

NOTICIAS DEL KERNEL

Nuevo Código de Capa de Búfer

Nick Piggin se ha adentrado en las oscuras profundidades del kernel y ha reescrito la capa de búfer. Esta parte del kernel interactúa con los dispositivos de bloque para llevar la cuenta de diversas informaciones clave, como el estado I/O de los bloques de disco. Pero, con el transcurso de los años, el código acusaba la edad. A pesar de que nunca ha sido una perita en dulce para los desarrolladores del kernel, algunos chicos han abogado por renovar completamente la capa de búfer.

La estrategia de Nick ha sido escribir desde cero un sustituto, denominado *fsblock*, que arregla un amplio abanico de problemas y defectos que se mantenían en la capa de búfer desde hace años. Todas estas mejoras son bastante esotéricas, ya que esta capa vive en la auténtica profundidad del kernel.

Una de esas mejoras ha sido el soporte para bloques de gran tamaño. La capa de búfer actual tiene una implementación compleja y confusa en este punto, mientras que la solución de *fsblock* es mucho más limpia y puede soportar de manera directa bloques grandes mayores de un gigabyte, en caso necesario. Otra mejora es que el código no provoca bloqueo mutuo. El búfer actual tiende a producir bloqueo mutuo debido a su administración de memoria. La limpia solución adoptada por *fsblock* evita esto.

Nick ha obtenido división de opiniones hasta el momento. Para algunos, como William Lee Irwin III y Chris Mason, este trabajo se había pospuesto en exceso y es bienvenido. Para otros, como Jeff Garzik, es demasiado pronto para echar las campanas al vuelo. La sensación de Jeff es que aún existen problemas sobre la mesa que Nick aún no ha abordado. Un posible motivo es que sólo ha convertido un sistema de archivos, Minix, a *fsblock*.

Minix es un sistema de archivos relativamente simple y probablemente aún no

ha estresado lo suficiente a *fsblock*. La verdadera prueba vendrá con ext3/4, ReiserFS y otros sistemas de archivos con journaling. Nick está completamente de acuerdo en que ese podría ser el caso. Quizá la crítica más dura vino por parte de Christoph Hellwig, que expuso que el código de la capa de búfer no era en realidad tan antiguo, y que las mejoras que ha hecho Nick en realidad no eran tan importantes.

En resumen, alguna gente parece pensar que el trabajo de Nick al menos se dirige en la dirección correcta, y que la capa de búfer existente es un lío que verdaderamente debería ser resuelto. Pero puede que pasen varias versiones del kernel antes de que veamos si el trabajo de Nick es suficientemente consistente para incluirlo y si se genera todo el código de los sistemas de archivos que faltan por completar. ■

Nuevo Driver VESA

Michal Januszewski ha anunciado *vesafb*, un driver genérico para tarjetas de vídeo compatibles con VBE2+. Basado en *vesafb* y *vesafb-tng*, este driver confía en una aplicación asistente en espacio de usuario que accede a la BIOS de la tarjeta de vídeo para aprovechar los controles de tasa de refresco estándar, cambios de modo de vídeo y el resto de configuraciones. De momento se ha escrito sólo para máquinas x86, pero Michal afirma que extender el código para sistemas no-x86 es prácticamente directo. ■

Nuevos Drivers Libres para Pantallas Táctiles

Ondrej Zary ha aprovechado los flujos de datos de las pantallas táctiles IRTouchSystems para deducir su comportamiento. El fabricante, Unitop, sólo distribuye drivers de código cerrado, por lo que Ondrej ha generado su propio driver bajo software libre basándose en los resultados de sus análisis. Ha presentado el parche en la lista de correo del kernel, pero no hubo discusión al respecto. Un día más, un driver más.

Mientras, Cyril Hrubis ha escrito un nuevo driver para la pantalla táctil de la PDA Sharp Zaurus SL-5500. Ha sido en realidad Pavel Machek quien ha postado el parche, dándole su apoyo al afirmar que estaba “basado parcialmente en *ucb1x00-ts.c*, pero que éste funcionaba de verdad”. ■

Nueva Herramienta de Monitorización Collectl

Mark Seger anunció que había pasado recientemente a software libre una utilidad en la que venía trabajando desde hacía cuatro años. Esta aplicación, denominada *collectl* (disponible en <http://collectl.sf.net>), es una herramienta de monitorización del rendimiento que es capaz de llevar la cuenta de una amplia variedad de estadísticas y las muestra en un conjunto de formatos. Está diseñada para generar resultados fácilmente legibles por las personas, pero también para interoperar de alguna manera con herramientas como Gnuplot y Excel. Como *collectl* se ha venido usando durante años, Mark dice que debería ser bastante robusta y fiable, aunque el minucioso análisis de una nueva comunidad de usuarios seguramente lleve a identificar bugs que aún puedan existir. ■

Drivers Intel bajo Software Libre

Anil S. Keshavamurthy, de Intel, ha postado un parche para dar soporte a nuevo hardware Intel IOMMU, también conocido como Virtualization Technology for Directed I/O. Es fantástico comprobar este tipo de soporte proactivo desde Intel, pero varias personas, incluido Andrew Morton, tienen algunas objeciones a este código en su estado actual. Están preocupados de que pueda ser demasiado lento, pero parece que existe pleno apoyo al trabajo de Anil, y el nuevo hardware tendrá, indiscutiblemente, soporte completo bajo Linux en un futuro próximo. ■