# ZACK'S KERNEL NEWS

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.

## New Generic Initramfs Project

Jeremy Katz has announced the Dracut project, which is essentially an implementation of initramfs that could be adopted by all Linux distributions. When the kernel boots, it must first create a filesystem in RAM before it can mount the real filesystems and transition the running kernel to those. The specific way it does this is a strange and esoteric sequence of events, implemented by an initramfs tool. Each Linux distribution creates its own way to accomplish these various tasks, and each initramfs has a different set of special super-excellent features that allow things like interrupting the boot process to perform a particular action at a particular point in that process.

Jeremy sees no need for all the distributions to labor over maintaining their own initramfs infrastructure. His proposal is to take all the complexity out of the tool and make something totally generic that every distribution can use and contribute to. Any other special boot-time features could be done immediately after initramfs finished the essential task of transitioning to the root filesystem on the device.

One reason such a simplified initramfs would be especially useful, Jeremy says, is that it would bring a five-second boot time into the realm of possibility if a particular distribution wanted to skip all but the most absolutely essential steps. Current initramfs implementations have so much heavy baggage that a boot-up might have to go through several unnecessary steps on a given system just because other systems using the same initramfs do find those steps necessary.

As Christoph Hellwig pointed out, Hannes Reinecke is also working on reimplementing initramfs, but whereas Jeremy had been basing his work on Red Hat's implementation, Hannes had been working off of openSUSE. Christoph suggested the two of them pool their efforts. Hennes said he was all in favor of this, but he and Jeremy didn't discuss it on the list.

Neil Horman pointed out that a stripped-down version of initramfs would not necessarily solve all the major problems. He pointed out that some embedded systems never bothered to transition the kernel to any other root filesystem and, for stability reasons, spent the entire session in the initramfs. Any implementation of initramfs that didn't accommodate that and similar uses would leave out important groups of users, and Loïc Grenié agreed.

Theodore Ts'o also added, "Part of the discussion at the Kernel Summit, and also what David Jones was looking to work at, was to do something that could be included as part of the kernel sources. The idea is that as responsibility for early boot is moved from the kernel, an mkinitramfs which is fixed and distributed by the distribution might not work with a newer kernel.org kernel. So the idea that was explored was adding a common mkinitramfs with basic functionality into kernel sources, with the ability for distributions to add various 'value add' enhancements if they like. This way if the kernel wants to move more functionality (for example, in the area of resuming from hibernation) out of the kernel into initramfs, it can do so without breaking the ability of older distributions from being able to use kernel.org kernels."

## New Patch Tracker

John Hawley announced *http://patchwork.kernel.org/*, a page that tracks patches sent to a given mailing list and presents them with a nice clean format. The idea is that maintainers of various parts of the kernel – or other projects – can then change the status of each patch to make it easier to track what's happening with development. So far, a number of kernel-related mailing lists are on the page, including ACPI, the build system, and the linux-kernel mailing list.

## Git Doc For Beginners

David Howells has written a simple guide to git usage for beginners – essentially his own list of git recipes converted into written explanations. Although quite a bit of git documentation is available, David's work, titled *The Git Hater's Guide To The Galaxy,* might find a unique niche among certain users. The guide is intended for people who'd rather not use git, but who recognize that they do need to learn it well enough to do some basic tasks. Whether this doc will continue to exist as a standalone file in the kernel source tree or be parceled out amongst various other docs, giving to each whatever it has that they lack, remains unclear.

## Video-4-Linux Docs Made XHTML Compliant

Németh Márton updated the V4L documentation to be XHTML compliant. A benefit of this is that the file can be parsed and any useful data snarfed into some other application. I'm not sure whether any other applications would have use for that, but Németh has made the possibility available.

## Tracking Pesky Daemons

Scott James Remnant has written a bit of code to shed some light on daemon behavior. Currently, a process daemonifies by forking a child and then exiting. The child is then reparented to the init process (the parent of all processes), where the child detaches itself from the terminal that invoked the daemon. The child then forks again and exits, essentially leaving a grandchild of the original process. This grandchild is reparented to the init process and performs all the tasks ofthe daemon. This happens every time you invoke a daemon: Fork, fork, and the resulting child sits on the system as a child of init.

The problem Scott wanted to address was that this daemon has been so completely disconnected from the process that started it, there is no way for init to associate it with that process. When the daemon dies, init will have no way of knowing that this was the process associated with the invocation that happened earlier. Among other issues, this makes it difficult to create meaningful system logs. It would be nice to be able to go through the logs and see the different parts of a service all logged together.

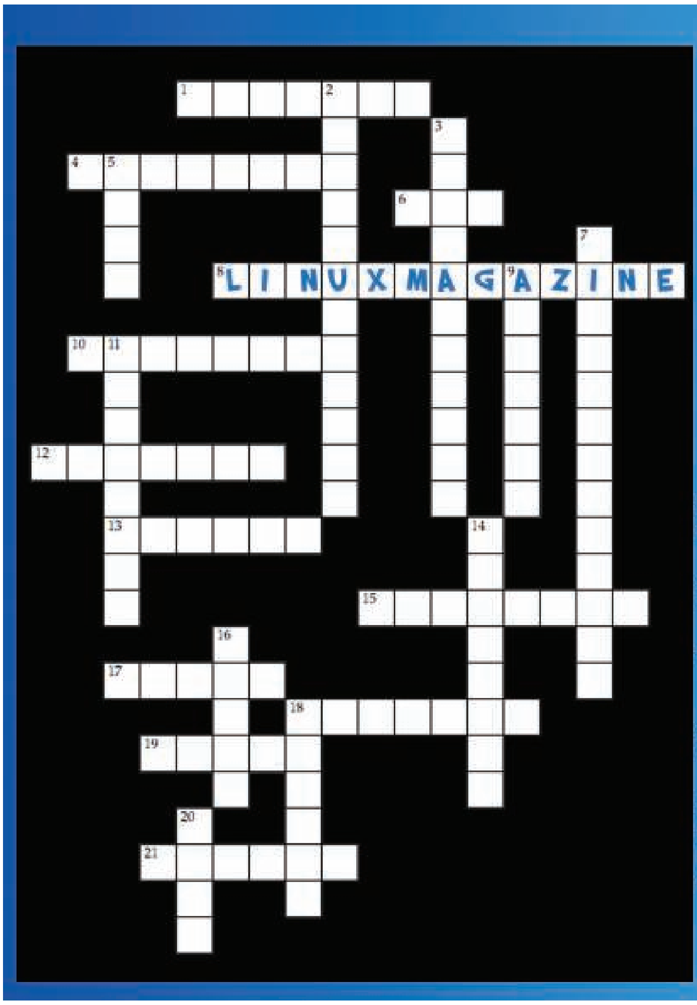
One way Scott found to keep track of the relationships between the original process and the daemon it leaves behind is to have init do a *ptrace()* on all processes; however, this has a number of drawbacks, including security issues, as well as causing some software to change its behavior precisely because it's running in *ptrace()*! No good.

The solution Scott proposes is to create a couple of new signals, specifically to handle this reparenting-to-init situation, that will carry information to init about the lineage of the children being reparented to it. Scott says this wouldn't require any changes to user code or cause software to behave differently under Linux than it would under other operating systems; it would just keep track of the reparenting side of things in a way that lets init associate child processes with their dead parents.

## 64-Bit Systems

Nick Warne has been running Slamd64 (*http://www.slamd64.com/*), an x86 64-bit system, for a while and thought that the time was ripe for a new framebuffer boot logo for the system. He submitted his own creation: *http://www.nick.ukfsn.org/stuff/logo_linux64_clut224.ppm*. Sadly, it wasn't tremendously well received. Some folks just didn't like it, whereas others had logos of their own to suggest. Markus Reichelt posted a link to his own creation at: *http://mareichelt.de/temp/logo_linux_clut224.ppm*. Folks reaching consensus didn't seem likely, but it does help illustrate the progress of 64-bit x86 systems lately.

**Linux Magazine Exclusive**



DOWN

2 UserFriendly's younger kernel hacker
3 Ewing and Young
5 XO-1 developed for this project
7 Like a very smart Mexican squirrel on speed
9 Gifted Welshman
11 Originally Software und System-Entwicklung
14 Plan 9 from _____
16 Clippy for vi
18 Originally Enoch Linux
20 Linux, Apache, mysql, perl/php/Python

ACROSS

1 Runs from a CD
4 The QPL is not an example of this
6 Founded by Ettrich
8 International Edition of this Publication
10 101010
12 Svelte, buoyant seafowl
13 Zulu's "humanity"
15 Master of GNU
17 Linux replaced this
18 The bane of S.C.O.
19 Desktop or Garden Statue
21 Koolu CTO

UserFriendly.Org