ZACK'S KERNEL NEWS

New Buffer Layer Code

Nick Piggin has gone into the deep dark realms of the kernel and rewritten the buffer layer. This part of the kernel interacts with block devices to keep track of a variety of key information, such as the I/O status of disk blocks. But over the years, the code has started to show its age. It's never been a darling of kernel developers, and some folks have advocated taking the buffer layer out entirely.

Nick's approach has been to write a complete replacement, called *fsblock*, that fixes a wide range of problems – or at least shortcomings – that have been ongoing with the buffer layer for years. All of these improvements are fairly esoteric, just because of how deep into the kernel the buffer layer lives.

One improvement has been support for large block sizes. The current buffer layer has a complex and messy implementation for this, whereas *fsblock*'s approach is much cleaner and could support in a straightforward manner block sizes larger than a gigabyte, if desired. In another improvement, the code doesn't deadlock. The current buffer layer tends to deadlock because of its memory-allocation handling; *fsblock*'s cleaner approach avoids this.

Nick has gotten a mixed reaction to his work so far. Some folks, like William Lee Irwin III and Chris Mason, feel that this work is long overdue and much welcome. Others, like Jeff Garzik, think it's too soon to start celebrating. Jeff's feeling is that problems are still lurking in the problem space that Nick just hasn't hit yet. One reason for this is that Nick has only converted a single filesystem – Minix – over to *fsblock*.

Minix is a relatively simple and straightforward filesystem and probably hasn't stressed *fsblock* too far. The real test will come with ext3/4, ReiserFS, and other complex journaling filesystems. Nick agrees wholeheartedly that this could be the case. Perhaps the harshest criticism came from Christoph Hellwig, who argued that the buffer layer code wasn't really that old, and the improvements Nick made weren't really that important.

On the whole, folks do seem to feel as though Nick's work is at least heading in the right direction and that the existing buffer layer is a mess that should really be replaced. But it might be many kernel versions before Nick's work is solid enough for inclusion and all the filesystem code that will need to comply with it has been updated.

New VESA Driver

Michal Januszewski has announced *uvesafb*, a generic driver for VBE2+-compliant video cards. Based on *vesafb* and *vesafb-tng*, this driver relies on a userspace helper application that accesses the video BIOS to take advantage of the standard refresh rate controls, video mode changes, and the rest. Currently it's been written for x86 machines only, but Michal says it should be straightforward to extend the code for non-x86 systems.

New Open Sourced Touchscreen Drivers

Ondrej Zary tapped into the data streams of the IRTouchSystems touchscreen to figure out its behavior. Unitop, the manufacturer, only releases closed source drivers, so Ondrej produced his own free software driver on the basis of the results of his analysis. He submitted the patch to the linux-kernel mailing list, but there was no discussion surrounding it. Another day, another driver.

Meanwhile, Cyril Hrubis has written a new driver for the touchscreen on the Sharp Zaurus SL-5500 PDA. Pavel Machek actually posted the patch, giving it his endorsement, saying it was "partly based on *ucb1x00-ts.c*, but this one actually works."

New collectl Performance Monitoring Tool

Mark Seger announced that he's recently open sourced a utility he's been working on for about four years. The utility, called *collectl* (available at *http://collectl.sf.net*) is a performance monitoring tool that's able to keep track of a wide variety of statistics and display them in an array of formats. It's designed to present human-readable output, but also to interoperate somewhat with tools like Gnuplot and Excel. Because *collectl* has been in use for years, Mark says it should be fairly robust and reliable, although a thorough pounding from a new community of users is likely to identify any bugs that remain.

Intel Open Source Drivers

Anil S. Keshavamurthy, from Intel, has posted a patch to support the upcoming Intel IOMMU hardware, also known as Virtualization Technology for Directed I/O. It's great to see this kind of proactive support from Intel, but various folks, including Andrew Morton, have some concerns with the code in its current form. They're worried it might be too slow, but it does seem as though there is plenty of support for Anil's work, and the new hardware will undoubtedly have full Linux support in the near future.

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.