

Aprendizaje de sistemas operativos mediante simulaciones interactivas

Sonia Pamplona Roche
Departamento de Ciencias e Ingeniería

Universidad a Distancia de Madrid
Carretera de la Coruña
(Via de Servicio, nº 15), KM.38,500
28400 Collado Villalba (Madrid)
sonia.pamplona@udima.es

Nelson Medinilla Martínez
Departamento de Lenguajes y Sistemas
Informáticos e Ingeniería del Software
Universidad Politécnica de Madrid
Facultad de Informática
Campus de Montegancedo
28660 Boadilla del Monte (Madrid)
nelson@fi.upm.es

Resumen

Según los estudiantes, los profesores y la literatura, la asignatura de sistemas operativos es difícil. Tales dificultades se han abordado mediante la programación de partes de los sistemas operativos y el uso de simuladores. Ambos caminos son buenos, pero aún muestran inconvenientes. El presente trabajo está dirigido hacia el aprendizaje del funcionamiento de un sistema operativo mediante simulaciones interactivas. Éstas poseen dos cualidades diferenciadoras: una interacción sencilla con el estudiante y la visualización, paso a paso, del funcionamiento del sistema operativo. Su aplicación en la asignatura Sistemas Operativos del Grado en Ingeniería Informática, en modalidad *e-learning*, ha mostrado resultados positivos.

Summary

According to students, teachers and literature, the Operating Systems is a difficult academic subject. This difficulty has been addressed by programming parts of the SSOO and using educational simulators. Both methods are good, but they still have drawbacks. The present work is aimed at learning how an Operation System works, using interactive simulations. These simulations have two distinguishing qualities: an easy interaction with the student and the display of the performance of the Operating System, step by step. Its application on the Operating Systems Subject (e-learning mode) of the Computer Engineering Degree has shown positive results.

Palabras clave

Sistemas Operativos, simulaciones interactivas, aprendizaje, enseñanza, e-learning.

1. Introducción

Sistemas Operativos (SSOO) es una asignatura obligatoria en los planes de estudio de Ingeniería Informática y su objetivo es que los estudiantes adquieran los conocimientos acerca del funcionamiento interno de un sistema operativo.

Sin embargo, nuestra experiencia como docentes de la asignatura nos muestra que este objetivo es difícil de alcanzar y la literatura lo confirma.

Los estudiantes encuentran los conceptos y mecanismos de los sistemas operativos difíciles de aprender y relacionar. Tales problemas se han tratado de resolver mediante la programación de SSOO instruccionales [1] y el uso de simuladores [3][4][8], caminos adecuados, pero mejorables.

En la programación de partes de SSOO, los estudiantes deben dominar el lenguaje de programación del sistema operativo (normalmente C y C++), competencia no adquirida normalmente en el momento en el que se imparte la asignatura. El uso de los simuladores tampoco es sencillo, y puede ocurrir que requiera más tiempo de aprendizaje que el necesario para realizar la práctica de laboratorio [6].

El estudio presentado adopta el segundo camino, la simulación interactiva, pero con un simulador más simple que visualiza aspectos escondidos de los sistemas operativos.

La metodología de trabajo ha seguido un modelo cíclico del que se han realizado dos iteraciones obteniendo dos versiones del simulador. En la sección 2, se describen las dos versiones y los resultados de los estudios preliminares de cada una de ellas. Finalmente, presentamos las conclusiones y una propuesta para investigaciones futuras.

Solución propuesta

El diseño de las simulaciones se realizó teniendo en cuenta las características del Grado de Ingeniería Informática donde han sido evaluadas: un curso online, sin clases presenciales, con sesiones de estudio con una duración media de dos horas. En este contexto no es posible el uso de la programación de sistemas operativos ni el uso de los simuladores existentes, ya que los estudiantes podrían pasar más tiempo aprendiendo a usar los entornos de aprendizaje que resolviendo las actividades propuestas. Por este motivo, se decidió diseñar un material específico.

El estudio se ha realizado durante el primer semestre del curso 2011-2012 en la asignatura Sistemas Operativos, que cuenta con un total de 12 estudiantes

1. Primer simulador. Descripción.

El primer paso para el diseño del primer simulador fue precisar los objetivos de aprendizaje que se pretenden facilitar.

Estos objetivos, extraídos del informe sobre Computing Curricula propuesto por ACM and IEEE Computer Society [7], son los siguientes: a) describir la necesidad de concurrencia en un sistema operativo, b) explicar los diferentes estados por los que pasa una tarea y las estructuras de datos necesarias para soportar la gestión de varias tareas y c) describir razones para usar interrupciones y cambio de contexto para soportar la concurrencia en un sistema operativo.

Para facilitar la consecución de los objetivos citados, el simulador muestra las acciones que realiza el sistema operativo para que el procesador ejecute los programas de usuario, las llamadas al sistema y las rutinas de tratamiento de interrupción.

En concreto, el simulador permite visualizar el efecto de los siguientes eventos del sistema operativo sobre el procesador y la memoria:

arranque y parada del sistema operativo, ejecución y finalización de un programa, inicio y fin de una llamada al sistema, e inicio y fin de una interrupción.

Aunque las tecnologías de visualización resultan atractivas para estudiantes y educadores, su eficacia para el aprendizaje no está asegurada en todos los casos y está relacionada con la implicación del estudiante en el uso de la simulación [9]. Por este motivo se creó una taxonomía que permite clasificar distintos niveles de implicación en el estudiante [5].

Nuestro simulador está situado en el nivel 2, *responding*, de esta taxonomía, ya que se proporcionó a los estudiantes una serie de preguntas que debían responder usando la simulación.

2. Primer simulador. Resultados.

Después del uso del primer simulador, los estudiantes lo evaluaron a través la siguiente encuesta de satisfacción:

¿Te ha ayudado la simulación a comprender mejor los siguientes conceptos? (Escala Likert de 1 a 5, en la que el 1 significa “Me ha ayudado muy poco” y el 5 significa “Me ha ayudado mucho”).

Proceso, llamada al sistema, interrupción, anidamiento de interrupciones, núcleo expulsable, sistema operativo.

¿Estás interesado en qué se desarrollen más simulaciones que ayuden a comprender mejor otros conceptos de esta asignatura?

Los resultados de la primera pregunta se muestran en la tabla 1. Todos los estudiantes contestaron de forma positiva a la pregunta 4, confirmando así su interés por el desarrollo de más simulaciones.

Concepto	Media	Varianza
Proceso	4,29	0,95
Llamada al sistema	4,14	0,90
Interrupción	4,57	0,79
Anidamiento de interrupciones	4,71	0,49
Núcleo expulsable	4,71	0,49
Sistema Operativo	4,43	0,79

Tabla 1. Comprensión de conceptos

El apartado de observaciones muestra un descubrimiento clave de uno de los estudiantes: “No me había dado cuenta de algo tan fundamental como que el sistema operativo deja de estar en el procesador cuando se ejecuta un

programa. En el minuto uno de la explicación me ha quedado claro. Visual y muy comprensible.”

Como consecuencia de estos resultados positivos, decidimos realizar un segundo simulador.

Segundo simulador. Descripción.

La diferencia de este segundo simulador respecto del primero es que presenta más detalles y los mecanismos están menos simplificados.

El primer simulador mostraba la secuencia de ejecución de programas, llamadas al sistema y rutinas de tratamiento de la interrupción en distintos casos, pero dejaba ocultos los mecanismos necesarios para pasar de un tipo de código a otro. En la segunda versión se hizo explícito este mecanismo para que los estudiantes pudieran justificar la existencia de las estructuras de datos necesarias para la gestión de procesos en un sistema operativo.

Segundo simulador. Resultados.

Respecto a este segundo simulador, los estudiantes contestaron dos bloques de preguntas.

El primer bloque motivaba al estudiante a una exploración completa del simulador y, al mismo tiempo, permitía comprobar si éste había identificado los pasos que se muestran en el simulador. Las respuestas de los estudiantes demostraron que los mecanismos ocultos del sistema operativo quedaron explícitos y claros, y se pudieron conocer a través de la interacción con la simulación.

El siguiente bloque de preguntas tuvo como objetivo evaluar la comprensión de los mecanismos presentados por el simulador. La primera pregunta comprobaba si la interacción con el simulador eliminaba la falsa creencia de que el procesador puede ejecutar al mismo tiempo el sistema operativo y un programa. Esta pregunta se ubica en el segundo nivel de la taxonomía cognitiva de Bloom [2], *comprensión*, ya que supone una reestructuración de las ideas y un cálculo de la relativa importancia y las relaciones recíprocas de dichas ideas.

¿Qué mecanismo permite la ejecución de una llamada al sistema durante la ejecución del programa?

- a. La modificación del contenido del registro contador de programa.

- b. La coexistencia en el procesador del código del sistema operativo y del programa.

- c. Las otras dos respuestas son ciertas.

Un 50% de los estudiantes contestó a la pregunta de forma correcta con la opción a, y el otro 50% contestó de forma incorrecta con la opción c.

La segunda pregunta tenía como objetivo aplicar lo aprendido en un sistema operativo multitarea. Este objetivo se corresponde con el tercer nivel de la taxonomía de Bloom, *aplicación*, ya que el estudiante debe seleccionar una abstracción en una situación en la que no se ha especificado ningún modo de solución.

El sistema operativo de la simulación es monotarea. Si se quisiera diseñar un sistema operativo multitarea. ¿Existiría alguna diferencia con respecto a la estructura de datos en la que se almacenan las direcciones de retorno? ¿Qué tipo de estructura necesitarías? Explica tu propuesta.

Los resultados de esta pregunta coincidieron con los de la pregunta anterior, que fue contestada de forma correcta por un 50% de estudiantes y como incorrecta por el otro 50%. Además, los estudiantes que contestaron de forma correcta e incorrecta son los mismos que los de la primera pregunta. Por tanto, los resultados son congruentes con el tipo de preguntas, ya que en la taxonomía propuesta por Bloom, para alcanzar un objetivo de aplicación, es necesario haber alcanzado previamente la comprensión.

En resumen, los estudiantes han reconocido los pasos que se visualizan en la simulación, pero la mitad de ellos no ha comprendido el motivo de los pasos, ni puede aplicar lo aprendido a otra situación.

Una posible explicación de estos resultados es que la interacción con la simulación no ha sido suficiente para apreciar que la complejidad está determinada por las limitaciones del procesador. Para hacer al estudiante consciente de este hecho, la próxima versión del simulador podría permitirle construir por sí mismo los pasos de la simulación. En este caso, el simulador tendría que presentar el problema a resolver (la ejecución de varios tipos de código: programa y llamadas al sistema) y las restricciones (comportamiento del procesador: incremento del registro contador de programa, ejecución de la instrucción y carga de la nueva instrucción desde la memoria).

Finalmente, los estudiantes evaluaron la segunda versión del simulador con preguntas similares a las de la primera versión. Los resultados de la comprensión de conceptos se pueden observar en la tabla 2. La media obtenida es ligeramente inferior que la del primer simulador. Estos resultados son coherentes con las preguntas de comprensión, que han tenido un 50% de aciertos.

Concepto	Media	Varianza
Proceso	4	0,76
Llamada al sistema	4	0,93
Sistema Operativo	3,87	0,99

Tabla 2. Comprensión de conceptos

Son especialmente interesantes las respuestas a la pregunta: ¿tienes ahora más claro algún aspecto de la asignatura que no hubieras entendido antes? ¿cuál? Las respuestas fueron positivas, sólo el 25% estudiantes contestó que no. Algunas respuestas relevantes son: “Sí, el funcionamiento de las llamadas a sistema desde una aplicación.”, “Tras la revisión de la animación, he aprendido que en memoria no sólo se guardan direcciones de instrucción, también se salva el estado del proceso en ejecución.”, “La forma en que el procesador controla el flujo de ejecución del sistema operativo, programas e interrupciones por medio del contador de programas.”, “Si, en principio todo los involucrados, especialmente interesante es la visualización de como el Contador de Programa está presente en todo momento, así como, dejar constancia de las rutinas de inicio del sistema”.

Conclusiones

Este trabajo presenta los primeros pasos de desarrollo de dos simuladores para facilitar el aprendizaje de la asignatura Sistemas Operativos.

Los resultados de las encuestas de satisfacción y las reflexiones de los estudiantes sobre su aprendizaje sugieren que los simuladores pueden ser un recurso educativo efectivo.

Sin embargo, se ha comprobado que los estudiantes todavía tienen dificultades con la comprensión de los conceptos. Es necesario dar un paso más creando un simulador que permita al estudiante construir y validar sus propias

soluciones, haciendo que su implicación sea mayor. También es necesario realizar estudios más rigurosos para confirmar y consolidar los resultados de este estudio preliminar.

Los simuladores están disponibles en los siguientes enlaces: <http://goo.gl/GKv2t> y <http://goo.gl/3kXpH>.

Referencias

- C. L. Anderson and M. Nguyen, “A survey of contemporary instructional operating systems for use in undergraduate courses,” *Journal of Computing Sciences in Colleges*, vol. 21, no. 1, pp. 183-190, Oct. 2005.
- B. S. Bloom and D. R. Krathwohl, *Taxonomy of Educational Objectives: the Classification of Educational Goals, Handbook I: Cognitive Domain*. Addison-Wesley, 1956.
- D. Jones and A. Newman, “RCOS . JAVA : A SIMULATED OPERATING SYSTEM WITH ANIMATIONS,” in *Proceedings of the Computer-Based Learning in Science Conference*. Rep. Tcheca, 2001.
- L. P. Maia and A. C. Pacheco Jr., “A simulator supporting lectures on operating systems,” in *Frontiers in Education 2003 FIE 2003 33rd Annual*, 2003, vol. 2, pp. F2C-13-17 Vol.2.
- T. L. Naps et al., “Exploring the role of visualization and engagement in computer science education,” *ACM SIGCSE Bulletin*, vol. 35, no. 2, p. 131, Jun. 2003.
- J. E. Pérez, J. García Martín, and I. Muñoz Fernández, “Cooperative learning in operating systems laboratory,” *ACM SIGCSE Bulletin*, vol. 40, no. 3, p. 323, Aug. 2008.
- The Joint Task Force on Computing Curricula, IEEE Computer Society, and Association for Computing Machinery, “Computer Science Curriculum 2008 : An Interim Revision of CS 2001 Report from the Interim Review Task December 2008”, 2008.
- S. Robbins, “An address translation simulator,” *ACM SIGCSE Bulletin*, vol. 37, no. 1, p. 515, Feb. 2005.
- C. Shaffer et al., “Algorithm Visualization: The State of the Field,” *ACM Transactions on Computing Education TOCE*, vol. 10, no. 3, pp. 1-22, 2010.