Comparing like with like

Is it conformant or is it compliant? The importance of being Posix. By **Philip Ling.**

It wasn't very long ago that the embedded world scoffed at the idea that an operating system designed for the palatial IT world could find its way in to an embedded system. Memory constraints and 'real time' requirements brought to mind the proverb involving a camel and the eye of a needle.

However, where there's a will there's normally a way and embedded Linux is now a force to be reckoned with. True, there are still some issues surrounding its real time performance, but equally there are just as many interpretations of what 'real time' actually means.

Such is the swing in the balance of power that, today, the open source and open standards communities strongly influence commercial vendors. Licensing models have been put through the corporate wringer to make them more attractive to proponents of 'open software', whilst the most radical example in recent history has been the ceremonial unveiling of source code. And it doesn't stop there; currently, there is an ongoing and noticeable shift towards the adoption of Posix – to a greater or lesser degree.

Posix stands for 'portable operating system interface' and it is intended to ensure portability of applications across hardware and operating systems. The IEEE standard 1003.1 (Posix.1) system interfaces (of which there is more than 1000 parts) defines a portable application programming interface (api) to an operating system. As such, applications written to adhere to Posix.1 system interfaces can run on any platform with any conformant operating system – providing the platform supports all of the Posix functions implemented in the application. This is an important proviso, because it isn't mandatory to implement all of the functions in every platform.

Commercial suicide?

At first glance, adopting Posix could appear to be commercial suicide. Code written for a specific OS makes that OS an integral part of any system using or reusing that code. But it is just this kind of dependency that developers are trying to avoid, particularly those in the aerospace and military sectors, where Posix is now practically a prerequisite.

Although it has been around

since the 1980s, Posix wasn't really adopted by the embedded community until the 1990s and, even then, it was through independent standards. The result was a fragmented and uncontrolled standard that was becoming unwieldy. At the turn of this century, the IEEE stepped in to clean up the mess and the result was Posix 1003.1, a unified version of Posix that catered for the enterprise and embedded markets. At the same time, it was rejuvenated with extensions for threads and real time such that it is now at a point where its functionality matches the best commercial operating systems and a lot (but not all) of the confusion has been removed.

So, adding Posix should be akin to professional golfers playing off scratch – no single OS receives an inherent advantage or disadvantage, which means all operating systems will now be judged on their merits. Making them Posix compliant/conformant is supposed to make it easier to transfer applications across platforms, so each new project has the opportunity to use any number of Posix friendly operating systems. Additionally, suppliers of applications that are written for Posix platforms have a greater pool in which to fish. Theoretically, this should make for greater competition, as each new design win will need to be truly won; not just awarded. So just how level does Posix make the playing field?

Apples with apples?

Posix comes from the same stable as Unix and Linux – insomuch as it is an open standard. Although Linus Torvalds, creator of Linux, has stated publicly he has no interest or intention of making Linux Posix conformant, the fact is, in many people's minds and because of their 'openness', the two are linked. In truth, opinions are divided on just how conformant Linux is. Bob Morris, vice president of sales and marketing at LynuxWorks, claims BlueCat – its embedded Linux product – is around 94% conformant with Posix, while LynxOS – its real time operating system – is 100% Posix conformant and binary compatible with Linux applications. "Open standard, not open source, is the real driver for Posix," said Morris, adding that 'the Linux train is coming' and all it will leave behind will be applications requiring hard real time, which is why there is the sudden interest in adding Posix portability.

Fundamentally, there are two definitions at play: Posix compliant; and Posix conformant. The former implies a level of Posix portability, whilst the latter demands full portability. However, as mentioned earlier, it isn't mandatory for all the 1000+ interfaces to be implemented in every application, so portability of code ultimately comes down to having the necessary interfaces present. The result is comparable to optimised versions of embedded Linux, resulting in a bespoke operating system that is no longer guaranteed to run all Linux applications. In a deeply embedded (closed) system that isn't a problem, but today more systems are open to allow field upgrades, so full support is more of an issue.

The Open Group and the IEEE are making every effort to bring order to chaos by introducing a certification programme. Under the programme, a vendor can claim conformance for a product family or a platform – which essentially means an OS running on a specific processor. The processor requirements are moderate; it basically needs a memory management unit that can support memory protection and virtual memory. Green Hills was the first commercial embedded OS vendor to attain this level of certification, for its Integrity 5.0 operating system running on a PowerPC. David Barnett, director of product marketing with Green Hills Software, explained that the platform certification was chosen because it was the simpler option for Green Hills' first foray in to Posix certification. The company now plans to apply product certification to Integrity.

Barnett also voiced his concerns about Linux' ability to become fully Posix conformant, citing conflicts between Posix and Linux that, if fixed now to attain conformance, would break any backwards compatability with older Linux version. "You can take a Posix application and recompile it under Linux, and the chances are it would run, but it would run differently," said Barnett. adding: "It isn't Linux anymore if it's Posix compliant."

As formal test suites are released, it is expected that more OS vendors will opt for certification under one of the four real time profiles defined by the Open Group. Morris remarked that attaining certification under the most stringent profile – profile 54 – will likely require most vendors to make significant enhancements to their micro kernels, as such he expects to see more certification activity around the slightly less difficult Profile 53. His advice is to implement the largest profile you can but write to the smallest, to ensure portability.

> "You can take a Posix application and recompile it under Linux, and the chances are it would run, but it would run differently."
>
> David Barnett **Green Hills Software**

With more vendors adopting Posix, competition is expected to increase significantly, along with software reuse. At the moment, it is contained to those sectors with multi tiered supply chains such as aerospace, military and some parts of the automotive industry. But the embedded software industry is still in a state of flux, reeling from the impact of open source and open standards, Posix could present a steadying influence and rapidly become an integral part of all embedded systems.