

# Utilización de tecnologías adaptativas para la gestión de la energía de un sistema operativo didáctico

Nicanor Casas<sup>1</sup>, Graciela De Luca<sup>1</sup>, Sergio Martín<sup>1</sup>, João José Neto<sup>2</sup>

<sup>1</sup> Universidad Nacional de la Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

Dirección: *Florencio Varela 1703 - Código Postal: 1754*

{ncasas, gdeluca, smartin}@unlam.edu.ar

<sup>2</sup> Universidade de São Paulo

Departamento de Engenharia de Computação e Sistemas Digitais

*Avenida Prof. Luciano Gualberto, trav. 3, n. 158 CEP 05508-900 São Paulo – SP – Brasil*

joao.jose@poli.usp.br

## Resumen

El presente proyecto de investigación se centra en la aplicación de dispositivos adaptativos —en nuestro caso, tablas de decisión— por parte de los usuarios para lograr que la gestión de la energía del sistema operativo se adapte, lo mejor posible, al uso de los recursos subyacentes, tanto procesador como dispositivos y motherboard, con el objeto de maximizar la eficiencia de consumo y el rendimiento general del sistema.

La motivación creciente por la búsqueda de la eficiencia energética como alternativa al paradigma de rendimiento a cualquier costo, impulsa la necesidad de herramientas que procuren reducir significativamente el consumo eléctrico y espacial con mínimo impacto en el rendimiento. Se busca reducir el costo económico y la tasa de fallas, e incrementar la duración de baterías de dispositivos portátiles.

Al finalizar, obtendremos una tabla de decisión inicial que mejor responda, de manera general, a los escenarios de gestión de energía sin información previa sobre las costumbres de uso del usuarios y, además, cuando se alcance una propuesta de acciones adaptativas que, a través del análisis de métricas que deberán ser definidas, logren una tabla diferente para cada usuario, que supere la eficiencia y el rendimiento de la inicial, luego de un lapso de tiempo.

### Palabras clave:

Power Management, Operating Systems, Adaptive Decision Tables, BIOS, SODIUM, ACPI Driver, APM Driver.

## Contexto

Este proyecto forma parte de la línea de investigación iniciada en 2005, que consiste en el diseño y construcción de un sistema operativo de características didácticas llamado SODIUM —Sistema Operativo del Departamento de Ingeniería de la Universidad de la Matanza (UNLaM)—, el cual se encuentra en el marco de investigaciones que coordina el Departamento de Ingeniería e Investigaciones Tecnológicas que pertenece a la Facultad de Ingeniería de la Universidad Nacional de la Matanza.

Se continúa el desarrollo de la investigación precedente [2], que fue presentada en el “XIII WICC”, realizado en la ciudad de Rosario. Como proseguimiento, se implementarán mecanismos de reducción de consumo de energía alternativos para complementar dicho estudio.

Un primer avance de la presente investigación consistió en la presentación de un diseño reconfigurable de aspectos del sistema operativo con tablas de decisión adaptativas [3] en el “VI Workshop de Tecnologías Adaptativas (WTA 2012)”, evento que fue organizado por la Universidade de São Paulo, en Brasil.

Estaremos coordinando el trabajo en conjunto con el grupo de investigación del Laboratorio de Lenguajes y Técnicas Adaptativas (LTA) de dicha universidad, dirigido por el doctor João José Neto.

## Introducción

### Avances de la investigación

Luego de la investigación realizada el año pasado sobre el modelo APM para la gestión de la energía en un sistema operativo didáctico [2], se logró incluir en SODIUM la posibilidad de efectuar tres acciones básicas a tales efectos:

- **Apagado por software (Shutdown):**  
El comando APM shutdown permite enviar una señal al BIOS para que apague el CPU y corte la alimentación de energía al resto del sistema. El API desarrollado para SODIUM para esta funcionalidad finaliza todos los procesos y permite especificar un conteo previo al apagado.
- **En espera (Standby):**  
El comando APM standby permite enviar una señal al BIOS para que ponga al CPU en estado de bajo consumo, deshabilite los puertos de video y reduzca las revoluciones de los discos rígidos. Además, concede el retorno a la actividad normal rápidamente. SODIUM se encarga de que no hayan peticiones de I/O o procesos críticos antes de enviar el comando.
- **Suspender (Suspend):**  
El comando APM suspend permite enviar una señal al BIOS para que ubique al sistema al nivel de más bajo consumo, que permita luego volver al estado anterior. También, en este caso, SODIUM se encarga de que no hayan peticiones de I/O o procesos críticos antes de enviar el comando.

Se hubo implementado, además, el comando “APMINFO” que permite saber con qué versión de APM cuenta el BIOS y cuáles son sus capacidades.

Todas las funcionalidades incluidas en dicha investigación reducen el consumo de energía de todo el sistema. Sin embargo, existen también comandos de ahorro de energía que pueden ser utilizados sin afectar al funcionamiento general del sistema. Estos pueden incluir, por ejemplo, el apagado de la pantalla —de retorno inmediato al recibir la interacción del usuario—, la detención momentánea del movimiento rotacional del disco rígido, la disminución de frecuencia de la CPU, entre otros.

Tipo de Componente	Nombre	Rendimiento	Consumo
Procesador	Sun Niagara 8-core	1.2 GHz	72 – 79 W
	Intel Pentium	2 GHz	48.9 – 66 W
DRAM	Kingston X64C3AD2 1GB	184-pin	3.7 W
	Kingston D2N3 1 GB	240-pin	2.3 W
Disco Rígido	DiamondMax 16	5400 rpm	2.9 – 7.0 W
	DiamondMax 9	7200	4.0 – 7.9 W

En el precedente recuadro, elaborado por Patterson y Hennesy [4], se puede observar el consumo energético particular de algunos componentes de hardware durante su ciclo normal de funcionamiento.

En un caso ideal, el sistema operativo que conozca a priori el uso que se le dará a los recursos de hardware puede hacer uso de estos comandos de manera óptima reduciendo el consumo de energía al mínimo posible en el que, aun, el usuario no vea afectada, de manera significativa, su experiencia respecto al funcionamiento con un consumo normal de energía.

El término “experiencia del usuario”, en nuestro caso, no se limita solo a la utilización cotidiana que haga una persona sobre una computadora —de escritorio o portátil—, sino también al uso efectuado sobre sistemas operativos instalados en servidores, de procesamiento distribuido o de procesamiento en lote.

El problema es que el citado caso ideal resulta impracticable en situaciones reales; no es posible conocer las acciones que realizará y las aplicaciones que iniciará el usuario. De ahí surgieron nuestras primeras hipótesis de trabajo:

H1 – *Es posible definir estrategias de ahorro de energía diferentes para cada usuario, que funcionen mejor en cada caso respectivamente.*

En la hipótesis 1 planteamos que, si bien no se conoce a priori lo que hará el usuario, es posible aprender del uso cotidiano, midiendo los aciertos y penalidades, y adaptar una estrategia predictiva que se mejore en cada uso. El dispositivo que aquí planteamos deberá adaptarse incluso a una alta variabilidad del comportamiento (usuario impredecible).

H2 – *Una estrategia de ahorro de energía definida a priori corre más riesgo de sufrir penalidades.*

Decimos que ocurre una penalidad cuando se toma, equivocadamente, alguna medida de ahorro de energía sobre algún componente de hardware que deba ser utilizado durante el período de tiempo en el cual se encuentra reducido su funcionamiento [Fig. 1].

Las consecuencias de una penalidad pueden implicar lo siguiente:

- Espera de tiempo hasta la puesta en actividad del dispositivo.

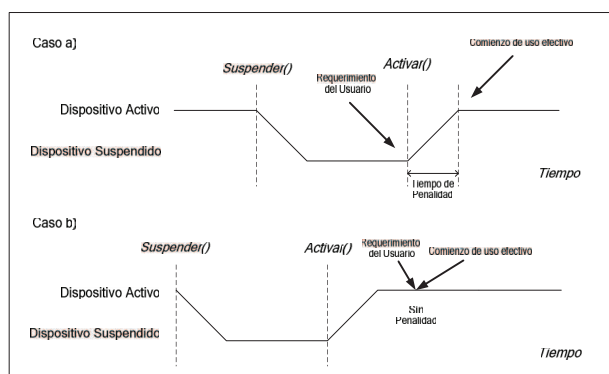


Figura 1 - a. Ejemplo de ahorro con penalidad, b. sin penalidad

- Baja en el rendimiento o en el tiempo de respuesta durante el período de actividad (especialmente al reducir la frecuencia del CPU).
- Pérdida de paquetes de red o pérdida irrecuperable de datos por recibir durante el período de inactividad.

H3 – *Ante la ocurrencia de una penalidad, siempre hay una pérdida de tiempo, una baja en rendimiento o una pérdida de información.*

Debido al problema de las posibles penalidades cuando la estrategia se define a priori, nuestro primer paso fue investigar en qué momentos y cómo se puede definir una estrategia de ahorro de energía.

La investigación nos llevó a elaborar una propuesta de diseño de sistemas operativos basada en aspectos reconfigurables [3]. Los aspectos se refieren a las divisiones funcionales del sistema, como por ejemplo, la administración de memoria, la planificación de hilos e, incluso, la gestión de energía, entre otros.

La implementación —también nos referimos a esto como *modos*— de un aspecto define su funcionamiento. Por ejemplo: la administración de memoria puede estar implementada con un método de particiones fijas, particiones variables, paginación, paginación con swapping, etcétera.

Definimos que, si alguno de estos aspectos de un sistema operativo cuenta con más de una posible implementación, entonces, se lo considera un aspecto reconfigurable.

Elaboramos un marco temporal desde el cual podemos definir la capacidad de cambiar la implementación de un aspecto por otra, de la siguiente manera:

- **Configuración en Tiempo de Compilación:** Aspectos que son reconfigurables únicamente mediante código y pueden ser cambiados luego de recompilar el sistema operativo.

- **Configuración en Tiempo de Arranque:**

En este nivel, se sitúan los aspectos que son configurables mediante un menú inicial o archivo de configuración. Para cambiar el modo de funcionamiento para el aspecto en cuestión, se deberá reiniciar el sistema operativo y seleccionarlo.

- **Configuración en Tiempo de Ejecución Limitada:**

En este nivel, se sitúan los aspectos que permiten ser configurados de manera diferente para cada proceso, pero que no pueden ser reconfigurados para este una vez iniciada su ejecución.

- **Configuración en Tiempo de Ejecución Pura:**

En este nivel, se sitúan los aspectos que permiten ser configurados en cualquier momento y aprovechados para y por un proceso durante su ejecución.

Observamos, en este punto, que, cuanto más cerca ubiquemos la reconfigurabilidad del aspecto de gestión de energía a la de tiempo de ejecución pura, menos información deberemos asumir a priori y, por lo tanto, tendremos menos tendencia a sufrir penalidades.

Definimos, entonces, que, cualquiera fuera la estrategia de ahorro de energía, esta debía ser implementada sobre los aspectos del sistema operativo que pudieran ser reconfigurables en tiempo de ejecución pura.

El siguiente paso fue precisar quién es el responsable de realizar la configuración. Consideramos que definir una estrategia general de ahorro de energía para todos los usuarios por igual provocaría más penalidades, ya que el uso del sistema podría ser diferente para cada uno. También consideramos que no debe ser una responsabilidad exclusiva del usuario, ya que este no siempre conocerá los detalles técnicos y las consecuencias de la reducción de energía sobre los componentes de hardware.

Se adoptó, entonces, como paso inicial, establecer una escalonada, en la que se utilizará una estrategia inicial pensada para la generalidad de los casos, pero que pueda ser cambiada manualmente por el usuario.

En este punto, luego de ponernos en contacto con el grupo de investigación en lenguajes y tecnologías adaptativas de la Universidad de San Pablo, pudimos introducirnos en el tema de los dispositivos adaptativos y elaborar nuestra cuarta hipótesis de trabajo:

H4 – *El sistema operativo puede adaptar la estrategia de ahorro de energía de la hipótesis 1 sobre la base del uso cotidiano que le da el usuario.*

Dado que los dispositivos adaptativos permiten, a partir de un comportamiento inicial predefinido, cambiarlo de acuerdo con ciertos estímulos —en nuestro caso, el uso

mismo del sistema— y aprender de ellos, consideramos que la hipótesis 4 puede ser respondida luego de implementar uno de ellos y medir sus resultados respecto a la estrategia escalonada básica.

Estimamos, entonces, que las tablas de decisión adaptativas permitirán que se agreguen y se eliminen acciones de ahorro de energía si las métricas indican el desuso de un recurso o si las penalidades indican que una acción (o su configuración) no debe ejecutarse, respectivamente.

Una tabla de decisión adaptativa [1] es un dispositivo que parte de la formulación de una tabla de decisión convencional, a la que se le agregan reglas adicionales que modifican —a priori o a posteriori de una transición— las reglas existentes. De esta manera, es capaz de modificar su comportamiento sobre la base de los diferentes estímulos del contexto en el se aplica.

Fue necesario, entonces, definir cuáles serían las posibles transiciones entre los diferentes modos del aspecto de gestión de energía, por lo que elaboramos, utilizando la administración de memoria como ejemplo, una propuesta para el diseño reconfigurable de aspectos del sistema operativo.

Primero, establecimos el uso de autómatas finitos para graficar todas las diferentes implementaciones de un aspecto del sistema operativo. Las transiciones de estado representan las acciones por realizar para efectuar el cambio de un modo a otro. Las transiciones son válidas solo si se cumplen las condiciones especificadas y ocurre el estímulo —al que llamamos input— que la pueda activar.

En la figura 2, se puede ver el autómata resultante de los posibles modos de administración de memoria que actualmente pueden seleccionarse en SODIUM.

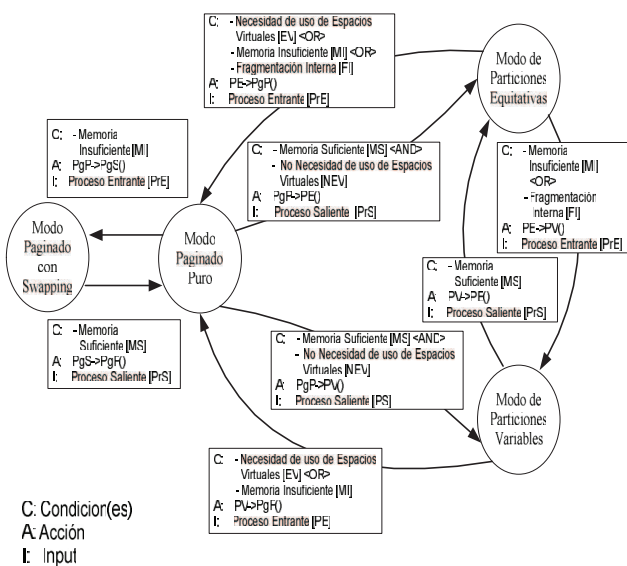


Figura 2 – Autómata del aspecto ‘Administración de Memoria’ en SODIUM

Luego, construimos una tabla de decisión estática sobre la base del autómata e indicamos condiciones, acciones e inputs en las filas, y las transiciones (reglas) como columnas [Fig. 3].

Regla N°	1	2	3	4	5	6	7	8	9	10	11	12
Condiciones	MI	x			x					x	x	
	MS			x	x			x				x
	FI		x			x						
	EV						x	x				
	NEV				x			x				
Inputs	PrE	x	x			x	x	x	x	x	x	x
	PrS			x	x			x			x	
Estado	PE	x	x			x	x	x				
	PV			x					x			
	PgP				x				x			x
Acciones	PgS									x	x	
	PE->PV()	x	x									
	PV->PE()			x								
	PgP->PE()				x							
	PE->PgP()					x	x	x				
	PgP->PV()								x			
	PV->PgP()									x		
	PgS->PgP()										x	x
PgP->PgS()												x

Figura 3 – Tabla de decisión estática ‘Administración de Memoria’ en SODIUM

Siguientes pasos de la investigación

En los siguientes meses, deberemos trabajar para implementar todas las acciones posibles de gestión de energía en la mayor cantidad de dispositivos que el hardware subyacente nos permita.

Deberemos, finalmente, elaborar un autómata y una tabla de similar manera para el aspecto de gestión de energía utilizando las acciones que describimos anteriormente al cumplirse una regla.

Antes de comenzar con las mediciones de consumo, será necesario generar una tabla de decisión adaptativa [Fig. 4] y agregar nuevas reglas que contengan acciones que, basándose en métricas, modifiquen directamente a la tabla misma y corrijan las acciones que hayan causado penalidades o que debieron haberse ejecutado con mayor frecuencia.

		ag	H	-	-	+	S	R	R	R	R	R	R	R	E	
Subject decision table	Condition	state=						T	J	J	J	J	K	L		
		material=		p1	p1	p1	p1			M	M	M				
		building=		p2	p2	p2	p2			O <sub>0</sub>	O <sub>0</sub>	O <sub>0</sub>				
		supplier=								F <sub>1</sub>	F <sub>1</sub>	F <sub>1</sub>				
		price=		BP	BP	GP	GP			BP	GP	BP				
		proximity=		NB	FB	FB	NB			NB	FB	FB				
Adaptive functions	Action	state:=		K	K	K	K	T	J	K	L	L	L	L		
		get (material)														
		get (building)														
		accept:=														
Adaptive functions	Other names	F	B	✓	✓	✓	✓									
		p1	P	✓	✓	✓	✓									
		p2	P	✓	✓	✓	✓									
		g1	G	✓	✓	✓	✓									

Figura 4 – Ejemplo de una tabla de decisión adaptativa [5] .

Problemas por resolver

Actualmente, contamos con un entorno de pruebas virtual mediante el cual resulta fácil obtener resultados

para cualquier cambio efectuado sobre SODIUM. Sin embargo, como será necesario efectuar mediciones de consumo eléctrico sobre entornos reales de hardware — notebooks y desktops—, deberemos mejorar el sistema de booteo actual para permitir el booteo real mediante dispositivos pendrive y para poder generar pruebas de uso automatizadas.

Será necesario actualizar las acciones de gestión de energía según el modelo APM con el que contamos al día de hoy hacia las establecidas por el modelo ACPI que resulta más compatible con los BIOS modernos, ya que, de otra manera, no podremos efectuar acciones avanzadas sobre los diferentes dispositivos.

## Líneas de investigación y desarrollo

Actualmente, nos encontramos trabajando sobre los siguientes aspectos:

- Implementación en SODIUM de los diferentes modelos e interfaces de gestión de energía existentes.
- Aplicación de tecnologías adaptativas sobre aspectos de un sistema operativo, en conjunto con investigadores de la Universidad Nacional de San Pablo.
- Diseño y construcción de un sistema operativo como herramienta didáctica.

## Resultados y objetivos

Hasta el momento hemos logrado lo siguiente:

- Implementar con éxito el modelo APM de gestión de energía en SODIUM.
- Elaborar un diseño de aspectos de un sistema operativo reconfigurables.
- Establecer los pasos que son necesarios para la utilización de tecnologías adaptables sobre esos aspectos.

Los objetivos de este año serán los siguientes:

- Implementar el más avanzado modelo ACPI manteniendo vigente el APM por su potencial didáctico.
- Elaborar la tabla de decisión inicial y también la adaptativa para todas las acciones que provea ACPI.
- Construir el entorno de pruebas unificado en el laboratorio con el que contamos en la UNLaM utilizando hardware compatible y osciloscopios digitales para medir el consumo energético.

- Elaborar los casos de prueba y realizarlas para comprobar o refutar cada una de las hipótesis hasta ahora planteadas.

## Formación de Recursos Humanos

Se está trabajando en conjunto con los alumnos de la Universidad Nacional de La Matanza para lograr que estos, además de los contenidos teóricos de la materia, puedan adquirir el conocimiento del diseño y construcción de un sistema operativo.

La presente línea de investigación forma parte del trabajo que el ingeniero Sergio Martin se encuentra realizando para su tesis de maestría.

Los avances relacionados con el análisis sobre la administración de memoria y el valor didáctico de las tecnologías adaptativas aplicadas a un sistema operativo dan soporte a las tesis de maestría que la licenciada Graciela de Luca y el ingeniero Nicanor Casas están realizando, respectivamente.

## Referencias y bibliografía

### Referencias

- [1] J. J. NETO, "Adaptive Rule-Driven Devices - General Formulation and Case Study" Lecture Notes in Computer Science, Vol.2494, Pretoria, South Africa, July 23-25, Springer-Verlag, 2001, pp. 234-250.
- [2] N. CASAS, G. DE LUCA, S. MARTIN, G. PUYO, W. VALIENTE, "Gestión de energía en el sistema operativo didáctico utilizando el modelo APM", XIII Workshop de Investigadores en Ciencias de la Computación, WICC 2011. Universidad Nacional de Rosario.
- [3] S. MARTIN, N. CASAS, G. DE LUCA, "Diseño de un sistema operativo reconfigurable para fines didácticos y prácticos", 6.º Workshop de Tecnología Adaptativa, WTA 2012. Universidad de San Pablo.
- [4] J. HENNESSY, D. PATTERSON, "Computer Architecture, A Quantitative Approach", 5.ª edición, septiembre 2011, The Morgan Kaufmann Series in Computer Architecture and Design, Elsevier.
- [5] PEDRAZZI, T.; TCEMRA, A. Y ROCHA, R., "Adaptive Decision Tables A Case Study of their Application to Decision-Taking Problems" ICANNGA 2005 Proceedings on Adaptive and Natural Computing Algorithms 2005, Coimbra, Portugal, March 2005.

### Bibliografía adicional

- VEITCH, A. C.; HUTCHINSON, N. C., "Kea-a dynamically extensible and configurable operating system kernel", Configurable Distributed Systems, 1996. Proceedings, Third International Conference on.
- SILBERSCHATZ, A.; GAGNE, G. Y GALVIN, P., "Operating System Concepts", 8.ª edición actualizada, julio 2011, Wiley.
- RUSSINOVIC, M.; SOLOMON, D. Y IONESCU, A., "Windows® Internals: Including Windows Server 2008 and Windows Vista", 5.ª edición, junio 2009, Microsoft Press.
- CORBET, J.; RUBINI, A. Y KROAH-HARTMAN, G., "Linux Device Drivers", 3.ª edición, febrero 2005, O'Reilly Media.
- Advanced Configuration & Power Interface (ACPI) Specification. <http://www.acpi.info/>