

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,

Universidad Nacional de la Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

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

{ncasas,gdeluca,smartin,cgergela,flo|tartaro}unlam.edu.ar

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 estudio de la norma 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.

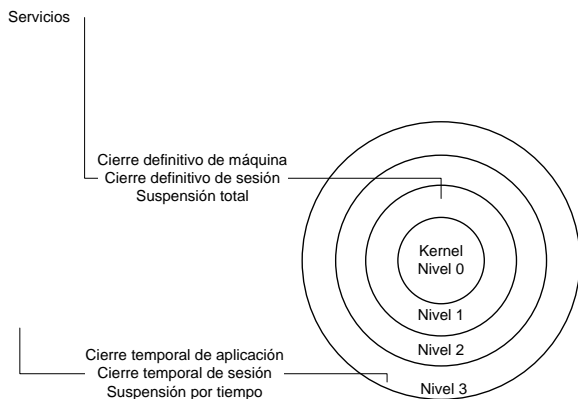
1. INTRODUCCIÓN

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.



En principio habíamos decidido implementar la suspensión ya que es uno de los estados más comunes implementados en los sistemas operativos y el que más energía ahorra 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 nos pudimos 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.

Lamentablemente las pruebas realizadas no arrojaban ningún resultado significativo o de relevancia para que se puedan sacar conclusiones. Fue entonces cuando se decidió que se probaron los comandos, considerados a priori por nosotros más sencillos, como son los utilizados para suspender y poner en stand by la máquina.

Cada vez que ejecutábamos los respectivos comandos no había ningún tipo de respuesta y ni siquiera teníamos mensajes de error que pudieran guiar nuestras investigaciones hacia otros puntos que permitieran solucionar el problema. 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 pudimos 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 para 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 dejar la herramienta no para la programación pero si para las pruebas, arrancando SODIUM directamente desde cualquier soporte permitido por el sistema operativo.

Luego de varios días de desarrollo y cambios en el sistema, finalmente pudimos 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 la que la habíamos dejado. Se había logrado realizar un "SUSPEND" con éxito.

Luego de varias pruebas estábamos listos para seguir implementado 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 supusimos 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.

Lo más positivo de todo es el trabajo en grupo que nos permite avanzar y siempre tener una segunda opinión sobre lo que estemos realizando y discutir diferentes opciones y posibles soluciones a los problemas que nos van sucediendo.

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 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=0000000d23

ORIGINAL APM BIOS CHECKING BY STEPHEN ROTHWELL

<http://mediatools.cs.ucl.ac.uk/nets/runeslippert/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
ELECTRONS EMBEDDED LINUX
EXPERT

http://lxr.free-electrons.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>