

NOTICIAS DEL KERNEL

Permitir Schedulers Arbitrarios

Pankaj Parakh quería probar su propio scheduler de procesos en Linux y preguntó cómo podía conectar el suyo para reemplazar el scheduler primario. En realidad, lo que Pankaj quería era crear una infraestructura completa dentro del kernel para facilitar a cualquiera el conectar su propio scheduler.

Peter Williams contestó que estaba en el proceso de resucitar el proyecto de plugin de scheduler para la CPU y que lo iba a renombrar como CPU_PISCH. Invitó a Pankaj a unirse, y Pankaj aceptó. De manera que Peter le envió algunos parches y remarcó que también estaba trabajando en un nuevo scheduler por su cuenta, que esperaba completar antes de lanzarlo como parche.

La lista de correo del kernel de Linux comprende lo principal de las actividades de desarrollo de Linux. El volumen del

tráfico es inmenso, alcanzándose a menudo los diez mil mensajes semanales. Mantenerse al día de todo lo que sucede en el desarrollo del kernel es casi imposible para una sola persona.

Sin embargo, Zack Brown es uno de los pocos valientes que lo intentan, y a partir de ahora podrá leerse lo último de las discusiones y decisiones con respecto del kernel de Linux llevados de la mano de este experto.

Zack ha publicado un resumen online semanal llamado "The Kernel Traffic Newsletter" durante cinco años. Linux Magazine te trae ahora la quintaesencia de las actividades del kernel de Linux del mayor especialista en el tema.



Es algo magnífico ver un proyecto como éste: es divertido facilitar a la gente que pueda experimentar con el comportamiento fundamental del kernel. Linus Torvalds dijo que sólo quería un único scheduler en el kernel, por lo que este proyecto podría ser sólo un parche por mucho tiempo. Si se hace muy popular y consigue muchos contribuyentes, Linus podría en última instancia reconsiderar su postura. Pero como funcionalidad, parece improbable que CPU_PISCH tenga interés para el usuario final. El tiempo lo dirá. ■

<http://lkml.org/lkml/2009/10/11/253>

<http://lkml.org/lkml/2009/10/27/50>

Mantenimiento de Kbuild

Sam Ravnborg ha dejado de ser el mantenedor de Kbuild. La noticia fue acogida con consternación por gente como Andrew Morton, Theodore Y. T'so, David S. Miller, Randy Dunlap, Stephen Rothwell y muchos otros que agradecieron a Sam el haber sido el mantenedor durante tanto tiempo. Sam no había hecho planes para el nuevo mantenedor, pero añadió que "ya no se reciben muchas quejas acerca de kbuild actualmente. Todas las funcionalidades básicas funcionan".

Aníbal Monsalve Salazar se presentó voluntario para mantener Kbuild. Michal Marek también comentó que quería hacerlo y sugirió que los dos podrían trabajar juntos. A Aníbal le gustó la idea, y parece que Kbuild tiene ahora dos nuevos co-mantenedores. ■

<http://kbuild.sourceforge.net/>

Estado de LogFS

LogFS ha hecho recientemente algunos avances importantes. El sistema de archivos flash está orientado a dispositivos más grandes que los que su compañero JFFS2 puede soportar. Pero LogFS había llevado a cabo algunos cambios que habían hecho que fuese menos bienvenido en el kernel oficial – en concreto, cambios en el formato de

información en disco necesarios. Cuando un sistema de archivos que ya existe en el kernel cambia su formato de disco, los usuarios existentes del viejo formato aún tienen que ser soportados, y esto fuerza a que el kernel mantenga múltiples bases de código para cada versión del sistema de archivos que cambie su formato. La última vez que los desarrolladores de LogFS intentaron introducirlo en el kernel principal, Linus Torvalds les dijo que volvieran cuando el formato de disco se hubiese estabilizado.

Recientemente, Jörn Engel hizo justamente eso. Posteo a linux-kernel diciendo que el formato de disco ya era estable, que se habían realizado otras mejoras, y que quería enviar el código para su inclusión. Arnd Bergmann comentó que el código tenía una pinta estupenda, y que la fusión era necesaria. Jörn pidió a Stephen Rothwell que añadiera LogFS a la rama Linux_Next, y Stephen así lo hizo, apuntando por lo tanto dos parches para la próxima ventana de fusión de Linux. ■

<http://logfs.org/logfs/>

Relicenciar LTTng

Mathieu Desnoyers anunció recientemente que la versión 0.164 de LTTng (Linux Trace Toolkit Next Generation) implicaría volver a licenciar el grueso del código para quedarse como GPL: ahora también incluye LPGL. Parte del código pasará a ser de licencia dual con GPL y BSD. Mathieu estaba aún esperando permiso de IBM para incluir algunas de sus contribuciones en la nueva licencia. ■

<http://ltd.polymtl.ca/>

Syscalls bajo Alpha

Michael Cree estaba molesto por el alto número de llamadas al sistema no implementadas en la arquitectura Alpha: más de una docena. Se ofreció voluntario para implementar algunas de ellas, pero no estaba seguro de cuántas.

les iban a ser las más sencillas, y no sabía si la tabla de llamadas al sistema debía mantenerse en el mismo orden que con otras arquitecturas. Matt Turner comentó que él también había considerado implementar estas llamadas al sistema, pero que también tuvo dudas técnicas similares. Parece que tienen intención de trabajar juntos. ■

SysFS Dinámica

Matthew Wilcox ha postado parches para permitir que SysFS se llene dinámicamente. La manera estándar de

SysFS es ir codificada en la rama de fuentes, pero Matthew fue capaz de usar sus parches para conseguir un sistema arrancable. Advirtió a todo usuario interesado que estuviese atento a condiciones de carrera y funcionalidades ocultas, y urgió a los contribuyentes interesados que fuesen cuidadosos en todos sus cambios en la implementación.

Claramente SysFS es una parte muy profunda y peligrosa de la configuración del kernel, y una implementación sólida es crucial. Tradicionalmente, SysFS ha

sido la única de muchas alternativas capaz de ejecutar la configuración del kernel en tiempo de ejecución. Se ha intentado con ioctls, ProcFS y otras infraestructuras. SysFS es uno de los principales contendientes para reemplazar a todos los demás, e indudablemente es necesario mucho cuidado al diseñar funcionalidades como esta. ■

<http://lkml.org/lkml/2009/10/20/28>
<http://lkml.org/lkml/2009/10/20/32>

Actualizar el Master Server del Kernel

Mientras muchos desarrolladores de kernel dormían en Japón durante el kernel summit, John Hawley decidió deslizarse en la oscuridad de la noche y actualizar el sistema operativo en *master.kernel.org*. Esperaba que no llevara más de seis horas y requeriría un reinicio. Supuso que la carga del servidor sería suficientemente baja. Logró actualizar, y, a pesar de algunos pequeños problemas, lo completó dentro de la ventana de seis horas. ■

<http://events.linuxfoundation.org/archive/2009/kernel-summit>
<http://lkml.org/lkml/2009/10/19/44>

```

From Bartlomiej Zolnierkiewicz <>
Subject Re: removing existing working drivers via staging
Date Thu, 15 Oct 2009 19:42:40 +0200

On Thursday 15 October 2009 18:47:26 Greg KH wrote:
> On Thu, Oct 15, 2009 at 09:39:51AM -0700, david@lang.hm wrote:
> > however, what I think I saw proposed was to move drivers that need to be
> > 'cleaned up', to staging and then dropping them if they don't get cleaned.
>
> What is "proposed" is the following:
>
> - For drivers currently in the kernel tree, that the subsystem
> maintainer, for whatever reason, feels is obsolete / broken /
> needs major cleaning / wants to get rid of, can be submitted
> to the staging maintainer to be moved to the drivers/staging/
> directory.

This is insanity and opens a door for various forms of abuse.
    
```

Figura 1: Algunos se resisten a la eliminación de drivers.

