

ZACK'S KERNEL NEWS

Tracking Kernel Contributions

Wang Chen created a web page for kernel contribution stats. The page breaks down information by person and employer, as well as by change sets and number of lines of code affected. Also, it's possible to search for a given person or employer and see a graph of only their contributions. According to Wang's stats, for example, Yahoo employees have contributed much more to version 2.6.25 than to other recent kernels, whereas Microsoft employees are not listed as contributing anything at all.

There are still some problems with the engine. For example, if a person has contributed to the kernel using multiple email addresses, it still doesn't seem possible to see a graph representing the total contributions from all those addresses. During the course of discussion, a number of folks pointed out this and other related issues to Wang, so presumably the problem will be addressed.

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching ten thousand messages in a given week, 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 to take on this task is Zack Brown. Our regular monthly column keeps you abreast of the latest discussions and decisions, selected and summarized by Zack. Zack has been publishing a weekly online digest, the Kernel Traffic newsletter for over five years now. Even reading Kernel Traffic alone can be a time consuming task. Linux Magazine now provides you with the quintessence of Linux Kernel activities, straight from the horse's mouth.



Duck And Cover!

Matt Mackall was a bit shaken to discover a kernel error message that began "Treason uncloaked!" The message went on to say that the TCP code had detected a broken peer. Matt thought the message was alarming, so he posted a patch to change it. Herbert Xu's reply was, "What's next, you're going to remove 'printer on fire' as well? This message has been there for eons and is part of Linux lore." But Alan Cox said, "It was changed. The printer on fire bit was adjusted to make it clear because the original message *did* confuse and worry a few people." He felt Matt's patch was a good idea.

However, it turned out that Alan was not quite right. As Herbert pointed out via grep, the *printer on fire* message was still in the code. Alan remarked, "Interesting. It got changed and then the change got lost between 2.2 and 2.4 somewhere. It was changed to report 'lp0 reported invalid error status (on fire, eh?)' and it looks like that needs re-fixing." But David S. Miller said this just showed how much it didn't matter, and that having fun with kernel development was one of the best parts. And Matt said that because no one had printer ports anymore, this particular message probably wasn't so dangerous. But he added, the "Treason uncloaked" message was filling the bulk of his system messages.

Event Tracking

Alessandro Di Marco has added some activity counters to the Linux kernel input drivers, which will be useful for a variety of event notifications, including intrusion detection. His code is in part a re-envisioning of SIN, a kernel module he'd written a couple years ago to provide similar behavior. On the mailing list, there was no discussion about these patches or about any differences between them and SIN. Alessandro did say that the SIN design had fundamental weaknesses that made it wiser to redo the whole project, in spite of the fact that he's kept SIN up-to-date and actively maintained all this time.

Hardware Compatibility

Adam Osuchowski was poking around in the deep dark places of the kernel and came upon some hard-coded assembly that used the *xadd* instruction. Because the 386 CPU didn't implement an *xadd* instruction, Adam asked whether Linux still supported the 386. The *xadd* instruction turned out to be just a bug, but the incident sparked a discussion about which older systems were and were not supported under Linux.

In terms of systems supporting Symmetric Multi-Processing (SMP), Alan Cox remarked that the first system to support Intel's MP standard was the 486 with external APIC; he reckoned those would be the oldest systems capable of running SMP Linux, although he thought the Earth might have been denuded of such systems long ago. Maciej W. Rozycki commented, "I failed to track down a single 486 SMP system that would adhere to the MP spec. There were and possibly still are APIC-based 486 SMP systems out there, but most likely they are not Intel MPS-compliant, by not providing the MP header at the very least. Thus Linux would have to be ported and I gather the interest in doing so is epsilon. Myself, I could not resist trying an APIC-based 486 SMP box and possibly fixing issues if I found one and it was MPS-compliant, but nothing beyond that I would say. Life's too short."

In terms of the 386, various people speculated but no one could say for sure whether Linux would run on them. Jan-Benedict Glaw said he had an old, still-functioning 386 that he'd dug out of storage, and that, "... it still powers on and boots up that ancient Debian version. Using a 20GB (right, gigabytes) HDD." He said he might try experimenting with more current kernels and see whether they worked. Various other folks pointed out that 386 CPUs were still used in various embedded systems, and Ingo Molnar remarked that he knew of someone who occasionally booted up a 386 with current kernels. So apparently the 386 is still kicking.

Further Networking Security

Michael Stone proposed implementing a kernel feature to allow a process to irrevocably give up the ability to perform unrestricted network I/O, not just for itself but for all its future child-processes and their descendants. He said this would let users protect themselves from a whole big batch of evil software. Andi Kleen replied that this could be partially done, at least with outgoing packets, using netfilter. Michael didn't like this idea, though, and clarified a number of issues surrounding what he wanted this thing to do – the main points being that it should be process-centric and easy for distributions and vendors to adopt. He and Andi went back and forth on the technical merits of netfilter's ability to support these requirements, but at one point Alan Cox pointed out that there were ways for a process to break out of any such restrictions and that, really, more fine-grained security solutions like SELinux were the way to go.

Tux3 Compatibility Status

Daniel Phillips announced that Tux3 was abandoning compatibility with older kernels, and going forward it will compile only on kernels newer than 2.6.28. Tux3 would, however, continue to track the current version of the Linux kernel git repository. Tux3's own development continues to take place using Mercurial (<http://www.selenic.com/mercurial/wiki/>), a version control system that came out at about the same time as git, and seems to be the only free version control system able to compete with git in terms of speed and distributed development features.

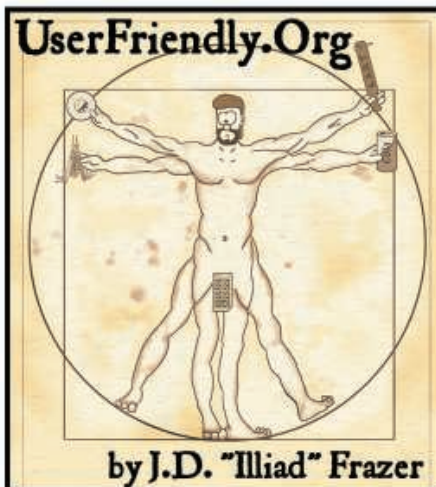
Getting back to Tux3, Daniel has announced that he is about to embark on implementing atomic commits, and so the code will be in a period of instability while that happens.

In any event, Tux3 is very much a development project, and should only be used by people who really enjoy testing filesystems.

Status Of SquashFS

Phillip Lougher is submitting SquashFS for inclusion in the main kernel tree again. The last SquashFS submission was in 2005, and he's made significant progress since then, including two major revisions and a nearly complete rewrite of all kernel-based code. Among other changes, the filesystem is now 64-bit, supporting fantasmagorically large files, and it presents the traditional '.' and '..' directories in directory listings.

With alternatives like CramFS already in existence, Phillip's earlier attempt to submit his code met with resistance, partly on the grounds that we don't really need more than one compressed filesystem. This time, Phillip had prepared a number of counters to that argument. In filesystem size, file size, block size, and various other measures, SquashFS dramatically exceeds CramFS's capabilities. And as Phillip pointed out, CramFS is currently listed as orphaned, while SquashFS is actively maintained.



Linux Magazine Exclusive