

NOTICIAS DEL KERNEL

Estado de Reiser4

El proyecto Reiser4 se encuentra todavía bajo desarrollo activo, a pesar del juicio por asesinato en el que está inmerso Hans Reiser. Eric Hopper pidió recientemente una actualización del estado del proyecto: específicamente, si había posibilidades de que el código se fusionase a la línea principal del kernel en un futuro cercano. Eric recibió diversas respuestas. Rik van Riel comentó que no le parecía que Reiser4 estuviese en la senda adecuada. Señaló que Namesys había tomado decisiones de desarrollo basándose en necesidades comerciales en lugar de lo que hubiera sido mejor para el kernel, como la decisión de la compañía de incluir soporte para plugins en el sistema de archivos en Reiser4 en lugar de en Linux VFS. Para Rik, decisiones como ésta, y en general la desorganización del proyecto, antes incluso de la acusación de asesinato sobre Reiser, complican que Reiser4 pueda entrar en el kernel a corto plazo.

Pero Andrew Morton tenía una sensación diferente. Los ingenieros de Namesys simplemente no habían pedido recientemente que se incluyera el código, lo que es típicamente suficiente para bloquear el proyecto para su inclusión. Además, Andrew señaló que los desarrolladores del kernel no han estado revisando el código de Reiser4. Sin esa revisión técnica para identificar los problemas que aún existen, sería casi imposible saber lo que los chicos de Namesys tienen que hacer con su código para llevarlo a un estado aceptable.

Los ingenieros de Namesys también tienen su propia postura respecto a la situación. Ante el incierto futuro de esta compañía, un asunto que los ingenieros han estado considerando es qué tipo de compromiso deben adoptar si el proyecto pasa a ser estrictamente desarrollado de manera voluntaria. Edward Shishkin de Namesys comentó

sus planes de dedicar cerca de un cuarto de su tiempo diario de trabajo al proyecto, sea pagado o no. Podría haber otro ingeniero también interesado en algún tipo de compromiso similar.

Edward tampoco estaba de acuerdo con la afirmación de Rik acerca del soporte de plugins en Reiser4. Según Edward, los plugins de Reiser4 están directamente relacionados con el diseño de los sistemas de archivo de disco y no tienen sentido técnicamente hablando en el VFS. Por implicaciones, parece que Edward está diciendo que Rik se equivoca al decir que Namesys toma decisiones de desarrollo siguiendo criterios comerciales. Edward también explicó que los ingenieros de Namesys no han solicitado la inclusión del código últimamente debido a que aún tienen un montón de tareas en su lista de cosas por hacer. En este punto de la discusión, diversas personas expresaron interés en probar el sistema de archivos y ayudar de distintos modos, y hubo otra gente que comenzó a discutir sobre los puntos buenos y malos de Reiser4, pero no había demasiados indicios de que ningún desarrollador del kernel tomase súbitamente un renovado interés en proporcionar feedback a los chicos de Reiser4. Las ásperas discusiones con Reiser de antaño generaron el entusiasmo de un buen número de personas que se ofrecieron activamente a proporcionar ese feedback, y parece que simplemente esta gente no está por la labor de volver a hacerlo. Pero al menos parece que hay lugar para su consideración, y Andrew parece receptivo a aceptar parches. ■

Eliminación del Código X86_SPEEDSTEP_CENTRINO_ACPI

Adrian Bunk ha postado un parche para eliminar el código X86_SPEEDSTEP_CENTRINO_ACPI. Este código ha

quedado obsoleto desde hace tiempo en favor del similar, pero mejor, módulo X86_ACPI_CPUFREQ. El parche se posteo antes de que saliera 2.6.21, y a Dave Jones le pareció que era demasiado tarde para eliminar el código a tiempo para ese lanzamiento. Dave sugirió que ni siquiera se considerara el parche hasta que 2.6.21 se hubiese lanzado. Bill Davidsen también era reacio al parche por diferentes razones. Su argumento era que esta funcionalidad se usaba en ordenadores relativamente recientes, por lo que era importante que todo el mundo supiese cuál era la alternativa antes de eliminar el código. Pero Adrian le recordó a Bill que el código ya se había marcado como obsoleto durante un tiempo y tenía un texto explicativo en el archivo *feature-removal-schedule.txt*. Independientemente de estas pequeñas objeciones, no parece que este código desaparezca en el próximo par de lanzamientos del kernel. ■

Estado de Software Suspend

Nigel Cunningham hizo recientemente una interesante consideración relativa a incluir su código suspend2 en el kernel. Señaló los diferentes puntos en los que su código es superior al del proyecto swsusp de Pavel Machek, en términos de organización del código, funcionalidades soportadas, facilidad de uso y un sólido soporte desde las listas de correo y por el propio Nigel.

Nigel también consideró algunas desventajas de fusionar suspend2, como el gran tamaño del parche, el hecho de que swsusp ya existía, y que swsusp podría implementar algunas de las funcionalidades de suspend2 en un futuro.

Argumentó que ninguna de estas objeciones era demasiado seria. Como señaló Nigel, el punto a favor de suspend2 era que swsusp simplemente no solucionaba suficientes problemas de

la gente y no los iba a solucionar pronto, mientras que suspend2 ya existía y funcionaba bien. La discusión inicial era escasa. Como viene sucediendo durante años, Pavel mantiene la porción del software de suspensión del kernel, y prefiere su propia implementación. Y como señala John Anthony Kazos Jr., Linus Torvalds no quiere tener múltiples versiones de software de suspensión compitiendo en el kernel.

Linus abordó la discusión diciendo que no estaba contento con el método que estaban adoptando los desarrolladores, y que esperaba que llegase alguien nuevo que abordase el proyecto y le diese una dirección completamente diferente. Creía que los actuales desarrolladores estaban demasiado atrincherados en sus posiciones para considerar las alternativas, pero afirmó que si se encontrase un buen software de suspensión, él lo apoyaría incluyéndolo en el kernel. No le parecía que este tipo de cosas estuviera destinado a ser un problema de espacio de usuario. Linus hizo también algo típico en él al abordar cierta clase de problemas. Parecía que ya había pensado en algún tipo de dirección completamente diferente que en última instancia solucionará los problemas del software de suspensión (desde su punto de vista), y ahora realiza declaraciones que parecen no tener sentido para el resto de la gente. Si Linus se mantiene fiel, un buen número de gente discutirá con él, entonces súbitamente la idea básica quedará clara, alguien vendrá corriendo, la implementará y aparecerá un software de suspensión completamente nuevo que

hará las cosas de manera diferente y mejor. ■

Nuevo Driver FBDev

Alan Hourihane posteó recientemente algo de código para proporcionar un driver FBDev para el chipset Intel LE80578. Intel lleva el trabajo con los drivers a través de Tungsten Graphics, que es donde trabaja Alan. Un buen número de personas ha hecho comentarios técnicos acerca del código, todos ellos bastante menores, y Alan replicó rápidamente con parches actualizados. Parece que no habrá obstáculos significativos para que se incluya este código, y nos preguntamos si estará presente en el próximo lanzamiento del kernel.

La lista de correo del kernel de Linux comprende lo principal de las actividades de desarrollo de Linux. El volumen de 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 quincuagesima presencia de las actividades del kernel de Linux del mayor especialista en el tema. ■

No sólo el cielo es azul...

