

●David Luis La Red Martínez y Nelson Fabián Rodríguez.
Argentina

Sistema Abierto Para La Enseñanza de Sistemas Operativos Mediante un Simulador

Open System for Teaching Operating Systems by Using a Simulator

RESUMEN

El estudio y comprensión de los algoritmos de gestión de recursos computacionales por parte de los sistemas operativos es fundamental para una adecuada asimilación de la importancia de los sistemas operativos como administradores de recursos. Esto ha motivado la realización de un simulador basado en la web, pero también descargable y ejecutable localmente sin conexión a Internet, que permita estudiar el comportamiento y comprender dichos algoritmos, el que se ha basado en applets de Java y ha resultado de utilidad para los alumnos dada su sencillez y amena presentación, aspectos estos muy valorados por los alumnos. Cada algoritmo implementado está acompañado de una descripción resumida de los fundamentos teóricos que lo sustentan y de las indicaciones para su utilización, la que además es muy intuitiva. Los algoritmos implementados en el simulador con los siguientes: planificación del procesador, estrategias de reposición de páginas en la memoria principal, análisis del rendimiento de un subsistema de disco, ejecución concurrente de hilos y sincronización y sincronización de procesos en sistemas operativos distribuidos.

ABSTRACT

The study and understanding of the algorithms of computational resources management by operating systems, is essential for proper assimilation of the importance of operating systems, such as resource managers. This has motivated the realization of a simulator based on the web, downloadable and executable locally without Internet connection; the simulator allows studying the behaviour and understanding these algorithms, is based on Java applets and has proved useful for students, given its simplicity and entertaining presentation, highly valued aspects by the students. Each algorithm implemented is accompanied by a short description of the theoretical foundations underpinning it and indications for use, which is also very intuitive. The algorithms implemented in the Simulator are as follows: planning of the processor, strategies of replacement of pages in main memory, the performance analysis of a disk subsystem, concurrent execution of threads and synchronization and synchronization of processes in distributed operating systems.

PALABRAS CLAVE / KEYWORDS

sistemas operativos, simulador didáctico, enseñanza-aprendizaje, aprendizaje basado en web, sistema abierto, simulador abierto. operating systems, training simulator, teaching and learning, web-based learning, open system, open simulator

1.- Introducción

Este trabajo fue realizado teniendo como objetivo principal que los alumnos de una cátedra universitaria de Sistemas Operativos puedan ejecutar simulaciones de algoritmos de administración de recursos de los sistemas operativos en un entorno web. Estos algoritmos corresponden distintos casos de estudio analizados frecuentemente en actividades de laboratorio de las mencionadas cátedras.

Para lograr esto se desarrolló una serie de applets en el lenguaje Java, que se ejecutan desde un sitio web, o localmente, en donde se podrá observar y estudiar el funcionamiento de algunos algoritmos de administración de recursos de los sistemas operativos.

Este trabajo presenta para cada caso de estudio un marco teórico explicando el funcionamiento de los algoritmos desarrollados y analizados.

2.- Marco teórico

En los últimos años se han realizado numerosos trabajos relacionados con la producción de e-contenidos y su disponibilidad a través de plataformas de e-learning, pero no en lo referente a simuladores didácticos para la enseñanza – aprendizaje de los sistemas operativos. Actualmente se tiene una concepción global e integral del e-learning, que trasciende a la mera disponibilidad de contenidos para el aprendizaje en cualquier momento y lugar (Nichols, 2008). La concepción actual es que queda mucho por hacer en la reingeniería de los procesos de aprendizaje para explotar la tecnología superando la mera representación de contenidos y su disponibilidad para ser compartidos, debiendo ofrecerse escenarios de aprendizaje nuevos (Motschnig-Pitrik & Holzinger, 2002), (Papert, 1999). Estos escenarios incluyen el aprendizaje combinado (blended learning o b-learning), donde se combina la utilización de herramientas de e-learning con la interacción presencial alumno - docente.

En este sentido, en este trabajo se considera que las modernas tecnologías de la información y comunicación (TIC) tienen el potencial para desempeñar un papel importante al permitir un abordaje más eficaz, en el sentido de los procesos de aprendizaje más profundos y más persistentes (Motschnig-Pitrik & Holzinger, 2002), mientras el peso de un aprendizaje efectivo permanece con las personas, sus capacidades y valores interpersonales (Derntl et al., 2011). Resumiendo, que la tecnología ha contribuido a proporcionar mayor espacio para la auto-dirección, la interacción significativa en clase y experiencias de aprendizaje más ricas (Derntl & Motschnig-Pitrik, 2005).

3.- Objetivos y Metodología

Se consideraron los principales algoritmos de planificación del procesador, donde se puede comprender el funcionamiento y los distintos estados de los procesos internos de un sistema operativo convencional. Se contemplaron distintas estrategias de planificación, como ser, FIFO, Round Robin, HRN y RNM, cada una de estas explicadas con detalle (Deitel, 1993), (La Red Martínez, 2004), (Tanenbaum, 2009), (Gagne et al., 2006), (Stallings, 2005). Se llevó a cabo la simulación por medio del desarrollo de un applet, donde se debe introducir la cantidad de procesos, la cantidad de ciclos de control y las estrategias que se desea simular, de esta manera se podrá observar un informe detallado de cada una de las estrategias seleccionadas al inicio de la simulación.

Se incluyeron las estrategias de reposición de páginas en la memoria principal determinadas por el esquema FIFO, detallándose algunas de sus características y estudiándose la llamada anomalía Belady o anomalía FIFO (Aggarwal & Chandra, 1988), (La Red Martínez, 2004), (Tanenbaum, 2009), (Vitter, 2008), mediante un applet donde se deberá introducir la cantidad de marcos de página para iniciar la simulación; se podrá observar el resultado de la simulación por medio de reportes escritos y gráficos, teniendo en cuenta distintas cargas de trabajo.

Se implementó el análisis del rendimiento de un subsistema de disco de una petición a la vez (La Red Martínez, 2004), (Tanenbaum, 2009), (Tanenbaum & van Steen, 2008), donde se puede estudiar el rendimiento de discos con un solo brazo cuando es sometido a distintas cargas de trabajo. Se simula conjuntos de peticiones de operaciones de acceso a los discos, que configuran cantidades de trabajo distintas a ser atendidas, calculando los tiempos de llegadas, tiempo de servicio, tiempo de espera, capacidad de la cola, etc. Al applet que soporta esta simulación se deben ingresar los valores representativos de la carga de trabajo a simular, produciéndose informes detallados de los cálculos efectuados, un análisis estadístico de los resultados y un gráfico donde se muestran los datos obtenidos.

También se implementó el análisis del rendimiento de un subsistema de disco de varias peticiones a la vez, que a diferencia del anterior, contempla el estudio del rendimiento de un disco con varios brazos independientes entre sí para atender solicitudes. En este caso también se realiza la simulación calculando los tiempos de llegadas, tiempo de servicio, tiempo de espera, capacidad de la cola, etc. Se muestran informes detallados de los cálculos efectuados, un análisis estadístico de los resultados y un gráfico de los datos obtenidos.

Además se implementó el estudio de la ejecución concurrente de hilos y la sincronización de los mismos (Deitel, 1993), (La Red Martínez, 2004), (Tanenbaum, 2009), (Stallings, 2005). Se desarrolló un applet que implementó el problema de procesos productores y consumidores, donde los procesos productores generan información en un buffer y los procesos consumidores retiran información del mismo buffer. El applet se diseñó de manera tal que permitiera generar en un arreglo el seguimiento de dicha simulación para posteriormente efectuar el análisis de la forma en que se ha desarrollado la misma.

Se desarrolló además un simulador de algoritmo de sincronización de procesos en sistemas operativos distribuidos para el uso de recursos críticos con exclusión mutua (La Red Martínez, 2004), (Tanenbaum, 2009), (Gagne et al., 2006), (Stallings, 2005). En este caso se planteó un esquema de control centralizado similar al de los sistemas operativos convencionales, donde existe un proceso coordinador para un conjunto determinado de procesos. El proceso coordinador actúa como administrador, otorgando el recurso crítico en el caso de que esté libre, bloqueando en el caso de que el recurso crítico esté siendo usado por otro proceso y desbloqueando los recursos críticos en el caso de que algún proceso termine de forma anormal cuando tenga apropiado algún recurso crítico. La simulación es realizada por medio de un applet que hace uso de hilos para representar a los procesos y de arreglos para representar cada región crítica. Para dar inicio a la simulación se debe introducir la cantidad de procesos, la cantidad de regiones críticas y el tiempo de simulación en segundos.

Asimismo se desarrolló una aplicación web para contener a los applets que simulan cada uno de los algoritmos de administración de recursos por parte de los sistemas operativos.

4.- Resultados

En las figuras 1 y 2 se muestran las principales pantallas del simulador de algoritmos de planificación del procesador.



Figura 1: Pantalla principal.



Figura 2: Resultados de la simulación.

Las figuras 3 y 4 muestran las principales pantallas del simulador de la reposición de páginas de memoria mediante la estrategia FIFO, lo que eventualmente podría conducir a la anomalía Belady.



Figura 3: Pantalla principal.



Figura 4: Resultados de la simulación.

Las figuras 5 y 6 muestran las principales pantallas del simulador de un subsistema de discos de una petición a la vez.

Las figuras 7 y 8 muestran las principales pantallas del simulador de un subsistema de discos de varias peticiones a la vez.



Figura 5: Pantalla principal.



Figura 6: Resultados de la simulación.



Figura 7: Pantalla principal.

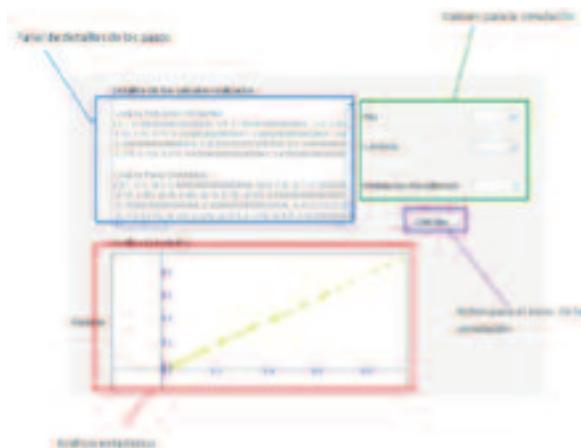


Figura 8: Resultados de la simulación.

Las figuras 9 y 10 muestran las principales pantallas del simulador de concurrencia e hilos.

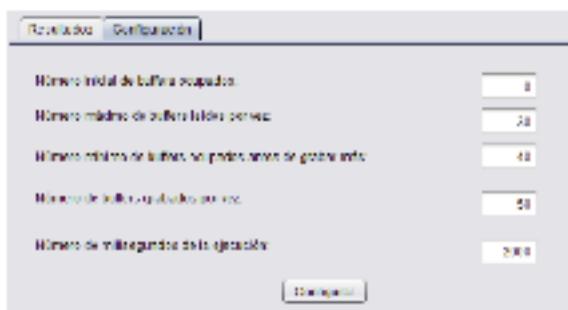


Figura 9: Pantalla principal.

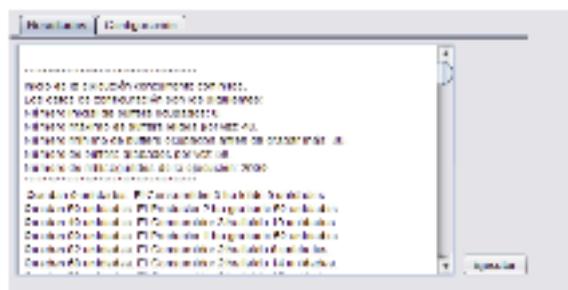


Figura 10: Resultados de la simulación.

Las figuras 11 y 12 muestran las principales pantallas del simulador de sincronización de procesos y exclusión mutua.

En las figuras 13 y 14 se muestran las principales pantallas del aplicativo que contiene a los applets que realizan las distintas simulaciones.



Figura 11: Pantalla principal.

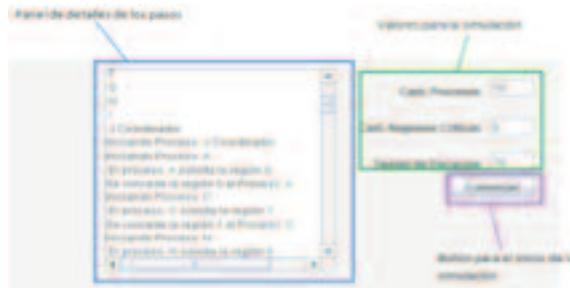


Figura 12: Resultados de la simulación.



Figura 13: Pantalla principal.



Figura 14: Menú vertical y carrusel de imágenes.

5.- Conclusiones

En primer lugar se hace notar que se han cumplido la totalidad de los objetivos planteados al inicio del presente trabajo.

Con las herramientas utilizadas se consiguió una aplicación amigable al usuario e intuitiva para ser utilizada como parte del trabajo de laboratorio de una asignatura universitaria de sistemas operativos.

Se ha comprobado la facilidad de uso del sistema con un grupo voluntario de alumnos, quienes han destacado su sencillez y amigabilidad.

Asimismo se ha verificado que los resultados obtenidos por los distintos componentes del aplicativo son totalmente coherentes con las previsiones teóricas aplicables en los distintos casos.

Como líneas futuras de trabajo se propone la incorporación de distintos sistemas expertos para optimizar la gestión de recursos computacionales, distintos casos de sincronización de procesos, sincronización de relojes en sistemas distribuidos, algoritmos de selección de procesos, etc.

Referencias bibliográficas

- Aggarwal, A. & Chandra, A. (1988). Virtual memory algorithms. STOC '88 Proceedings of the twentieth annual ACM symposium on Theory of computing Pages, 173-185, ACM New York, NY, USA.
- Deitel, H. M. (1993). Introducción los Sistema Operativos - 2E. Addison Wesley. Longman.
- Derntl, M., Hampel, T., Motschnig-Pitrik, R. & Pitner, T. (2011). Inclusive social tagging and its support in Web 2.0 services. *Computers in Human Behavior* 27(4): 1460-1466.
- Derntl, M. & Motschnig-Pitrik, R. (2005). The role of structure, patterns, and people in blended learning. *Internet and Higher Education* 8 (2005) 111-130 Elsevier Inc.
- Gagne, G. & Baer Galvin, P. (2006). Fundamentos de Sistemas Operativos – 7 / E. McGraw-Hill / Interamericana de España S.A.U.
- La Red Martínez, D. L. (2004). Sistemas Operativos. EUDENE-UNNE.
- Motschnig-Pitrik, R. & Holzinger, A. (2002). Student-centered teaching meets new media: concept and case study. *Journal of Educational Technology and Society*, 5(4), 160-172.
- Nichols, M. (2008). E-Primer Series – E-Learning in Context. Laidlaw College, Auckland, New Zeland.
- Papert, S. A. (1999). *Mindstorms* (2nd ed.). New York7 Basic Books.
- Stallings, W. (2005). Sistemas Operativos – 5 / E. Pearson Educación.

Tanenbaum, A. S. (2009). *Sistemas Operativos Modernos - 3E*. Pearson Educación.

Tanenbaum, A. S. & Van Steen, M. (2008). *Sistemas Distribuidos. Principios y Paradigmas - 2 / E*. Pearson Educación.

Vitter, J. S. (2008). *Algorithms and Data Structures for External Memory. Foundation and Trends in Theoretical Computer Science*. Publishers Inc. USA.