

Zack's Kernel News

■ Breaking compatibility

Linus Torvalds added a new "IRQ status" return value to IRQ handlers in the 2.5 kernel tree, to help detect a variety of problem situations, including the case when an interrupt has been enabled before a corresponding handler has been attached. This will help the kernel recover cleanly when someone loads a buggy driver into the system.

However, changing the return values of established APIs breaks backward compatibility at the source level, which requires most driver code to be modified and recompiled before it will run in the new environment. For drivers that are included in the official kernel source tree, this is not so difficult, though tedious. Shortly after Linus made the change, Andrew Morton and others had gone through hundreds of files in the tree, correcting the code according to Linus' new specification.

Third-party drivers that are distributed independently are much more difficult to update, as their sources are scattered across many disparate locations, and

their maintainers may not be aware that any change is needed.

Drivers distributed only as binaries are even more difficult to fix, because users are unable to send patches that might help. The decision to break backward compatibility always entails a mess, in which some people are happy – typically those who wanted the new feature – and some are not.

On the other hand, once the decision is made to break backward compatibility in one way, then there is much less at stake in deciding to break it in other ways as well. Often in this kind of situation, a developer will say, "since we're already breaking most drivers for the new IRQ handler return value, we might as well also break this other thing that we've wanted to change for a long time, but always held back because of backward compatibility." So this "IRQ status" return value may open the floodgates for other changes that would not on their own have justified breaking backward compatibility at the source level. ■

■ Spy on Spies

A number of wireless LAN chip makers are keeping their specifications under wraps, apparently because the chips can be used to monitor military and government communications, and transmit on the same frequencies. Access to the specs would make this trivial to accomplish. Of course, binary-only drivers do exist for these chips under MS Windows, and hackers are already starting to reverse engineer them, to write their own more powerful drivers.

The vendors themselves, at least in some cases, would seem to be happy to publish specs and free drivers, but they are hesitant to upset various government agencies around the world. The next generation of this hardware will almost certainly make a serious effort to prevent this kind of use, but for now many hackers are dripping with desire to hack these chips, just for the challenge.

Someone wants to hide information, so these hackers want to expose it. I wouldn't be surprised to see many free drivers cropping up in the near future, capable of monitoring and transmitting on all frequencies physically supported by the devices. This situation is somewhat different from the usual case of a hardware company refusing to release information or drivers.

Typically, the hardware maker would like to protect secrets or hide bad work. In the old days, public specs were the exception, not the rule.

Linux, FreeBSD and other free operating systems changed all that, and now manufacturers will often release specifications and datasheets without being asked. These wireless LAN chips seem to fall into a new category, of companies that refuse to release specs because their hardware is too powerful. ■

■ Full Time Linus

With the recent release of the 2.5.72 kernel, Linus Torvalds broke some unexpected news, a leave of absence from Transmeta so that he can work full time on kernel development. Having been with Transmeta for more than six years, Linus has been in the lucky position of benefiting from their relaxed attitude, which has allowed him to continue work on the Linux Kernel.

Guilt, in a small way, has gotten the better of Linus, so he has taken the opportunity to work full time at OSDL, the Open Source Developers Lab. This will allow him to devote his full working time to developing the kernel in a vendor independent and neutral Linux environment. ■

■ Promoting Bart

Bartlomiej Zolnierkiewicz has replaced Andre Hedrick as the IDE maintainer, at least for the moment. Andre has stepped aside to work on SATA (Serial ATA) and vendor chipset issues. IDE politics have always been fairly convoluted. The official maintainer of IDE in 2.5 has been Alan Cox, after Martin Dalecki stepped down early in the 2.5 series.

Alan was chosen as an intermediary between Linus and Andre, who had always had trouble communicating. Although starting out as "only" an intermediary, Alan ended up cleaning up the entire IDE driver in a way that had not been accomplished for years, in spite of numerous attempts by very talented programmers.

Technically, Alan is still the official maintainer, and so Andre's decision to promote Bart to maintainer must be looked at more carefully. It might be more accurate to say that other developers working on IDE code should send their patches to Bart now, instead of Andre; and Bart will pass them along to Alan after his own review.

Meanwhile, Andre's work on SATA promises to bear sweet fruit, and could bring tremendous speed improvements to IDE users. Serial ATA controllers are already available on the market from companies like 3ware, and users are starting to tinker with them. ■

■ Testing standards

The Open POSIX test suite has reached version 1.0.0; this project intends to test any operating system's compliance to the POSIX specifications.

Signals, message queues, locking, timers, and process scheduling are all areas covered by the Open POSIX suite. POSIX conformance is seen as a big credential in the operating system world; even MS Windows supports POSIX APIs, although most developers prefer Windows' native functions.

Linux has always had a critical relationship to POSIX, and really to all official standards. It is something of a tightrope walk, because standards often

provide a mechanism for programs to be ported easily from one system to another. They also represent an attempt by serious people to solve the problems they address, which might otherwise appear intractable. At the same time, standards often represent an allegiance to the way various projects worked in the past, while better solutions may have been discovered since.

In such cases, compatibility with the past overwhelms good sense, and new conforming projects are held back from accomplishing the good things they might accomplish. Standards are also often ambiguous in places, causing the

various implementations to behave differently from one another, as their developers struggle to interpret the standard in ways that make sense within the framework of other technical decisions.

Linus has always encouraged developers to express their own ideas of how things should be, and not to be intimidated by the fact that something may be an "official" standard. So, while POSIX and the Open POSIX test suite may help support the vast majority of cases within Linux, there may always be some areas of POSIX and some tests from the suite that Linux refuses to handle. ■

■ Forking projects

Roman Zippel rewrote the HFS+ filesystem driver, the standard filesystem used in Mac OS X. His work, supported by Ardis Technologies and based on an earlier driver by Brad Boyer, supports full read and write access, hard links, and some significant performance improvements over the old driver.

When he announced his work, shouts of jubilation were heard from iPod owners and other folks around the world. Unfortunately, there was a bit of open source toe-stepping involved, as Roman had begun his code fork without first attempting to contribute to Brad's existing driver; traditionally, the authority of a maintainer is accepted to the point that they accept reasonable patches in a timely fashion.

Developers are expected to make every effort to work with an existing maintainer, and only to fork if it seems clear that the project leader is unwilling or unable to take their code.

One justification of this hesitancy to fork is to avoid too many projects pursuing the same goals. In the case of Roman's HFS+ goals, Brad made it clear after Roman's announcement, that he wanted to fold Roman's work into his existing driver.

Whether the two projects continue in competition, or Roman decides to submit patches to Brad in the future, or whether one or the other of them will abandon the work, remains to be seen in the future. ■

■ Freezing features

As of early May, specifically the 2.5.69 kernel release, Linus has taken another step toward moving 2.5 into a stable 2.6 or 3.0 series. In October 2002 he declared a feature freeze, but kernel development has never had a successful feature freeze on the first attempt, and this one was no exception. As more and more new features found their way into the kernel anyway, folks like Alan Cox started to remark that the freeze really didn't seem to be in effect anymore.

However, with the release of 2.5.69, Linus said that all subsequent patches should be either very clear, or else should go through one or more of his lieutenants for approval. While not quite the same as a code freeze, this policy should help to put the brakes on some of the wilder kernel developments, and start to solidify existing code in preparation for a real push for stability.

With all the work on the scheduler, the block layer, the IDE driver, and other big areas, Linus has said he'd consider bumping the major number of the next stable series, to 3.0; but only if the advances made in the 2.5 series were enough of a departure from 2.4 to warrant such a thing. Since the first discussion of a possible 3.0 kernel, Linus has never mentioned it again, and consistently refers to a 2.6 series. Unless other developers feel strongly enough about the issue to petition Linus for a 3.0 version number, the next stable series is almost certain to be called 2.6. ■

■ No politics – just features

Linus made his position on DRM (Digital Rights Management) a bit clearer in a now-famous post to the Linux-kernel mailing list in April. He believes DRM to be just a feature like any other. So long as it is implemented and used in accordance with the GPL, he is willing to accept patches to support it.

In spite of a relatively quiet reception to his post, DRM is an extremely controversial issue. A number of laws under consideration in the US Congress relate to this issue, some of which could be interpreted as making it illegal to write or distribute free software. Many free software advocates believe DRM and related prospects to be purely evil, and would prefer not to support it in any way.

By stating his willingness to include DRM support in Linux, Linus has remained true to form, refusing to commit himself too strongly on any given political issue. Aside from the inherently political act of developing a free operating system kernel, Linus has always done his best in public to de-politicize all issues surrounding Linux. This has occasionally made him unpopular with Linux's own developers, as when he adopted the proprietary BitKeeper version control system.

In spite of these sometimes unpopular decisions, Linus has remained in control of Linux kernel development, partly because his stances on the various issues are always based on reasons that the other developers can respect at least. ■