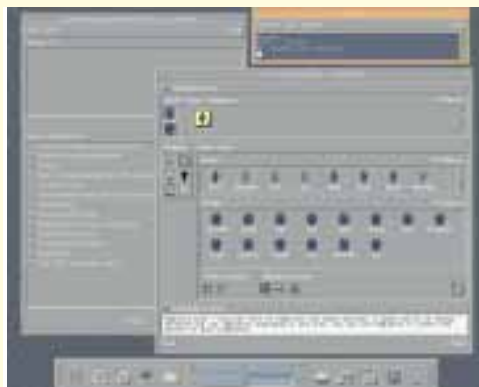
AIX is now (since version 4) a fully multi-threaded implementation of Unix. Version 3 implemented [pthreads], which is library support for threading in the user space only. In version 4, thread support was carried through to the kernel, so that every user thread has a kernel thread to support it. The importance of this is that each thread can now be handled in real-time, and in a multiprocessor AIX machine, different threads can be handed off to different processors.

Since I last looked at AIX, SMIT, AIX's



X-based system management tool, has been supplemented by a new Visual System Manager, VSM, which is activated through drag and drop. If you want to create a new user, it's simple: Pick up a "new user" icon and drop it among the collection of existing user icons. Pick up a password icon and drop that on the new user in order to pop up a dialogue box requesting the new password. And so on. There's a lot to drag and drop to in the CDE, the Common Desktop Environment now finding its way into the operating systems of the major Unix manufacturers. Mark showed me how you drop an Mpeg data file on the Mpeg player to power up the movie, and you can also, of course, open the movie directly by double clicking on the data item.

All this is distinctly old hat, I know, to OS/2 users. Indeed, it's from OS/2 that the technology to do this in AIX is coming, with the difference that the CDE associations between data files, their icons and their executables aren't dynamically established through SOM (System Object Model). Like Caldera's Looking Glass desktop manager, the AIX CDE associations are set up through a data file that you explicitly edit, either directly or via a special editor. CDE includes a floating launch tool very like the OS/2 Launchpad, but again this has to be configured via an editable data file. You can't just drag icons to it the way you can with OS/2.



The AIX Visual System Manager extends System Management Interface Tool with a drag-and-drop interface for tasks like setting up new users. OS/2 users may be wondering about the Launchpad-like bar at the bottom of the screen. This is a standard part of the Common Desktop Environment, and stems from the same Hewlett-Packard user interface technology that inspired Warp's Launchpad

Ironically, SOM is already a part of AIX, and as I understand it there's no reason that the AIX desktop shouldn't be as object-orientated as its OS/2 counterpart. I gather from Mark that this will happen eventually. But at the current pace of AIX development it may not be wise to hold your breath.

Christmas Wishes

This time last year, I drew up a list of goodies I'd like Father Christmas to stuff into my stocking. I wanted Cairo and Taligent, but guessed — rightly, as it turned out — that I'd have to make do with Windows 95 and another bunch of promises. I also wished for some rapid developments on the Linux front that would make the embryo Windows emulator, Wine, able to run a few standard Windows apps like Ameol, Excel and Word. That hasn't happened yet, but somehow I'm not holding my breath, and Linux is progressing fine without it. In the same spirit, I asked for a rather more substantial list of 32-bit apps for OS/2 to run. Again, disappointment. But IBM has promised that the new Windows extensions to the API will now, at last, bring them flooding in. I'll believe that when I see it.

Actually it's not apps I want anymore. I'm now in the mood for components. Deliver, somebody, somewhere, at least some of the promises surrounding OpenDoc. Just enough to get me believing that it's real. At Disneyland, Paris, I saw OpenDoc running on OS/2, and I can't say that the demonstrator was exactly fired with enthusiasm, or brimming with knowledge of the subject. It takes people to make this technology work, and alas real OpenDoc evangelists such as Kurt Piersol, one of the OD architects I met last year, are few and far between.

I also expressed a pious wish that the Unix community would get their act together. COSE, the Common Open Software Environment, is starting to become a fact of life, but in other respects, Unix seems more diffused than ever. SCO now owns the System V strain, which is probably good because they'll know how to sell it, something that totally defeated Novell. But I fear that SCO and AIX and the newly named Digital Unix are now going to be fighting each other rather than pulling together; something that's going to do Windows NT sales no harm at all.



Seminar

Speaking of OS/2, I had a chance to talk to Paul Giangarra, one-time chief architect of OS/2. He told me that it was his second time in Disneyland Paris that year — the previous visit had been as keynote speaker at what turned out to be an AIX seminar.

Closing the DisneyFest was David Barnes, IBM's OS/2 demo god. Personally I find Dave's high octane, joke-packed presentations just a little too shrill, particularly at a time when IBM is supposed to be concentrating on OS/2's corporate credibility.

But one message came very clearly through the non-stop patter. He likened the way OS/2 runs multiple 16-bit Windows sessions to a mainframe operating system running virtual machines; "something IBM really knows how to do. Doesn't matter what the operating system is in the virtual machine..." He was talking, of course, about OS/2's potential ability to cope (or not) with future 32-bit Windows applications. "Is our technology built with ones and zeros," he asked rhetorically? "You bet it is. Just like Microsoft's. I can't say definitely that we'll support Windows 95. But if at some time there are compelling applications that our customers tell us they have to run, we can do that."

Incidentally, if any of you are listening to the gossip about IBM downgrading OS/2, or even killing it off, Paul Giangarra is living proof that IBM has no such thing in mind. Yes, it's closing down the Boca Raton OS/2 development in Florida and moving the whole thing to Austin, seat of IBM's AIX development.

Barnes joked that he'd sensed this coming, and made the move a couple of months ahead of time. Giangarra told me afterwards that Barnes's early move wasn't so smart, because it meant he'd missed the relocation package IBM was offering. "They are so serious about continuing with OS/2," said Giangarra, "that they've offered me a fabulous deal to haul up my roots and go to Austin, provided I promise to stay working on OS/2 for the next two years. And that's exactly what I want to do anyway. My wife works for IBM too, and the combined deal means we can go down there and build the house we've always wanted."

PCW Contacts

Chris Bidmead is a consultant and commentator on advanced technology. He can be contacted at bidmead@cix.compulink.co.uk