

NOTICIAS DEL KERNEL

¿Un Compilador C dentro del Kernel?

Ingo Molnar sugirió que el kernel debería incluir su propio compilador de fuentes dentro de la base de código, una alternativa a GCC fuertemente integrada en Linux que eliminaría la pesadilla de los conflictos que en ocasiones hacen estragos entre los desarrolladores del kernel y los desarrolladores de GCC. Si alguien menos implicado en el desarrollo del kernel de Ingo sugiriera algo como esto, sospecho que encontraría un alto grado de escepticismo. De hecho, incluso siendo Ingo el que lo sugiere, hubo algo.

Su idea era combinar un precompilador, compilador, ensamblador y enlazador en una sola herramienta que se incluiría con las fuentes del kernel y permanecería “hombro con hombro” con las demás, evitando no sólo los

conflictos en el desarrollo, sino también la gran cantidad de piruetas que tiene que hacer el kernel en la actualidad para adaptarse a GCC. Como primer paso, la herramienta podría realizar simplemente un pre-procesado que alimente a GCC. Luego, podrían añadirse gradualmente más funcionalidades del compilador. Esto, señaló Ingo, tendría un beneficio inmediato en términos de simplificar el código del kernel.

Inicialmente, hubo cierto apoyo a la idea. Steven Rostedt planteó la cuestión, sugiriendo que la propuesta de Ingo era buena, y luego Ingo respondió con la solución comentada. Y Anton Ertl señaló que una alternativa a GCC no sólo ayudaría a conseguir que el kernel se librase de los problemas del GCC, sino que ayudaría también a gran cantidad de software en espacio de usuario. Pero pensaba que hacer un fork a GCC como punto de partida tenía algunos buenos argumentos a favor, incluida la posibilidad de compilar código para una gran variedad de arquitecturas de hardware. David S. Miller señaló que una desventaja de escribir un nuevo compilador sería perder la eficiencia de GCC al hacer el preprocesado y la compilación en el mismo binario, en lugar de pasar datos a través de una tubería. No le gustó la idea de Ingo en general. En lugar de trabajar en un compilador, pensaba que la gente debería centrarse en escribir código del kernel. Como también, que todo el proyecto llevaría mucho más tiempo y esfuerzo que el esperado por Ingo. Después de haber escrito un compilador anteriormente, Eric W. Biederman, confirmó igualmente que se necesitaría mucho tiempo, pero pensaba que la idea de Ingo podría merecer la pena si daba lugar a grandes avances en la depuración y la velocidad de compilación.

Christoph Lameter también pensó que algo más pequeño que un compilador completo podría ser de utilidad. Algo que pudiera deshacerse de todas

las complejidades y casos especiales relacionados con el preprocesamiento actual sería una mejora significativa, señaló. Alexander Viro no estaba de acuerdo con que la situación actual del preprocesador fuese muy compleja, e invitó a la gente a participar en el hilo para leer el estándar C [2], específicamente el Capítulo 5 y el artículo 6.10. En este punto, la conversación se agotó.

Lo que puede salir de esta discusión no está claro, pero parece haber llamado la atención de algunos programadores poderosos. Puede que se concrete lo que la gente cree que son las deficiencias en GCC en la actualidad, pero es probable que alguien intente investigar más profundamente, al menos en cuanto a pre-procesamiento. ■

Herramientas de Sistema de Archivos Liberadas por SGI

Greg Banks anunció que SGI iba a publicar varios elementos de software bajo la licencia GNU, versión 2. SGI no participaría en el soporte a los programas. De hecho, una parte del código estaba incompleto, y otra dependía de infraestructura interna de desarrollo de SGI que no iba a liberarse. Así que este caso no parece ser un intento de colaborar con el mundo del software libre, sino más bien simplemente lanzar código al exterior y esperar a que a alguien le resulte útil.

Entre las herramientas liberadas está check-stream, la cual puede detectar automáticamente ciertos tipos de corrupción del sistema de archivos. Otra herramienta, weber, está diseñada para poner una carga pesada en un sistema de archivos NFS. En total, fueron liberadas nueve herramientas. ■

La lista de correo del kernel de Linux comprende lo principal de las actividades de desarrollo de Linux. El volumen del tráfico es inmenso, alcanzán-



dose 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.

RECURSOS

[1] MTP: http://en.wikipedia.org/wiki/Media_Transfer_Protocol

[2] C estándar: <http://www.open-std.org/jtc1/sc22/WG14/www/docs/n1256.pdf>