

NOTICIAS DEL KERNEL

Nueva Implementación FIFO in-Kernel

Stefani Seibold ha reimplementado completamente la nueva API de kfifo (first in, first out). La nueva implementación pone un front-end genérico al código que implementa toda una gama de fifos en función de lo que los desarrolladores del kernel necesitan en su propio código. Para tener esta flexibilidad aún manteniendo un entorno de ejecución rápido, Stefani implementó gran parte del código en macros C que se resuelven y simplifican en tiempo de compilación, por lo que el código del kernel que se produce verdaderamente es sólo lo que se necesita, en lugar de forzar al kernel en ejecución a hacer un conjunto de elecciones fifo cada vez que se usa uno.

Entre otros problemas, Stefani reconoce que las macros que ha creado suelen ser

muy complejas y difíciles de mantener. Pero de acuerdo con sus pruebas, el resultado generado por el compilador es tan bueno como el código máquina escrito a mano. En última instancia, la implementación de Stefani haría del kernel como conjunto algo mucho menos complicado debido a que los programadores de drivers no estarían forzados a implementar sus propias fifos sobre el kernel. Parece que esta es la clase de cosa que típicamente sería bienvenida en el kernel. Veremos si Linus Torvalds cree que tener código complejo pero que se ejecuta rápido es mejor que tener código claro y limpio pero un poquito más lento. ■

Soporte de Backlight más Genérico

Rafal Milecki anunció su intención de rehacer completamente la clase del dispositivo backlight. El problema era que el código actual era demasiado inflexible y simplemente no podría controlar la variedad de pantallas y configuraciones que la gente quería usar. El método de Rafal era “imaginar el portátil más loco con configuraciones de pantalla externas que pudiésemos tener”.

Henrique de Moraes Holschuh fue muy entusiasta con este método, pero incluso encontró que Rafal era demasiado reticente a solucionar todas las cosas raras susceptibles de solucionar. Henrique comentó que “si vas a diseñar algo, hazlo adecuadamente desde el primer día. Soluciona todos los problemas principales: múltiples backends para cada dispositivo, autoselección del backend por defecto en función de la información reportada por los drivers del backend [...], *y* dispositivos múltiples con conexión en caliente”. Y añadió que “esto no será, después de todo, sobreingeniería para el futuro. Es un problema que tenemos ahora con los dispositivos del mundo real que se están fabricando”. Los dos discutieron los detalles técnicos durante un tiempo, y finalmente se les unieron Matthew Garrett y Zhang Rui. Henrique había puesto también en copia al

mantenedor de backlight Richard Purdie en su respuesta original a Rafal, pero Richard no participó en el debate.

Parece que ciertamente hay suficiente interés en limpiar el código de la clase del dispositivo backlight. Presumiblemente, de todo esto resultará algo fantástico, e imagino que Richard estará también contento. ■

Nueva Infraestructura de Testing

Alguien bajo el nombre de Joe Moo posteo un anuncio del proyecto The Linux Testers. El objetivo es ayudar a los usuarios a descargar el último kernel de Linux, compilarlo y probar varias funcionalidades. Este proyecto ayudaría a los usuarios a probar todos los parches incluidos en el kernel entre lanzamientos oficiales. En teoría, ampliar este tipo de pruebas a un gran número de usuarios habituales incrementaría enormemente la velocidad a la que se encuentran fallos. ■

Posible Violación de la GPL

Gerhard Wiesinger tenía la impresión de haber descubierto una manera en la que VMware ha violado la licencia GPL. Específicamente, pensaba que había encontrado evidencias de que su servidor ESX, que es de código propietario, estaba basado en código libre de Linux. Apuntó a su propia documentación en la que dice que el driver BusLogic estaba basado en el driver ncr53c8xx versión 3.4.3b de Linux y en otro basado en la versión 1.7.3 del driver sym53c8xx de Linux.

Florian Weimer sugirió que esos drivers también podrían haberse lanzado con licencia FreeBSD, lo que podría permitirles relicenciar bajo licencias propietarias a discreción de VMware. Pero Gerhard contestó que las versiones específicas de los drivers usados por VMware tenían código exclusivamente GPL, y que por tanto, como el código se había enlazado al propio servidor ESX, todo el servidor ESX se tendría que volver a publicar bajo GPL por parte de VMware.

El debate acabó en este punto, y es interesante recordar que sin abogados y jueces,

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.

las acusaciones solas no llegan muy lejos. Pero es bueno reportar los verdaderos esfuerzos de miembros de la comunidad para mantener a la industria rindiendo cuentas ante la GPL. ■

Ajuste Fino de Gestión de Energía

Salman Qazi de Google comentó que parte del código de gestión energética de Google que está en funcionamiento en sus datacenters podría también ser aplicado a portátiles y teléfonos móviles. Preguntó cómo podría hacerse la subida del código al kernel.

La idea básica del trabajo de Salman era desarrollar un ajuste fino sobre dónde dirigir la electricidad disponible dentro de un datacenter, ubicando “capas de energía” en máquinas individuales de forma dinámica. De esta manera, conforme cambia la demanda de los usuarios a lo largo del día, las máquinas que necesitan menos energía pueden pasar a un modo de bajo consumo, mientras que a las máquinas con más demanda se les podría asignar una parte mayor de la electricidad disponible. En última instancia, esto permitiría a Google poner más máquinas en sus datacenters sin

tener que preocuparse de que los picos de demanda requieran más energía para operar que la verdaderamente disponible.

Salman señaló que los usuarios habituales podrían también beneficiarse de este código. Si un usuario tiene un portátil que necesita estar encendido el mayor tiempo posible usando sólo su batería, explica Salman, y el usuario también quiere ejecutar algunas tareas específicas con el portátil, como ver una película, el usuario podría usar este código para reducir la energía hasta justo el punto en el cual el vídeo comenzase a perder calidad al visualizarse. Entonces el usuario podría estar seguro de que el consumo de la energía ha sido el menor posible, aún visualizando la película con buena calidad.

Salman quería calibrar el interés entre los desarrolladores del kernel y aprender la mejor manera de enviar los parches, ya que era en realidad una cantidad de código considerable, y no quería simplemente soltarlo sin más, sin sondear el terreno.

Andi Kleen dijo que se habían discutido ideas similares en el pasado, aunque quizás no fueron tan sofisticadas como las que describía Salman. Pero señaló que algunos elementos de esas ideas ya estaban presen-

tes en el kernel de la rama principal. Sugirió leer las discusiones anteriores acerca de estos asuntos.

Theodore Y. T'so también remarcó que “la diferencia principal es que Google quiere la capacidad para controlar el sistema de energía/envolvente térmica desde espacio de usuario, a diferencia de dejar que el hardware haga una petición en una situación de emergencia. Esto tiene sentido si estás intentando balancear los requerimientos de energía/térmicos a lo largo de un gran número de sistemas, a diferencia de responder a una emergencia energética/térmica local señalada por el firmware de la plataforma”. Un buen grupo de históricos del kernel entraron en la discusión, con extensas consideraciones acerca de diversos complicados algoritmos. Los muchachos del kernel tienen un gran número de ideas acerca de lo que piensan que sería apropiado y bueno para esta clase de funcionalidad en el kernel, mientras que la gente de Google tiene una gran cantidad de código ya programado. Cómo se complementan estas dos situaciones, aún está por verse. Podría ser que porciones de código de Google fuesen finalmente incorporadas. ■

