

# NOTICIAS DEL KERNEL

## Nueva Arquitectura del Kernel

Michal Simek ha creado una nueva arquitectura de Linux para dar soporte a la CPU Microblaze. Preguntó cuáles serían las funciones del mantenedor y si podía tener su propio repositorio git en kernel.org para albergar las ramas. Respecto al repositorio git, Jan Engelhardt sugirió que un repositorio en kernel.org no era realmente necesario. En lugar de esto, aconsejó mantener un repositorio separado, lo que además ofrecería a Michal más libertad para administrarlo a su antojo.

Stefan Richter estuvo de acuerdo en que, en términos de flujo de trabajo, no tiene mucho sentido elegir un repositorio con base en kernel.org frente a cualquier otra solución de hosting, pero señaló que había algunas ventajas en albergar el repositorio en kernel.org. Para empezar, los objetos git podrían compartirse con la rama oficial de Linus Torvalds, cosa que ya hacen un buen número de repositorios con base en kernel.org, dando como resultado repositorios que tienen un tamaño que es una fracción del que serían de otro modo. De igual manera, los diferentes propietarios de las ramas pueden recuperar las otras ramas de manera local en kernel.org sin tener que salir a Internet, lo que ahorra un tiempo considerable en el caso de que Michal lo necesite con regularidad.

Diversas personas respondieron a la consulta de Michal acerca del mantenimiento.

Todas coincidieron en considerar las tareas como livianas, requiriendo tanto tiempo como Michal quiera dedicarle, aunque Pekka Enberg señaló que sería mejor limitar la lista de mantenedores oficiales a un número pequeño. Sugirió además que si más gente iba a mantener el port, sería útil comparar la situación con la de otros ports y posiblemente iniciar una lista de correo separada para la discusión entre los mantenedores.

Stefan Richter indicó que los parches necesitarían al menos una revisión pública bien en una lista de correo especializada o en la lista del linux-kernel. Bryan Wu señaló que, a la larga y para su consideración, los parches deben ir a la lista linux-kernel. Añadió que Michal podría tener que prestar atención a las respuestas que recibiera acerca de sus parches, y estar preparado para desarrollar revisiones rápidamente. Luego, generalmente, sus parches irían a la rama `-mm` de Andrew Morton, donde se pulirían hasta terminar pasando a Linus para su inclusión final en la rama oficial.

Bryan y otra gente estuvieron discutiendo acerca de la “ventana de fusión”, el esfuerzo que está encabezando Linus actualmente para generar lanzamientos en plazo. Bryan sugirió enviar parches a la lista de correo una vez de que se abriese la ventana de fusión, generalmente dos semanas después del anterior lanzamiento. Pekka no estuvo de acuerdo y comentó que esperar a la ventana de fusión sólo retrasa-

ría la inclusión del código, y Stefan explicó que la ventana de fusión sólo se aplicaba a código que se había revisado y probado minuciosamente. Para alcanzar ese estado, Michal tendría que enviar su código tan pronto como fuese posible.

Arnd Bergman aportó algunos consejos para Michal, especialmente relativos a la fusión inicial en la rama de Linus. Arnd recomendó separar todos los drivers en sus propios conjuntos de cambios git, y luego dividir cada uno de ellos en piezas fácilmente revisables que se puedan abordar individualmente. ■

## Configurar Numlock desde la BIOS

Bodo Eggert ha escrito un parche que lee el estado por defecto del bloqueo del teclado numérico desde la zona de datos de la BIOS de un IBM PC. Esta vieja tecnología se remonta a 1981. Conseguir que el Numlock estuviese activado por defecto en Linux ha sido tradicionalmente difícil, requiriendo que los usuarios desarrollasen sus propias soluciones. El código de Bodo promete facilitar este reto. ■

## Generalizar Dependencias de Scripts

Andreas Mohr estaba descontento tras encontrar shell scripts en la rama de fuentes del kernel que confiaban en bash pero que especificaban `#!/bin/sh` en su código. En su sistema, por ejemplo, esto provocaba que la shell por defecto se colgase porque usaba el shell dash en lugar de bash. Andreas decidió eliminar todas las características bash por lo menos del script de parcheo del kernel, para lo que envió un parche. A todo el mundo le gusta una buena discusión sobre bash, por lo que hubo gran cantidad de comentarios sobre su trabajo, enviando varios parches revisados para su consideración. En un momento determinado, Adrian Bunk señaló educadamente que una solución más rápida podría ser arreglar la línea superior del código, sustituyéndola por `#!/bin/bash` en su lugar. Sin embargo, no cundió el desánimo para abordar este reto en la práctica. ■

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.



Horas y horas programando un código sin errores.  
Un desarrollo web impecable.  
Miles de visitas avalan tu buen trabajo.  
Y va tu servidor y se cae... ¡No responde!

# ¿Tu servidor web no responde?

La próxima vez, confía en **arsys.es**

En **arsys.es** sabemos lo que te juegas en cada proyecto, por ello te ofrecemos nuestros **Servidores Dedicados IBM** con soluciones específicas a la medida de tus necesidades. El mejor hardware. La seguridad garantizada por la certificación ISO 27001. La garantía del líder español. 12 años de experiencia. 260 profesionales a tu servicio 24 horas al día, 365 días al año. Uno más si es bisiesto. No te la juegues con tu servidor web.



Acceso  
a Internet

**Servidores  
Dedicados**

Dedicado Genérico  
Dedicado Administrado  
Dedicado de Correo

Hosting

Dominios

Housing

Aplicaciones

[www.arsys.es](http://www.arsys.es) / 902 11 55 30

### Ext4 por la Vía Rápida

A pesar de ser un código experimental y poco fiable, el proyecto ext4 ha recibido un trato preferente y vive en la rama estable del kernel desde hace tiempo, en parte debido a que los desarrolladores de ext4 son casi los mismos que los de ext3, que son muy conocidos y reputados. Incluso así, el código aún no está listo para su uso generalizado, a pesar de que resultaba atractivo para los nuevos usuarios simplemente por el hecho de estar en la rama principal, algo que se pensó para permitir que cualquier persona pudiese probarlo.

Adrian Bunk percibió que la situación era demasiado peligrosa, pues observó que los usuarios adoptaban ese código simplemente porque estaba allí. Bunk envió un parche para que el sistema de archivos dependiera de *BROKEN*, lo que al menos proporcionaría un aviso adecuado.

A algunas personas esto les pareció demasiado extremo, y opinaban que con marcar el código con *EXPERIMENTAL* sería suficiente. Esto permitiría a los usuarios seleccionar el código desde la interfaz más

usada en lugar de eliminar manualmente la dependencia sobre *BROKEN*, lo que traía a colación otro problema que Adrian había estado intentando resolver: la confianza excesiva que los usuarios tienen en funcionalidades marcadas como *EXPERIMENTAL*. Debido a los muchos drivers necesarios que confían en esto, los usuarios suelen habilitar las funcionalidades experimentales por defecto, permitiendo que cosas como ext4 estén presentes en sus configuraciones como si estuviesen realmente listas para ser usadas. Adrian ha estado intentando erradicar la configuración *EXPERIMENTAL* del kernel durante meses.

Otra gente protestó porque entendían que colocar ext4 en el kernel principal tenía el sentido de dejar que la gente lo probase, pero Adrian recordó a estos chicos que ext4 es un caso especial, siendo muy inusual tener una funcionalidad tan inestable en el kernel principal (al menos en esta fase del desarrollo).

Había bastante oposición al cambio de Adrian, incluyendo un comentario de

Alan Cox en el sentido de que deshacerse de *EXPERIMENTAL* era algo equivocado y un intento de reescribir la historia. La discusión terminó sin llegar a una conclusión, pero quedó claro que ext4 tiene poderosos valedores que quieren potenciar su uso en fase experimental tanto como sea posible, y que no quieren saber del aro por el que pasan normalmente los sistemas de archivos.

Esta situación recordaba a la de ReiserFS, que fue criticado y excluido del kernel por no solucionar los problemas que varios de sus desarrolladores tenían con él. De hecho, los conflictos personales con Hans Reiser son probablemente la razón de la exclusión de ReiserFS de la rama del kernel, aunque ya se suponía que otros sistemas de archivos tendrían que pasar por el mismo duro examen. Debido a que ext4 está siendo desarrollado por "insiders" del kernel, ha tomado un estatus especial. Queda por ver cuán especial seguirá siendo y qué precedentes supone para proyectos futuros. ■

