



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

¿Porque hay dos sistemas de memoria virtual compitiendo en el kernel?

(18617 lectures)

Per **Ricardo Galli Granada**, [gallir](http://mnm.uib.es/gallir/) (<http://mnm.uib.es/gallir/>)

Creado el 21/10/2001 13:21 modificado el 21/10/2001 13:21

Alguien me puede explicar que es lo que pasa entre Alan Cox y Linus respecto a la memoria Virtual. ¿Por que en el directorio del kernel aparecen los parches de A.C. por separado? Esta es una pregunta que hicieron en un [artículo anterior](#)⁽¹⁾. Como la respuesta me estaba quedando muy larga, decidí moverla a este artículo donde explico brevemente, y según mi punto de vista, las razones de la co-existencia y competencia de ambos sistemas.

Andrea Arcangeli enviaba parches desde hace tiempo para cambiar el VM de Rik van Riel. Según Andrea el sistema de memoria virtual debe ser simple para que sea más fácil hacerlo funcionar en la mayoría de los casos.

Esos parches no había sido aceptados por Linus, hasta que mirando los reportes de problemas del VM, sobre todo en el exceso de *swapping* por el crecimiento desmesurado del page-cache (ver [este artículo](#)⁽²⁾) cuando se usa el `mmap()` en vez del `read()` o `write()` debido a una mala selección *víctimas*. Durante la revisión notó que el VM tenía varios bugs y que no los había detectado antes porque no se había puesto a seguir en detalle cada camino posible desde donde eran llamadas las funciones del VM.

El uso del `mmap` algo bastante común en la mayoría de los programas que haces uso extensivo de operaciones sobre ficheros. Por ejemplo los reproductores MP3 u Ogg. Cuando se usa el `mmap`, la gestión de cache reside en el sistema de page cache, y justamente en este sistema no se detectaban aquellos bloques que eran *read-once*, es decir que son leídos una sola vez en mucho tiempo, como en el caso de la reproducción MP3. En este caso el sistema no lo detectaba y dejaba dichos bloques en memoria RAM (en el page-cache) mientras enviaba a disco páginas de programas en ejecución. Justamente el efecto contrario al deseado.

El caso es que al mismo tiempo que se decidió a unificar el sistema buffer-cache y page-cache (ya pondré en Bulma un artículo que escribí para unas revistas, en cuanto salga publicado), también decidió que era mejor tener un VM simple como el de Andrea, en vez del que era el estándar en aquel momento (2.4.9), el de Rik van Riel.

El caso es que Alan Cox piensa lo contrario, aunque todavía no ha tomado una decisión sobre el futuro, por ahora él sigue manteniendo al VM de Rik en su árbol, los `xxxx-ac`.

Según Alan, los sistemas de memoria virtual, que entre otras cosas gestionan el page-cache, son sistemas complejos y que necesitan mucho tiempo para ir ajustando a los diferentes casos y cargas, por ello prefiere el de Rik que ya está muy evolucionado y probado en muchos casos, aunque hay áreas y casos específicos, donde tendría que mejorar.

Como resultado tenemos que el kernel estándar desde la 2.4.10 usa el VM de Andrea, el 2.4.+10-ac por el contrario usa el tradicional, aunque mejorado día a día, de Rik van Riel. Ambos están compitiendo por ser el gestor estándar de Linux. Aunque parecería que el Andrea lleva ventaja por estar en el *oficial*, no hay que olvidar que Alan será el responsable del mantenimiento del 2.4 en cuanto comiencen a trabajar en el 2.5.

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=930>
 2. <http://bulma.net/body.phtml?nIdNoticia=908>
-

BULMA: ¿Porque hay dos sistemas de memoria virtual compitiendo en el kernel?



E-mail del autor: gallir_ARROBA_uib.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=932>