

Zack's Kernel News

■ Adding RAM

Yasunori Goto has coded up a patch to allow users to hot-plug RAM chips on NUMA computer systems. Eventually he would like to port the patch to ia64 and ia32 architecture machines, but those are just a couple items on a very long To Do list.

Hotplugging has always been a big idea with kernel developers, even back in the day when the idea of plugging a new RAM chip into a running system was somewhat akin to the idea of flapping your arms hard enough to get off the ground and fly.

In some cases, of course, hot-plugging even a keyboard is just not feasible, because of the possibility of sparking a

power surge that could blow out sensitive components on the system. For those cases in which the hardware itself prevents hot-plugging, there is not much the software can do to mitigate the danger. (In practice, of course, keyboard and mouse hot-plugging is done routinely; nevertheless on some systems there is a realistic danger of breakage).

On modern systems, more and more hardware is being designed to plug in and out on the fly. Among other things, this poses the possibility of stackable systems, where to add a new drive or other component, you just stack it like a Lego piece onto the running system, and presto! Full functionality. ■

■ Superior encryption

Christophe Saout has written dm-crypt, a device-mapper target for handling filesystem encryption. According to Fruhwirth Clemens, the maintainer of cryptoloop, dm-crypt is a "vastly superior" mechanism than cryptoloop for handling filesystem-based encryption.

Fruhwirth has recommended that cryptoloop be maintained in 2.6 as a deprecated feature, just long enough to get dm-crypt fully operational. At that point, he feels the time will be ripe to remove cryptoloop entirely, and use loopback only for files. Andrew Morton seems to favor this approach in 2.6, with an idea of removing cryptoloop somewhere in the 2.8 time-frame.

Cryptoloop, itself a recent innovation, attempts to solve the problem of transparently maintaining encrypted data on an otherwise unencrypted filesystem. This feature has been hotly desired by security-conscious users for years.

While cryptoloop was initially hailed as a wonderful solution to the problem, perhaps its replacement, dm-crypt, will turn out to be just another step on the ladder as well. Almost certainly, any solution to the problem of encrypted data storage will be followed by many requests for further enhancements. ■

■ XFS shout loudly

XFS has been merged into the 2.4 tree at last. It almost didn't make it. When it became clear that 2.6.0 was imminent, Marcelo Tosatti decided to put the brakes on the 2.4 tree in a big way, refusing to merge any new features.

With such a sudden and unexpected cut-off, there were bound to be some shouts from disgruntled developers, and those in the XFS camp were among the loudest. Folks had been working hard to get XFS into the 2.4 tree, and had actually been working with some encouragement from Marcelo to get it in.

Eventually, Marcelo agreed to let XFS through the blockade, after asking some other developers to give it a thorough going-over. One of the biggest shouting-points during the public debate was the fact that ReiserFS, ext3, and the other major journaling filesystems had long been included in 2.4, and only XFS remained a separate patch.

After XFS' inclusion in 2.4, Marcelo also had to accept supplementary XFS patches over time; but the overall decision to drastically curtail incoming patches seems to be standing firm; and the XFS debate has undoubtedly played a big role in clarifying Marcelo's determination to follow through on that. ■

INFO

The Kernel Mailing List comprises the core of Linux development activities. Traffic volumes are immense and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls that take on this impossible task is Zack Brown.

Our regular monthly column keeps you up to date on the latest discussions and decisions, selected and summarized by Zack. Zack has been publishing a weekly digest, the Kernel Traffic Mailing List for several years now, reading just the digest is a time consuming task.

Linux Magazine now provides you with the quintessence of Linux Kernel activities straight from the horse's mouth.



■ New life

In a bizarre twist, Ian Kent has volunteered to take over maintainership of DevFS. Originally coded by Richard Gooch, DevFS has had a very painful life.

For years it remained an externally maintained patch, but was accepted by Linus, in a somewhat different form, into the 2.4 tree. Some developers have argued that it has congenital problems that can never be overcome, such as certain race conditions, and that the only answer is to remove it entirely and replace it with something else.

Most recently, leaving out some surprises along the way, Greg KH's udev code has been the favored replacement. But udev has not yet reached the stage where it could completely replace DevFS, and other similar projects have come and gone, all well-intentioned.

With Richard Gooch's departure from kernel development, an eventual expulsion of DevFS seemed to be a plausible scenario at any moment. Suddenly, however, it seems that several folks, including Ian, are willing to stand up for DevFS and take it under their wing.

Andrew Morton has said he'd be willing to accept DevFS patches into the 2.6 tree, if only to keep DevFS in some kind of working order until udev is fully able to replace it. But he acknowledges that there may be good reason to keep DevFS around even into the 2.8 cycle. ■

■ Linux Kernel 2.6

The 2.6.0 kernel was released in late December 2003, fulfilling Linus' hope of getting the next stable series out the door before the end of the year. This is the first time that his early predictions about the release schedule have come true. 2.0.0, 2.2.0 and 2.4.0 all took much longer than he or anyone else originally anticipated; and before the 2.0 series, the kernel development structure was even more chaotic with regards to release timings.

Undoubtedly, one reason why the 2.6 kernel was under better control than its progenitors is because of Linus' adoption of the proprietary BitKeeper revision control system, which continues to be a very controversial decision among kernel developers. However, success is a powerful argument, and BitKeeper remains entrenched at the very core of Linux development.

Another reason for the timeliness of the 2.6.0 release is undoubtedly Linus' hard-line restriction of patches. Feature freeze and code freeze held firm, with much less backsliding than in previous releases. And Linus was less afraid of reasserting his policies after they'd begun to flag than he had been in earlier days.

When it looked like the feature freeze might not stick, for instance, he didn't find it necessary to call another feature freeze before the code freeze; rather, he just treated the original feature freeze as having been in effect the entire time. While this might have seemed brutal to developers having their patches rejected while other code got in, it also had the effect of giving more weight to each successive policy change: developers had to assume that the official word might hold true at any moment.

In earlier stable series, Linus would have been much more likely to just completely abandon a feature freeze after he had back-slipped far enough to be embarrassing and so cause even more release delays; with 2.6 this was not the case. Of further interest in the 2.6.0 release is the handling of kernel maintainership, and the relationship between the current stable series and the current unstable series.

Traditionally, after the first release of each new stable series, there is a waiting period before the next unstable series forks off. During that period, only one kernel series is truly current. For all the rest of the time, both the stable and unstable series are developed in parallel.

In the case of 2.4 and 2.5 for instance, Marcelo Tosatti, the 2.4 maintainer, slowed development down to a crawl as soon as it became clear that the 2.5 tree had begun to crystalize into 2.6; now that 2.6 has been released, Marcelo is rapidly becoming less and less likely to accept patches that do not directly fix bugs. At the same time, it is likely that 2.6 development will progress relatively rapidly compared to the late days of 2.5, once the 2.7 kernel forks off. This strange symbiotic relationship between the stable and unstable kernel is one of practicality.

It is useful to have the official stable series be as up-to-date and featureful as possible, but this is hampered by the fact that a lot of wild development on the stable series would rapidly destabilize it. The solution that has emerged in the Linux kernel

world, has been to develop features in the unstable series, and then port them back to the stable series once they have stabilized.

In many cases this has proven to be possible with a minimum of destabilization, and it results in a more modern, advanced stable series containing all the features that the front line users require. Marcelo's decision to drastically reduce 2.4 development was in some sense a social one: to his thinking, there "should" be only one primary stable Linux kernel series. With the option of using the 2.6 kernel now available to users, he felt it was no longer necessary to keep the 2.4 tree as up-to-the-minute as possible thus forcing those users who want the new features to move to the later trees.

This attitude is also justified by the fact that, no matter how many features are back-ported from an unstable tree to a stable one, there will almost certainly be many enhancements and new features that cannot be back-ported, and even some bugs that cannot properly be fixed.

2.6 will certainly be a much better kernel than 2.4; when Linus forks 2.7 away from 2.6, we can expect Andrew and others to back-port a lot of features from 2.7 as they become stable. How this will play out as 2.7 begins to stabilize for 2.8, time will tell. ■

XTops