

Implementación de la Gestión de energía en el Sistema operativo didáctico utilizando el modelo APM

Nicanor Casas, Graciela De Luca, Sergio Martín,
Carlos A. Gergela, Fernando J. Lo Tartaro,
{ncasas,gdeluca,smartin,cgergela,flo|tartaro}@unlam.edu.ar

Universidad Nacional de la Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

Dirección: *Florencio Varela 1703* - Código Postal: 1754 - Teléfono: 4480-8900/8835

CONTEXTO

Este trabajo forma parte del proyecto de investigación de la confección de un sistema operativo de características didácticas, el cual se encuentra en el marco de investigaciones que coordinada el departamento de Ingeniería e Investigaciones Tecnológicas que pertenece a la Facultad de Ingeniería de la Universidad Nacional de la Matanza

RESUMEN

El presente trabajo de investigación se centra en el manejo de los estados de energía por parte del sistema operativo utilizando la norma The Advanced Power Management Standard APM, como introducción a la norma ACPI, que es la que aceptan las notebooks y las netbooks, para manejo de la energía en sistemas de computación y su respectiva implementación en el sistema operativo de características didácticas SODIUM.

Informamos también sobre los problemas encontrados en la utilización de diferentes sistemas, especialmente con herramientas de máquinas virtuales y los desvíos que se produjeron por el uso de las mismas, que retrasaron la construcción de la base de investigación.

Por último hacemos constar los alcances establecidos y se explica hacia dónde apuntaremos nuestra atención en la definición de las líneas de investigación que dejamos abiertas para las futuras investigaciones relacionadas.

Finalmente en las conclusiones expresamos nuestra visión y opinión refiriéndonos en normas generales a la forma de trabajo adoptadas.

KEYWORDS

Driver APM, Advanced Power Management, BIOS, SODIUM, Shutdown, Full On, Suspend, Standby.

INTRODUCCIÓN

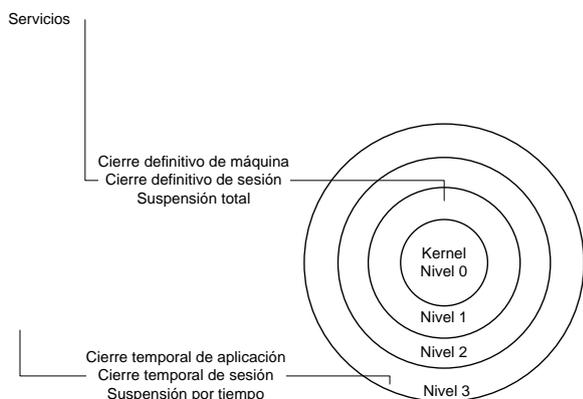
El Advanced Power Management (APM) Standard (revisión 1.2) define diferentes niveles de estado de energía teniendo diferentes niveles de consumo. El APM Standard define los siguientes estados de energía en orden decreciente de consumo: **Full On** (sin manejo de energía); **APM Enabled** (Systema está encendido pero los dispositivos que no están en uso pueden no estar consumiendo energía); **APM Standby** (un nivel de consumo de energía reducida); **APM Suspend** (otro nivel de consumo de energía reducida) y **Off** (el sistema no está trabajando y está apagado).

Hasta el momento de la presente investigación el sistema operativo SODIUM constaba solamente de un bosquejo de implementación de la interfaz APM en la funcionalidad del apagado completo del sistema si ningún tipo de parámetro que permitiera la elección de una o más opciones.

Esta función se encontraba residente en el núcleo del sistema operativo formando una unidad monolítica que implicaba una cierta complicación dado que la misma al agregársele opciones debería permitir ser ejecutada desde el nivel de protección usuario o del nivel de protección administrador.

El objetivo de nuestros estudios es agregar funcionalidades a la opción de apagado de la máquina incluyendo que se pueda indicar el momento del apagado en segundos, minutos, horas y hasta días. También planeamos agregar nuevos estados de los definidos en APM.

El esquema ideado en función a los requerimientos es el que muestra la figura.



Primeramente se trató de implementar la suspensión (**Suspend**) ya que es uno de los estados más comunes implementados en los sistemas operativos y el de mayor ahorro de energía en el cual la energía puede ser cortada desde una gran cantidad de dispositivos, se detiene el **CPU clock**, el CPU core queda en su mínimo estado de energía y el reloj se detiene (lo que causa que los System bus devices se deshabiliten) pero nos hemos encontrado con la dificultad de que el sistema operativo SODIUM no tiene funcionando al 100% el sistema de FILE SYSTEM. Esto es un impedimento a nuestras intenciones dado que al momento de entrar en estado de suspensión el sistema debe guardar el estado de la CPU y la información de la memoria RAM en el disco rígido.

En los primeros meses se pudieron realizar algunos lentos avances en la investigación sobre la implementación de gestión de energía. Las demoras se debieron principalmente a que no se pudo detectar, al principio, que se estaba cometiendo un error en cuanto a las pruebas del sistema. Generalmente se utilizan herramientas para la generación de máquinas virtuales que facilitan la programación y prueba de procesos del sistema operativo. Hasta ese momento las pruebas se estaban realizando sobre el sistema de gestión de máquinas virtuales BOCHS, ya que la misma ofrece ventajas respecto a otros ambientes como por ejemplo que la función de depuración es sencilla e intuitiva.

Debido a que las pruebas realizadas no arrojaban ningún resultado significativo o de relevancia para que se puedan sacar conclusiones., se tomó la decisión de probar con los comandos, considerados a priori por más elementales, como son los utilizados para poner en estado **SUSPEND** y **STANDBY** la máquina.

Cada vez que se ejecutaban los respectivos comandos los mismos no producían ningún tipo de respuesta y no se obtenían mensajes de error que nos permitieran realizar una planificación para guiar nuestras investigaciones hacia otros puntos de convergencia

para solucionar el/los problemas presentados. Esto nos llevó a sospechar de la herramienta que estábamos utilizando. Comenzamos por el análisis de la implementación del BIOS del BOCHS. Entonces nos dedicamos a consultar directamente el código y efectivamente se pudo comprobar que ciertas funcionalidades de **APM** no están debidamente desarrolladas o están sencillamente incompletas. Por ejemplo, faltaba la implementación de la función "**GET CAPABILITIES**" de **APM**, la cual nos arroja como resultado si el BIOS soporta el modo "**SUSPEND**" y el modo "**STANDBY**", algo clave para nuestra investigación. Habiendo descubierto esto, decidimos agregar funcionalidad a un comando, que entre otras cosas nos devuelva esta información para poder informar al usuario que lo que le pide al sistema es posible o no. Este comando es el "**APMINFO**".

A partir de este momento las pruebas siempre se realizaron en máquinas reales, se decidió abandonar la utilización de la herramienta para las pruebas, siendo esta únicamente dedicada para la programación, realizando el arranque del SODIUM directamente desde cualquier soporte físico permitido por el sistema operativo.

Luego de varios días de desarrollo, cambios y pruebas en el sistema, finalmente se pudo lograr que una computadora entre en el modo de suspensión.

Antes de suspenderse esta máquina emitió 3 sonidos conocidos como "BEEP" por el speaker, acto seguido se apago casi totalmente la PC, dejando la fuente en una potencia baja. Al pulsar nuevamente el botón de encendido de la máquina, esta volvió al mismo estado en el que se encontraba antes de la ejecución del comando de suspensión. Se había logrado realizar un "**SUSPEND**" con éxito.

Luego de varias pruebas se continuó con la implementación de otras funcionalidades. El siguiente objetivo: "**STANDBY**".

Al igual que con el comando "**SUSPEND**" creamos otro comando llamado "**STANDBY**". Otra vez, luego de algunas horas de implementación pudimos lograr que el sistema se ponga exitosamente en este estado.

A pesar de obtener, en principio parte de la meta a la que apuntamos, hay varios puntos por revisar y mejorar en el desarrollo. Por ejemplo, hemos incluido una nueva opción para que tanto el comando "**SHUTDOWN**" como el "**STANDBY**" y el "**SUSPEND**" puedan ejecutarse con un temporizador, indicando los segundos, minutos, horas y hasta días como parámetro.

PROBLEMAS A RESOLVER

Un problema mucho más importante por resolver hasta el momento es que los comandos "**SUSPEND**" y "**STANDBY**" funcionan en ciertos tipos de máquinas. Podríamos decir que funciona bien en máquinas de arquitectura anterior y se supuso que deberían

funcionar correctamente en todos los modelos nuevos apelando a la norma de compatibilidad hacia atrás pero esto no es real porque si funcionan en algunas y en otras no. No obstante, en diversas pruebas que realizamos, los resultados obtenidos podemos catalogar como positivos. Al realizar las pruebas del "SUSPEND" notamos que algunas máquinas del realizan el sonido característico antes de realizar la suspensión (como había sucedido en las pruebas exitosas) pero en este caso el resultado fue diferente.

Luego de realizar los sonidos mencionados, el sistema operativo simplemente siguió funcionando, es decir no permanece en el estado en el cual se lo intenta poner. El resultado fue el mismo para el "STANDBY": el sistema pareciera que va a entrar en este estado cuando se ve claramente que la pantalla parpadea pero luego continúa ejecutándose el sistema operativo de forma normal. Nos queda la impresión que el sistema entra momentáneamente en el estado deseado pero se vuelve a reanudar sólo instantáneamente.

Actualmente nos encontramos investigando y realizando más ajustes al sistema para poder dar con el problema y resolverlo. Además, estamos desarrollando la parte de comunicación entre el **driver APM** y el **BIOS**. Con esto se podrá establecer un handshake entre ambas partes y así el **driver APM** tomará conocimiento de los eventos **APM** que el **BIOS** le informe y en consecuencia realizar las acciones necesarias dependiendo del evento que se trate.

LIMITACIONES

Los equipos actuales ya sea PC, notebooks, netbooks se puede decir que son incompatibles con el modelo de administración de energía que intentamos implementar. Como consecuencia de esto la utilidad real de esta implementación se podrá apreciar solo en los equipos que tengan compatibilidad con APM.

CONCLUSIONES

Una vez que nos encontramos teóricamente en conocimiento del tema, lo siguiente es encarar la implementación, lo cual lleva un considerable trabajo de entendimiento del sistema operativo, implementación y muchas pruebas. Para poder comenzar esta parte del trabajo se debe aprender a usar nuevas herramientas, las cuales pueden ser de gran utilidad o también pueden generarnos confusión como nos pasó con el uso del emulador BOCHS que tras varias pruebas nos llevó a darnos cuenta que no estaba implementada justo la parte que nos interesaba a nosotros. También debemos incorporar los otros desarrollos del sistema a los que actualmente hemos llegado, tales como el desarrollo de un file system

FAT12, FAT16 y FAT32 elementales que nos permitirán utilizar el disco rígido, así como también el USB. con sus correspondientes Drivers, del cual actualmente hemos podido arrancar el sistema en máquinas con BIOS genéricos o compatibles.

Lo más positivo de todo es el trabajo es la posibilidad de avance en el conocimiento de las distintas herramientas, aplicaciones identificación de las diferencias en los equipos disponibles mediante comandos y configuración de un sistema que gracias a sus características didácticas permita la enseñanza del diseño de los Sistemas Operativos., ver y discutir diferentes opciones y posibles soluciones a los problemas que se van sucediendo, tales como herramientas a utilizar, compatibilidades, funcionamiento teóricos y reales de APM . En el futuro se tratará de estudiar la aplicación de ACPI basado en la experiencia adquirida en APM.

FORMACIÓN DE RECURSOS HUMANOS

Se realizó:

- La primera transferencia de los conocimientos obtenidos a los alumnos que cursan Sistemas Operativos, ya que realizaron el análisis de la arquitectura, las normas estándar, sus aplicaciones, compatibilidades, utilización de herramientas y las distintos formatos de ejecutables conjuntamente con el análisis del SODIUM e intervinieron en el desarrollo de los administradores.
- Transferencia de conocimientos a los alumnos de Sistemas de Computación II de la Universidad de La Matanza y a los alumnos de Sistemas Operativos de la Universidad Tecnológica Nacional, Regional Buenos Aires.
- Publicación de los avances en la investigación en dos congresos internacionales.
- Se prevé continuar con las publicaciones en otros congresos internacionales

Se está estudiando:

- El realizar convenios de colaboración con otras universidades nacionales estatales y privadas de las cuales recibimos ofrecimientos de colaboración, con el objetivo de intercambiar conocimientos y ampliar los alcances del sistema.

En esta línea de investigación tenemos:

- Dos trabajos de la Maestría en informática en curso.

BIBLIOGRAFÍA

INTEL - APM BIOS INTERFACE SPECIFICATION REVISION 1.2.

<http://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/APMV12.rtf>

INTEL - POWER MANAGEMENT HISTORY AND MOTIVATION.

http://www.intel.com/intelpress/samples/PPM_chapter.pdf

TOSHIBA

http://es.computers.toshiba-europe.com/innovation/download_whitepaper.jsp?service=ES&WHITEPAPER_ID=000000d23

ORIGINAL APM BIOS CHECKING BY STEPHEN ROTHWELL

<http://mediatools.cs.ucl.ac.uk/nets/runes-lippert/browser/6lowpan-linux/arch/x86/boot/apm.c>

MICROSOFT TSR SPECIFICATION

<http://www.delorie.com/djgpp/doc/rbinter/ix/15/53.html>

LINUX CROSS REFERENCE - FREE ELECTRONICS EMBEDDED LINUX EXPERT

http://lxr.free-electronics.com/source/include/linux/apm_bios.h

UBUNTU - MANUALS

<http://manpages.ubuntu.com/manpages/jaunty/man4/apm.4freebsd.html>

BIOS-LESS APM DRIVER FOR ARM LINUX - JAMEY HICKS ADAPTED FROM THE APM BIOS DRIVER FOR LINUX BY STEPHEN ROTHWELL

<http://rswiki.csie.org/lxr/http/source/drivers/char/apm-emulation.c>

http://rswiki.csie.org/lxr/http/source/arch/x86/kernel/apm_32.c#L1284

BOCHS X86 EMULATOR

<http://bochs.sourceforge.net/cgi-bin/lxr/source/bios/apmbios.S>

MANAGING POWER COMPUTER SYSTEM

Mustafa et al. Inventors Mazin A Mustafa Houston , Texas. Hewlett-Packard Development Company, L. P. Houston TX(US) UNITED STATES PATENTS NRO 6678831 JAN 13 2004

ACPI in Linux – Myths vs. Reality Len Brown - Intel Open Source Technology Center

<http://www.kernel.org/doc/ols/2007/ols2007v1-pages-65-74.pdf>