

NOTICIAS DESDE EL KERNEL

Estabilidad y Desarrollo

Adrian Bunk ha propuesto una sencilla reforma al método actual del desarrollo del kernel. En estos momentos, el árbol oficial del kernel dispone de unas semanas de aceptación de código nuevo, seguido de unas semanas de esfuerzos de modificación y estabilización. Se prosigue la estabilización en el árbol w.x.y.z después del lanzamiento oficial, que implementa estrictas restricciones a la hora de la aceptación de parches.

Según las propuestas de Adrian, esto seguiría siendo igual, pero después de cada lanzamiento oficial, el árbol w.x.y.z anterior relajaría las restricciones de parches para que el proceso se pareciera más al de la época en que las versiones pares buscaban la estabilidad y las impares buscaban el desarrollo.

Una vez se liberase la versión 2.6.17, por ejemplo, Adrian sugiere que el árbol 2.6.16.z continúe su desarrollo de la misma manera en que se desarrollan los árboles 2.4 y 2.2, buscando la estabilidad global a la vez que se aceptan cautelosamente retroportes y actualizaciones adicionales.

La idea actual de Linus Torvalds y André Morton, así como la de otros mantenedores primarios, es que las distribuciones Linux son las que tienen que responsabilizarse de la estabilización del kernel para sus propios lanzamientos. Se reconocen oficialmente las distribuciones como parte integral del

proceso de desarrollo que cuida de la estabilidad. Esta decisión no se basa en una consulta a los fabricantes de distribuciones. Simplemente reconoce que las distribuciones liberan kernels fuertemente parcheados, que la mayoría de la gente utiliza kernels de distribuciones y que las pruebas de estabilidad se incluyen en las modificaciones de cada distribución.

El árbol w.x.y.z se creó para tratar los problemas de estabilidad más recalcitrantes del árbol principal. Se impusieron estrictas limitaciones a este árbol, precisamente porque Linus no deseaba que se convirtiera en otra rama de desarrollo. A la luz de lo cual, la propuesta de Adrian parece sugerir que la actual indiferencia hacia la inestabilidad del modelo de desarrollo no ha funcionado, y que se debe volver a un modelo más similar al que había antes. No es sorprendente que éste haya sido un tema polémico pero que sin duda, volverá a surgir.

Una de las mayores objeciones a la idea de Adrian es que sería una labor más tediosa y aburrida de lo que cualquiera pudiera soportar. Linus lo ha dicho de las propuestas iniciales para el árbol w.x.y.z y la gente se lo está diciendo ahora a Adrian. El problema con ese argumento es que hemos tenido muchas series estables que han funcionado. Y la objeción de Linus de que la estabilización es "aburrida" es más un reflejo de sus gustos personales que el interés del desarrollo del kernel. En todo

caso, sí han habido numerosos intentos de crear un árbol estable al viejo estilo.

Sin embargo, incluso los mantenedores del árbol w.x.y.z, un proyecto mucho más restringido, afirman que los problemas de un esfuerzo de estabilización más activa son casi insuperables. Por otra parte, la sugerencia de Adrian indica que hay algunos desarrolladores del árbol w.x.y.z que encuentran que el depositar las tareas de estabilización en las distribuciones es insuficiente.

Licencia Incorrecta

Se ha admitido código de Silicon Graphics que inadvertidamente contenía notificaciones muy restrictivas en materia de copyright que, a efectos prácticos, ha puesto al kernel en posición de violación de copyright durante varios meses. Resulta que el ingeniero de SGI que donó el código simplemente se olvidó de cambiar los términos de la licencia a los de la GPL en el fichero de cabecera. Este descuido fue un error sin mala intención que se corrigió fácilmente, pero el hecho es que este tipo de error ocurre como media un par de veces al año.

Linus Torvalds ha implementado una especie de sistema de seguimiento, donde cada revisor "firma" la calidad y legalidad de cada parche. Este sistema de revisión se creó específicamente para evitar futuros inconvenientes de ataques tipo SCO y para que Linus pudiera encontrar fácilmente los verdaderos orígenes de cualquier tipo de código en el kernel y evitar así violaciones de la legalidad.

A pesar de que el sistema de seguimiento no es perfecto, tal vez no tiene por qué serlo. El ataque de SCO tenía como intención dañar a Linux, y tuvo éxito en el sentido en que los desarrolladores del kernel tuvieron que adoptar un poquito más de burocracia para su protección.

Un sistema perfecto entrañaría incluso más burocracia y haría que fuese incluso más difícil que desarrolladores, (sobre todo nuevos desarrolladores), contribuyeran al desarrollo del kernel.

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.



Miniconfig

La nueva característica miniconfig, que apareció en el kernel 2.6.15, está llamando mucho la atención a los desarrolladores. Miniconfig permite a los usuarios especificar sólo las características del kernel que desean. El proceso de compilación se encarga *a posteriori* de resolver todas las dependencias.

Debido al hecho de que los ficheros miniconfig son mucho más sencillos que los ficheros config tradicionales, tienden a ser portables de una versión del kernel a otra, se pueden editar a mano sin romper nada, y al leerlos, se ven de un vistazo todas las características principales que soportará el kernel.

Rob Landley está involucrado en mucho del trabajo que se esté llevando a cabo para extender miniconfig. Ha implementado un nuevo objetivo de compilación *make miniconfig* con el cual, en vez de utilizar un fichero miniconfig para generar un fichero config antes de la compilación, se genera un nuevo kernel directamente del propio miniconfig, lo que es mucho más conveniente que el mecanismo incluido por defecto en la versión 2.6.15.

Rob también ha escrito un script que genera ficheros miniconfig a partir de un fichero config completo. Lo que hace en esencia es eliminar todas las dependencias de un fichero config hasta que sola quedan las características primarias. En estos momentos, el script es muy lento, al utilizar el método iterativo de reconstruir el fichero config completo después de cada eliminación para ver cómo afecta al conjunto. A pesar de su lentitud, funciona, y existe la posibilidad de encontrar métodos más veloces para la generación de ficheros miniconfig.

Suspensión de Software en Espacio de Usuario

Pavel Machek recientemente ha implementado una versión prototipo de suspensión de software que coloca mucha de su funcionalidad fuera del kernel, y sólo una mínima parte tiene una interfaz con el núcleo. La parte de la interfaz de hecho se sigue discutiendo, al ser muy dificultoso elegir entre las numerosas, pero imperfectas, opciones disponibles.

Dave Jones ha dicho que si los planes de Pavel tienen éxito, Red Hat podría modificar su propia versión de

suspensión de software, ya que su kernel dispone de un rootkit para la prevención de parches que podría interferir con la escritura a regiones de memoria arbitrarias. También hay que destacar que Dave considera la suspensión de software en espacio de usuario una mala idea, se mire como se mire.

Sin embargo, la gente de Red Hat se lo puede tomar con calma, ya que la aceptación de los parches de Pavel no necesitarán de la eliminación de cualquier característica existente. Ambos métodos de suspensión podrían coexistir pacíficamente durante una buena temporada antes de la toma de cualquier decisión.

Nigel Cunningham también está trabajando en una versión de suspensión que plantea a su vez nuevos problemas. Por un lado, se incluye mucho código en el kernel. Por otro, de tener razón Pavel, mucho de lo que hace el código de Nigel podría realizarse en espacio de usuario.

La gente involucrada en el desarrollo de la suspensión de software no parece llevarse muy bien y muchas de las discusiones tienden a acabar en enfrentamientos directos.

Una lástima. ■

¿Quieres conocer lo más nuevo?
Apuntate ya y recibe mensualmente
y en exclusiva las novedades
de **LINUX MAGAZINE**



info@linux-magazine.es
subs@linux-magazine.es
anuncios@linux-magazine.es
atrasados@linux-magazine.es
preguntas@linux-magazine.es
correo@linux-magazine.es
eventos@linux-magazine.es
dvd@linux-magazine.es
director@linux-magazine.es
boletin-subs@linux-magazine.es
encuesta@linux-magazine.es
boletin@linux-magazine.es

www.linux-magazine.es