**ANÁLISIS DE LAS COMPETENCIAS, COMPORTAMIENTOS Y DERIVACIONES DE PLANIFICADORES DE PROCESOS EN TIEMPO REAL EN UN SISTEMA OPERATIVO DIDÁCTICO**

Ing. Hugo Ludovico Ryckeboer Ing. Nicanor Blas Casas

Director Codirector

hugor@ing.unlam.edu.ar ncasas@ing.unlam.edu.ar

Lic. Graciela De Luca - Ing. Sergio Martín - Ing. Gerardo Puyo - Ing. Waldo Valiente

**Resumen**

Este trabajo de investigación se concentró en la tarea del análisis de los pasos necesarios para obtener la conjunción de dos sistemas que trae aparejado una serie de inconvenientes en especial para la evaluación de rendimientos, debido a que están sustentados en una misma plataforma existiendo tiempos iguales para funciones distintas.

Para que los planificadores de caracterísiticas diferentes puedan funcionar juntos se necesita que la aplicación de tecnologías adaptativas **[TCH10]** genere los cuadros de relación suficientes y que los mismos incluyan todas las variables necesarias aunque las mismas puedan no ser llevadas posteriormente a la práctica.

La necesidad de lograr una función didáctica superadora a la anterior, se decidió realizar una serie de modificaciones a la estructura monolítica que hasta este momento caracterizaba al SODIUM. Fue así que se comenzó por sacar del KERNEL, llamadas al sistema que efectuaban diversas funciones colocándolos en área de usuario. Se pensó entonces, en la posibilidad de pasar a un sistema operativo de tiempo real **[BUR03]**, 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 **[LIU00]**.

**Opciones planteadas**

Esto nos llevó a la disyuntiva de tener que optar por cuatro opciones teniendo en cuenta que en cada una de las opciones se incluye la aplicación de tecnologías adaptativas.

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

1. La experiencia recogida permite obtener un producto mejor estructurado y dinámico
2. Se pueden aplicar técnicas de tecnologías adaptativas desde un comienzo, no como en el actual que sólo los últimos módulos la contemplan.

Por otro lado tiene como desventaja los siguientes items:

1. Que al momento se están desarrollando nuevas funcionalidades definidas para el sistema operativo tradicional que están en test o en implementación no podrían ser establecidas para el nuevo.
2. 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.
3. El mantenimiento por separado de dos sistemas operativos requeriría esfuerzos muy importantes que demorarían la estabilidad del sistema.
4. El mantenimiento separado de los mismos también requiere esfuerzos importantes.
5. 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:

1. 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.
2. 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:

1. Complica la instalación del sistema operativo debido a que el tamaño del mismo hace que tenga un peso importante
2. 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.
3. 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:

1. Que nos permitiría fijar la atención en un solo desarrollo.
2. Que los nuevos sistemas operativos están ya prácticamente enrolados en esta característica.

Tienen como contrapartida que:

1. En las cátedras actuales de la materia Sistemas Operativos se sigue trabajando sobre sistemas operativos tradicionales como punto de partida.

La cuarta *modificar el Kernel del SODIUM y realizar una biblioteca de System Calls* para tiempo real tiene por ventajas:

1. No se requiere una programación compleja para adaptar el actual KERNEL del SODIUM a los requerimientos de la planificación en tiempo real.
2. Sobre la base de las llamadas al sistema originales se puede implementar las nuevas focalizando su comportamiento en requisitos de tiempo real tales como la previsibilidad y reducción del overead.

Como contrapartida es posible enumerar lo siguiente:

1. El sistema operativo que surja de esa estructura dificultará los procedimientos de comparación entre las performances de los diferentes algoritmos.
2. La parte didáctica del sistema sólo mostrará las utilidades de los algoritmos.
3. Hay que designar a un integrante del equipo para realizar el seguimiento del análisis posterior para la implementación del sistema.

Como consecuencia de lo expuesto se llegó a las siguientes decisiones:

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. Duplicar las funciones de los diferentes algoritmos para evitar que los sistemas operativos deban ser compilados cada vez que se quiera probar performance de una función.
3. Aplicar 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.

Esto derivó en que era factible realizar, en primera instancia, una conjunción entre el Kernel del sistema operativo vigente y agregar los algoritmos de tiempo real.

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 PRIORIDADES (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 explicación de acciones emprendidas para llevar a cabo la investigación.

* Investigación de resultados de otras experiencias aplicables a nuestra propuesta.
* Recibir información detallada de los pasos que se están ejecutando, en el momento que se lo requiera, en el sistema durante su operación.
* Modelado lógico del objeto de estudio utilizando métodos estandarizados.
* Elegir en cada arranque los parámetros que lo configuren de una manera particular.
* Configurar en tiempo de ejecución algoritmos diferentes para realizar pruebas.

Validación teórica del modelo mediante walkthroughs usando lotes de datos de prueba

**Resultados**

Podemos evaluar los resultados obtenidos en estos dos años desde el punto de los logros obtenidos y de los problemas presentados en el desarrollo

1. Se incrementó el conocimiento de los planificadores cíclicos y de prioridades tanto estáticas como dinámicas **[ACI08]**, permitiendo realizar análisis y comparaciones para trabajar con diferentes casos de estudio.
2. Se estructuraron los métodos necesarios para que convivan los diferentes planificadores **[RAM94]**, lo que permitió la generación de una biblioteca para control de tareas en tiempo real.
3. Se logro mejorar el diseño de las System Calls conjuntamente con las interrupciones, sirviendo este aporte al sistema operativo original.
4. No se logró unificar los diferentes planificadores por lo que los mismos están en la actualidad ubicados en distintas plataformas.
5. Los programas realizados no pudieron ser probados con un alto grado de multiprogramación.
6. Solo pudimos realizar ejecuciones con máquinas con un solo procesador, que son las que tenemos disponibles para realizar las pruebas.

**Conclusiones**

El trabajo logrado debe ser continuado aprovechando la experiencia adquirida por el equipo en adaptatividad, en Sistemas Operativos y Planificadores de Tiempo Real con un proyecto que abarque desarrollos para más de un procesador y con un incremento del grupo de investigación para poder unificar las diferentes líneas de investigaciones que están produciendo los alumnos en distintos grupos de trabajo, lo que está retrasando un poco la aparición de nuevas versiones completas del sistema operativo SODIUM.

**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

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

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

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

**[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.

**[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.

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

**Publicaciones en Congresos, Workshops y Revistas.**

“Generación de una biblioteca para el control de tareas en tiempo real en un sistema operativo didáctico”, XV Workshop de Investigadores en Ciencias de la Computación (WICC), Universidad Autónoma de Entre Ríos, Paraná, Entre Ríos. 18/04/2013 ISBN: 978‑987‑28179‑6‑1

**[MAR12]** 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.

“Use of Extended Adaptive Decision Tables on Reconfigurable Operating Systems”, 7° Workshop de Tecnología Adaptativa (WTA 2013) en Universidade de São Paulo, Brasil. 06/02/2013.

“Diseño de un sistema operativo reconfigurable para fines didácticos y prácticos”, 6° Workshop de Tecnologia Adaptativa (WTA 2012) en Universidade de São Paulo, Brasil 26/01/2012 , ISBN: 978-85-86686-66-5

Projeto de um sistema operacional reconfigurável para fins didáticos e prácticos Revista de Sistemas e Computação v.2, n.2 Páginas: 120-129 (10) Publicado Por: Universidade Salvador (Bahia, Brasil) ISSN: 2237-2903 Año 2012-