

NOTICIAS DEL KERNEL

La Nueva Herramienta PowerTOP para Baterías

Arjan van de Ven, de Intel, anunció la nueva herramienta PowerTOP. PowerTOP analiza todo lo que se está ejecutando en el sistema y reporta cuánta potencia se está consumiendo. Permite identificar un gran número de maneras de ahorrar batería en un portátil. Las pruebas de Intel muestran una mejora en la duración de una batería “típica” de más de una hora. Independientemente de estas pruebas, parece claro que esta herramienta puede ser verdaderamente útil, tanto para los usuarios normales como para los mantenedores de aplicaciones que puedan estar consumiendo demasiada batería sin saberlo. El software libre en general se dispone a recibir una avalancha de parches para reducir el consumo. Se supone que, ejecute o no PowerTOP, su portátil comenzará a tener una duración de batería mayor gracias a toda esta nueva información. <http://www.linuxpowertop.org>

Nuevo Driver para el Chipset 965GM Express

Keith Packard, de Intel, ha anunciado un nuevo driver fabricado por la compañía para el chipset 965GM Express. Tal y como señaló, este chipset “representa el primer producto para portátiles que implementa la cuarta generación de arquitectura gráfica de Intel. Se ha dise-

```
PowerTOP version 1.4 (C) 2007 Intel Corporation
Cn      Avg residency (10s) Long term residency avg
C0 (cpu running) ( 0.8%)
C1      0.0ms ( 0.0%) 0.0ms
C2      20.2ms (99.2%) 20.4ms
C3      0.0ms ( 0.0%) 0.0ms
C4      0.0ms ( 0.0%) 0.0ms

Wakeups-from-idle per second : 49.1
no ACPI power usage estimate available

Top causes for wakeups:
29.1% ( 21.3) S06cpuspeed : queue_delayed_work_on (delayed_work_timer_fn)
15.6% ( 11.4) <interrupt> : ipw2200, Intel 82801DB-ICH4
 9.4% ( 6.9) xchat : schedule_timeout (process_timeout)
 8.9% ( 6.5) artsd : schedule_timeout (process_timeout)
 5.9% ( 4.3) gnome-terminal : schedule_timeout (process_timeout)
 5.5% ( 4.0) <kernel module> : usb_hcd_poll_rh_status (rh_timer_func)
 4.1% ( 3.0) gnome-power-man : schedule_timeout (process_timeout)
 3.3% ( 2.4) emerald : schedule_timeout (process_timeout)
 3.0% ( 2.2) Xorg : do_setitimer (it_real_fn)
 2.1% ( 1.5) <kernel core> : queue_delayed_work_on (delayed_work_timer_fn)
 1.9% ( 1.4) gam_server : schedule_timeout (process_timeout)
 1.8% ( 1.3) <interrupt> : libata
 1.4% ( 1.0) Xorg : schedule_timeout (process_timeout)
 1.4% ( 1.0) gnome-screensav : schedule_timeout (process_timeout)
 1.4% ( 1.0) artsd : do_setitimer (it_real_fn)
 0.7% ( 0.5) <kernel module> : neigh_table_init_no_netlink (neigh_periodic)
 0.7% ( 0.5) events/0 : ipw_gather_stats (delayed_work_timer_fn)
 0.4% ( 0.3) cupsd : schedule_timeout (process_timeout)
 0.4% ( 0.3) nscd : futex_wait (hrtimer_wakeup)
 0.3% ( 0.2) pam_timestamp_c : schedule_timeout (process_timeout)

Suggestion: Disable the unused bluetooth interface with the following command:
hciconfig hci0 down ; rmmod hci_usb
Bluetooth is a radio and consumes quite some power, and keeps USB busy as well.
Q - Quit R - Refresh B - Turn Bluetooth off
```

Figura 1: Ahorra energía en un portátil con PowerTOP.

ñado para que soporte funcionalidades avanzadas de rendering de las API's gráficas modernas, e incluye soporte para vertex programable, geometría y sombreadores de fragmentos". Mucha gente va a

alegrarse al leer esto. Jeff Garzik comentó: "Ahora esperamos que Intel fabrique una tarjeta de vídeo por su cuenta, para robar más cuota de mercado a sus competidoras de código cerrado".

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.



Cuándo Eliminar Código Antiguo

Robert P. J. Day tomó una iniciativa para actualizar el archivo *Documentation/feature-removalschedule.txt*, de manera que las funcionalidades programadas para su eliminación correspondan efectivamente con aquellas que deberían estarlo. Todo el mundo está de acuerdo en que el archivo está terriblemente obsoleto, pero no está claro cual es la mejor manera de actualizarlo. Una posibilidad es, como sugiere Stefan Richter, simplemente pro-

bar a eliminar el código y ver si alguien se queja. El problema de este método es que como cuesta trabajo eliminar código, el único resultado es que se vuelve a programar una nueva fecha para la eliminación. Parece que es mejor solución hacer todo el trabajo en primer lugar. Con lo que volvemos a la idea de actualizar el archivo *feature-removal-schedule.txt*.

No está claro si el esfuerzo emprendido por Robert llegará a buen puerto. Generalmente, aquellos que podrían verse afectados por una eliminación particular de código no son conscientes del debate hasta que alguien finalmente quita algo, punto en el que protesta airadamente. Puede que el intento de Robert para alcanzar un consenso previo simplemente no sea viable. ■

El Driver Raw se Mantiene

El driver raw se ha sacado de la lista de funcionalidades del kernel obsoletas marcadas para eliminarse próximamente. Dave Jones comenta que ha descubierto, tras repetidos intentos de eliminarlo, que aún tiene una significativa base de usuarios que permanecerá todavía algún tiempo. Ha posteado un parche para sacar al driver de la lista de "a eliminar" y también ha cambiado su status en el sistema de configuración para que no vuelva a considerarse obsoleto.

Este último cambio generó algunas críticas. Hubo gente que señaló que la funcionalidad aún debería considerarse como obsoleta para evitar que la usen nuevos usuarios. El problema es que, como señaló Dave, dejarlo como obsoleto sin más sólo conducirá a que otros desarrolladores del kernel piensen que alguien olvidó pasarla a la lista "a eliminar".

Esta gente volvería a enviar parches para añadirla a la lista nuevamente, y

nos encontraríamos ante un círculo vicioso. ■

Estado de los Drivers MMC

Pierre Ossman ha comentado que va a postear un parche para marcar como huérfanos algunos drivers MMC, a menos que los mantenedores den un paso adelante para responsabilizarse de ellos. Estos drivers incluyen el AMD AU1XX0 MMC, el ARM Primecell MMCI PL180/1, el ATMEL AT91 MCI, el Motorola I.MX MMCI, el PXA MMCI y el TI OMAP MMC Interface. Pavel Pisa se ofreció voluntario para mantener el driver Motorola I.MX MMCI y ha suministrado un parche en el que aparece él mismo como mantenedor oficial.

Khasim Syed Mohammed comentó que Carlos Aguiar y Anderson Briglia podrían estar interesados en mantener el driver TI OMAP MMC Interface para las placas existentes, y él (Khasim) añadiría el soporte para las placas nuevas. Asimismo, Russell King se presentó voluntario para mantener el archivo *mmci.c*. ■

Mantenimiento del Subsistema y DMA Intel I/OAT

Chris Leech ha posteado algunos parches en los que aparece él mismo como mantenedor oficial del subsistema DMA Generic Memcopy y el driver Intel I/OAT DMA. No parece haber ningún problema en ello, salvo que David S. Miller quiere que Chris mande sus parches a la lista de correo de netdev, ya que mucho de su trabajo toca código TCP. ■

¿Escalabilidad con el DMA Genérico del Kernel de Linux?

Rajib Majumder preguntó recientemente si Linux era capaz de escalar hasta 8

CPUs hoy día. Tony señaló acertadamente que existen sistemas con 512 y hasta 1024 CPUs actualmente en producción, y que se había montado un sistema con 4096 como prototipo. Christoph Lameter predijo que veremos sistemas con 16384 CPUs funcionando antes de que termine 2007. Pero Christoph señaló también que cuando el hardware se convierte en un factor limitante con tantas CPUs, sería esencial un buen sistema de archivos (él recomendó XFS). ■

Un Poco de Limpieza de Código Antiguo

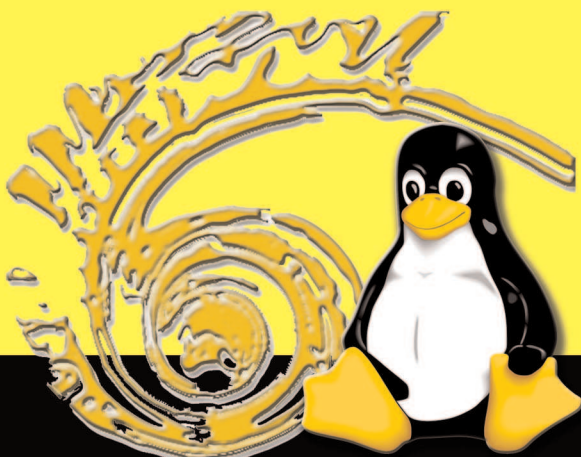
Adrian Bunk y Wim Van Sebreeck han posteado un parche para eliminar el driver watchdog *i8xx_tco*. Su eliminación está prevista en la versión 2.6.22, y su funcionalidad ya se ha reemplazado con el driver *iTCO_wdt*. ■

Purgar UMSDOS del Kernel

El que una vez fue un valiente punto de entrada en el mundo Linux, UMSDOS, ya no está desde la versión 2.6.11. O, mejor dicho, UMSDOS casi se ha ido. Jesper Juhl posteo recientemente parches para eliminar unos pocos archivos de configuración y referencias a archivos que aún persistían. Las únicas cosas de UMSDOS que aún aparecen son las entradas en el archivo *CREDITS* y los números *ioctl* que usaba UMSDOS.

Originalmente, el parche de Jesper eliminó también a *ioctls*, pero sonaron las alarmas cuando H. Peter Anvin señaló que *ioctls* no debía reusarse por otro código, incluso aunque fuese a eliminarse.

La versión posterior de Jesper documentaba todas las *ioctls* de las UMSDOS eliminadas para que no fuesen a usar los mismos números por error. ■



Asociación Linux Español