Generación de una biblioteca para control de tareas en tiempo real en un sistema operativo didáctico

Nicanor Casas, Graciela De Luca, Sergio Martín, Waldo Valiente, Gerardo Puyo, Federico Díaz

Universidad Nacional de la Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

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

ncasas@ing.unlam.edu.ar, {graciela.edl, eienburuu,gerardopuyo, fedediazceo}gmail.com

RESUMEN

RESUMEN

En este trabajo de investigación se estudiaron los métodos de planificación de tiempo real y se buscó generar una biblioteca en busca de aportar soluciones a algunas de las cuestiones la política de planificación de tiempo real. Se dispusieron herramientas teóricas que permitieron conocer si el sistema podrá en todo momento, garantizar la correcta ejecución de todas las tareas críticas. Los planificadores basados en prioridades dinámicas, a pesar de ser capaces de garantizar un mayor número de tareas que los de prioridades estáticas, no disponen de un test de planificabilidad eficiente.

Los sistemas de tiempo real están compuestos tanto por tareas periódicas, que suelen asociarse a actividades criticas, como por tareas aperiódicas, sin ninguna urgencia en su ejecución, es deseable que estas se completen lo antes posible sin poner en peligro los plazos de las periódicas. Se han propuesto dos algoritmos para servir tareas aperiódicas basados en el concepto de holgura. El primero de ellos se apoya en una tabla pre calculada para aceptar las peticiones aperiódicas. El segundo, a cambio de reducir la complejidad espacial, tiene mayor coste temporal pues realiza todos los cálculos dinámicamente. Ambos algoritmos ofrecen, a las tareas aperiódicas, el menor tiempo de respuesta posible.

PALABRAS CLAVE

Tiempo real, Granularidad, Baseline, Tecnologías Adaptativas

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

INTRODUCCIÓN

El desarrollo de un sistema operativo didáctico de características generales y de estructura tradicional escrito por los alumnos del Departamento de Ingeniería e Investigaciones Tecnológicas de la carrera de Ingeniería en Informática que recibe el nombre de SODIUM (**S**istema **O**perativo **D**epartamento **I**ngeniería **U**niversidad de La **M**atanza) llegó a un punto en que debía realizarse un análisis de código a fin de lograr una mejor respuesta, pero principalmente estandarizar los diversos programas.

Fue así que se comenzó por sacar del KERNEL del sistema operativo SODIUM, llamadas al sistema que efectuaban diversas funciones colocándolos en área de usuario., lo que mejoró la comprensión de su función y/o funciones y permitió una mayor estabilidad durante la ejecución del sistema operativo.

Se pensó entonces, en la posibilidad de pasar a un sistema operativo de tiempo real (TR), lo cual llevó a realizar un estudio para detectar diferencias entre lo existente y lo nuevo para saber cuáles son los pro y los contra de incluir o modificar el actual sistema operativo y llevarlo a un sistema operativo en tiempo real.

Para realizar la aplicación del pasaje de un sistema operativo de características tradicionales a otro sistema

operativo de tiempo real se debe basar en un estudio exhaustivo de las diferencias existentes en cada uno de ellos.

Esto nos llevó a la disyuntiva de tener que optar por tres opciones: la primera es generar otro sistema operativo de estudio; la segunda es establecer uno solo que permita la convivencia de los dos sistemas y la tercera transferir el sistema operativo actual a un sistema operativo en tiempo real exclusivamente. En cada una de las opciones se incluye la aplicación de tecnologías adaptativas.

La primera opción, generar otro sistema operativo, tiene las siguientes desventajas:

a) Que al momento se están desarrollando nuevas funcionalidades definidas para el sistema operativo tradicional que está pasando por alguna de la siguientes etapas: de escritura, de test o de implementación que no podrían ser establecidas para el nuevo.

b) Realizar una copia del sistema operativo tradicional para luego generar las estructuras necesarias para convertirlo en tiempo real no nos garantizan que las mismas funcionen al estar reestructuradas como servicios, para el primero.

c) Que toda la programación es desarrollada por los alumnos que cursan la materia Sistemas Operativos, lo que impide solicitar modificaciones porque las mismas podrían comprometer la cursada.

d) Generar dos sistemas operativos el mantenimiento separado de los mismos requeriría esfuerzos muy importantes que demorarían la estabilidad del sistema.

e) Que se complica la posibilidad de realizar comparaciones de performance y funcionamiento entre ambos sistemas ya que deberíamos proveer de un tercero que tomara los datos de ambos para unificarlos y facilitar las conclusiones.

La segunda, establecer un solo sistema operativo que permita la convivencia de los demás, tiene los siguientes puntos a favor:

a) Para que sea un sistema operativo didáctico se deben tener los dos sistemas ejecutándose de acuerdo a las necesidades de los alumnos a fin de que se puedan establecer las comparaciones necesarias.

b) Permitiría realizar un mantenimiento más acorde a las fuerzas de desarrollo con que cuenta el equipo reduciendo los tiempos de test y de corrección que conlleva cada avance.

Por otro lado tiene como contrapartida los siguientes puntos:

a) Complica la instalación del sistema operativo debido a que el volumen del mismo hace que tenga un peso importante

b) Es necesario que partes importantes del Kernel que se desarrollaron y que son compiladas en forma monolítica deban ser compiladas en forma separadas para que los mismos se brinden en función de servicios.

c) Que el punto anterior, en muchos de los casos sería como generar un nuevo sistema operativo.

La tercera opción, convertir el actual sistema operativo a uno de tiempo real tiene a favor:

a) Que nos permitiría fijar la atención en un solo desarrollo.

b) Que los nuevos sistemas operativos están ya prácticamente enrolados en esta característica.

Tienen como contrapartida que:

a) En las cátedras actuales de la materia Sistemas Operativos se sigue trabajando obre sistemas operativos tradicionales como punto de partida.

Como consecuencia de lo expuesto se llegó a la siguiente conclusión:

1) Generar una función mixta en la cual se mantendría al sistema operativo tradicional pero separando algunos servicios del Kernel actual para que se encuentren ubicadas a nivel de usuario, con el propósito de que las mismas cumplan la función especificada y no influyan en la performance de un modelo u otro.

2) Se decidió que se duplicarían las funciones de los diferentes algoritmos para evitar que los sistemas operativos deban ser compilados cada vez que se quiera probar una función en performance.

3) Aplicación de tecnologías adaptativas para permitir la transmutación de uno a otro de los algoritmos involucrados como también de la elección del tipo de sistema operativo que se quiere utilizar.

Se llegó a la conclusión de que era factible realizar, en primera instancia, una conjunción entre el Kernel del sistema operativo vigente y agregar los algoritmos de tiempo real utilizando para ello tecnologías adaptativas.

Se realizaron los análisis correspondientes a los diferentes algoritmos que se van a aplicar, teniendo en cuenta que serán en principio los mismos que se utilizan en los sistemas operativos tradicionales, como base de selección, para poder realizar comparaciones necesarias con el objetivo de marcar las diferencias entre los diferentes sistemas operativos.

Sin embargo es propio aclarar que no todos los algoritmos de planificación pueden verificar todos los eventos. Si tomamos como ejemplo los algoritmos FIFO, SJFS o PRORIDADES (non-preemptive) no pueden evaluar los cambios de estado ya que son algoritmos no expropiativos lo que implica que cambiar de algoritmo en medio de la ejecución de un proceso implicaría violar su integridad, porque al cambiarlo en medio de su ejecución se perderían los beneficios que intrínsecamente cada uno tiene.

Por otro lado no son sensibles a la creación de un proceso, ya que estos se encolarán a través de su propio criterio, pero no afectarán la ejecución actual.

Como los eventos a utilizar serán únicamente dependientes del Modo Actual y no al modo en que solicitarán las transiciones será posible codificar funciones en el SODIUM que automáticamente monitoreen los eventos necesarios según el algoritmo de ejecución. Así no hará falta incluir eventos como condiciones de la tabla de decisión final.

Una de las características más importantes del proyecto es que la codificación de los módulos principales que implementarán en el sistema operativo será realizada por los alumnos que cursan la materia Sistemas Operativos Avanzados.

Esto implica un consumo de tiempo elevado, debido a:

- Que los alumnos interactúan con otras materias y con sus responsabilidades laborales lo que determina una carga importante en el desarrollo del sistema operativo.
- Que los nuevos cursos deben tener capacitación sobre lo ya desarrollado para poder continuar con la generación del sistema operativo.
- Generar los trabajos prácticos necesarios para un normal desarrollo del proyecto.

Se presenta entonces la necesidad de acortar tiempos para que el producto pueda estar disponible en un lapso prudencial para que cumpla con los requerimientos para los que fue ideado, siendo la responsabilidad del equipo de investigación de generar los puentes necesarios para amalgamar los diferentes programas generados por los alumnos, como el de preparar la base necesaria para que los próximos cursos puedan continuar con la tarea emprendida.

Uno de esos puentes es la de construir una biblioteca que localice y circunscriba los desarrollos referentes a tiempo real

PROBLEMAS A RESOLVER

La generación de una biblioteca de tareas que permita el control de planificaciones estáticas en tiempo real se constituyó en un trabajo de gran envergadura por los inconvenientes que se presentaron en la misma.

En un principio la generación de procesos de poca complejidad e independientes a los que se les asignó un prioridad externa, y sin posibilidades de modificación (prioridades internas) pareció ser un elemento simple pero posteriormente se fue complicando en forma escalonada.

La biblioteca está basada en un planificador de características expulsivas (premptive) que provee una significativa garantía cuando ciertas condiciones de utilización del procesador son reunidas.

En principio las características de construcción de la biblioteca se establecieron en base a programas escritos en lenguaje C que interactuaban en área del Kernel del sistema operativo lo que para programas de usuario cortos no presentaban problemas teniendo un tiempo de respuesta interesante por ser una primera aproximación, aunque pasó a ser lento cuando se incorporaron procesos usuario más sofisticados. Esto hizo que una de las premisas establecidas anteriormente (cambio de tarea – context switch – nulo o despreciable) no se cumplieran, por lo que se está pasando a una colección de procesos livianos, en los cuales los hilos trabajen en área de usuario para obtener una mayor velocidad de respuesta y minimizar los tiempos de context switch, con lo que esperamos lograr una eficiencia máxima.

La biblioteca de tareas comprende un planificador rate monotonic que puede proporcionar una planificación eficiente cuando se cumplen ciertas condiciones en la utilización del procesador.

La diferencia entre un sistema de computación en tiempo real y un sistema de computación de propósitos generales se encuentra no en las especificaciones de performance, pero si en la importancia que cada sistema, en lo referente a tiempos, tiene como consideraciones.

En aplicaciones de tiempo real la exactitud de una computación depende no solo del resultado de la computación sino también del tiempo en que las salidas son generadas.

Pensamos que para poder medir la importancia del desarrollo de la biblioteca tenemos que tener en cuenta los siguientes puntos:

a) Velocidad predecible para la respuesta a eventos urgentes.

b) Alto grado de planificación. Definida planificación como el alto grado de utilización de recursos o el menor tiempo de requerimientos que puedan ser garantizados para todos los procesos intervinientes.

c) Estabilidad bajo carga transitoria. Cuando el sistema es sobrecargado por eventos, el cumplimiento de todos los plazos se hace imposible. Es en este caso en que la biblioteca debe garantizar los plazos para tareas críticas.

Un sistema operativo de tiempo real debe satisfacer estas aplicaciones de computación con plazos implícitos, a través de la fuerza bruta implementada a través de hardware o por un golpe de suerte. Históricamente los más recientes sistemas operativos de tiempo real intentan ser "realmente rápidos" en lugar de ser de "tiempo real". Esto es simple de explicar pero no tan simple de implementar. Lo último significa que la ejecución en tiempo de los servicios y las operaciones internas de un sistema operativo se ejecuten lo más rápidamente posible, para minimizar el tiempo de ejecución promedio y de esa manera tener un relativamente predecible límite superior para el peor caso de tiempo de ejecución.

También hay que tener en cuenta que esos supuestos a menudo no son explícitamente identificados o siquiera conocidos. Cada sistema puede operar satisfactoriamente en tiempo real y proveer soluciones específicas para ciertas aplicaciones.

Tradicionalmente, un sistema en tiempo real usa funciones cíclicas para planificar hilos de ejecución concurrentes. Bajo este acercamiento, un programador puede establecer una línea de tiempo de ejecución a mano para serializar la ejecución de secciones críticas y reunir plazos de tareas.

Todo esto viene a establecer que las funciones manejables para sistemas simples se torna terriblemente inmanejable para sistemas grandes.

Todo esto implica un doloroso proceso de desarrollo de código de modo que se ajuste a los intervalos de tiempo de un ciclo ejecutivo asegurando al mismo tiempo que la sección crítica de diferentes trabajos no se intercalen.

Conclusiones

El trabajo debe ser continuado con un proyecto que abarque desarrollos para más de un procesador y con un incremento del grupo de investigación para poder unificar las diferentes investigaciones que están produciendo los alumnos, que divididos en distintos grupos, generan más información que la que el grupo puede concentrar, lo que hace lenta la aparición de nuevas versiones del sistema operativo SODIUM completas, ya que la prioridad de las incorporaciones se realiza de acuerdo a los próximos compromisos asumidos con los nuevos cursantes

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

[ACI08] – Aciti, Claudio - Una Panorámica del Problema de Inversión de Prioridades en Sistemas Operativos de Tiempo Real - INTIA/INCA - Depto. de Computacion y Sistemas - Facultad de Ciencias Exactas Universidad Nacional del Centro de la Provincia de Buenos Aires (7000) - Tandil - Argentina

[ALF12] – Alfenas Daniel Assis; Danilo Picagli Shibata; Marcos Ribeiro Pereira-Barretto; João José Neto – Sistemas de Markov Adaptativos: Formulação e Plataforma de Desenvolvimento – 6to. Workshop de Tecnología Adaptativa – San Pablo – Brasil 2012.

[ANG98] – Angulo José M y Funke Enrique M – Microprocesadores Avanzados 386 y 486 – 1998 – Editorial Paraninfo – Cuarta Edición.

[ATT08] Attiya Hagit, Leah Epstein, Hadas Shachnai, Tami Tamir. "Transactional Contention Management as a Non-Clairvoyant Scheduling Problem". Springer Science+Business Media, LLC 2008

[BRE01] – Brey Barry B – Los microprocesadores INTEL – 2001 - Prentice Hall – Quinta Edición

[BRA96] Bradford Nichols, Dick Buttlar & Jacqueline Proulx Farrell, "Pthreads Programming A POSIX Standard for Better Multiprocessing", Addison-Wesley, 1996.

[BUR03] Burns, A. "Sistemas de tiempo real y lenguajes de programación", Addison Wesley 2003.

[CAS07] - Casas Nicanor, Martín Cortina, Graciela De Luca, "Desarrollo de un sistema operativo

didáctico". *XIII Congreso Argentino de Ciencia de la Computación*. Chaco, Argentina, 2007.

[CAS08] - Casas Nicanor, Graciela De Luca, Martín Cortina, Gerardo Puyo, Waldo Valiente, "Implementación de distintos tipos de memoria en un sistema operativo didáctico". *XIV Congreso Argentino de Ciencia de la Computación*. La Rioja, Argentina, 2008.

[CAS11] - Casas Nicanor, Graciela De Luca, Sergio Martin, Gerardo Puyo, Waldo Valiente, "Gestión de energía en el sistema operativo didáctico utilizando el modelo APM". *XIII Workshop de Investigadores en Ciencias de la Computación*. Santa Fe, Argentina, 2011.

[EDM08] Edmonds Feff. "How to think about Algorithms". Cambridge University Press. First Edition. 2008.

[GUY07] Guy Even, Magnús M. Halldórsson, Lotem Kaplan, Dana Ron, "Scheduling with conflicts: Online and Offline Algorithms"

[HOL97] Hollstein Thomas, Andreas Kirschbaum y Manfred Glesner. "A prototyping environment for fuzzy controllers", pp:482-490, Lecture Notes in Computer Science 1304, Ed. Springer, FPL, London, Sep 1997.

[KNU68] Knut Donald E. "The Art of computer programming". Standford University. Addison-Wesley Publishing Company. First Edition. 1968

[KOR09] Korhonen Vesa. "TIES246 Real-time systems". JAMK University of Applied Sciences, School of Business Administration, Mankola. Edition 2009 GSM 0400 451 752

[LIU00] Liu, Jane W. S. "Real-Time Systems", Prentice Hall, 2000.

[MAR12a] – Martín, Sergio – Aplicación de Tecnologías adaptativas en un sistema operativo reconfigurable – Tesis de Maestría en Informática – Universidad Nacional de la Matanza – Diciembre 2012

[MAR12b] - Sergio Martin, Casas Nicanor, Graciela De Luca, J. J. Neto, "Utilización de tecnologías adaptativas para la gestión de la energía de un sistema operativo didáctico". *XIV Workshop de Investigadores en Ciencias de la Computación*. Misiones, Argentina, 2012.

[PAT06] Patchrawat "Patch" Uthaisombut. "Generalization of EDF and LLF: Identifying All Optimal Online Algorithms for Minimizing Maximum Lateness". Springer Science+Business Media, LLC 2007

[RAM94] Ramamritham Krithi, and John a. Stankovic. "Scheduling Algorithms and Operating Systems Support for Real-Time Systems". Proceedings of the IEEE, vol. 82 N° I, 1994

[RIM93] Rimvall, C.M. and Jobling, C.P. "Computer Aided Control System Design". IEEE Control Systems Magazine, volume 13, pag 14, 1993, IEEE.

[SEN02] – Senart Aline, Olivier Charra, Jean-Bernard Stefani, "Developing dynamically reconfigurable operating system kernels with the THINK component architecture". W*orkshop on Engineering Context-aware Object-Oriented Systems and Environments*. Washington, Estados Unidos, 2002.

[SIL12] – Silberschatz Abraham, Peter Galvin, Greg Gagne – Operating System Concepts – Jhon Wiley & Sons – New Jersey, Estados Unidos, 2012 – Octava Edición

[STO81] - Stonebraker Michael "Operating system support for database management". *Communications of the ACM*, 24(7) – ACM New York, Estados Unidos, 1981.

[TAN97] – Tanenbaum Andrew S y Woodhull Albert S – Operating systems Design and implementation – 1997 – Prentice Hall – Segunda Edición.

[TCH08] - Tchemra Angela Hum – "Aplicação da Tecnologia Adaptativa em Sistemas de Tomada de Decisão: Uma Abordagem Estratégica na Seleção de Fornecedores" – *Segundo Workshop de Tecnologia Adaptativa*. Escola Politécnica, USP, San Pablo, Brasil, 2008.

[TCH09] - Tchemra Angela Hum – "Tabela de decisão adaptativa na tomada de decisões multicritério". – *Tesis de doctorado*. Escola Politécnica, USP, San Pablo, Brasil, 2009.

[TCH10] – Tchemra, Angela Hum – Adaptatividade na tomada de decisão multicritério – 4to Workshop de Tecnología Adaptativa - Escola Politécnica, USP, Sao Pablo, Brasil 2010

[TIN98] – Tinetti Fernando G y De Giusti Armando – "Procesamiento Paralelo Conceptos de Arquitecturas y Algoritmos.

[VEI96] – Veitch A. C.; N.C. Hutchinson – "KEA" - A dynamically extensible and configurable Operating System Kernel – 3rd. International Conference on Configurable Distributed System – Vancouver – Canada, 1996

[VEI98] – Veitch A. C. – A dynamically reconfigurable and extensible Operating System – Tesis de doctorado, Department of Computer Science, The University of British Columbia, Canada, 1998

[WOJ07] Wojciech Jawor, Marek Chrobak, Christoph Dürr. "Competitive Analysis of Scheduling Algorithms for Aggregated Links". 2005. Springer Science+Business Media, LLC 2007