

Simuladores de Planificadores de Sistemas en Tiempo Real

Francisco J. Aliaga García, Isabel M. Aliaga García,
Joaquín Olivares Bueno¹, Juan C. Gámez Granados¹, José M. Palomares Muñoz¹

¹ Dpto. de Arquitectura de Computadores, Electrónica y Tecnología Electrónica de la
Universidad de Córdoba
Córdoba, España
{i72algaf, i52algai, olivares, jcgamez, jmpalomares}@uco.es

Resumen. En este artículo se presenta un simulador desarrollado que permite ejecutar diferentes planificadores de Tiempo Real, como el algoritmo de planificación cíclica, Algoritmo de la Razón Monótona (RMA) y EDF (Earliest Deadline First) para un conjunto de procesos con unos datos dados y muestra los resultados obtenidos. Mediante este simulador se facilita a los alumnos el aprendizaje de los algoritmos de planificación.

Palabras Clave: Planificadores, simuladores, sistemas en tiempo real

Abstract. This paper presents a simulator that has been developed to allow the execution of scheduling algorithms such as the Cyclic Non-preemptive Executive, Rate-monotonic scheduling (RMS) and Earliest Deadline First (EDF) for a given set of processes with different values and the simulator displays the results. With this simulator, students are able to learn about scheduling algorithms.

Keywords: Schedulers, simulator, Real-Time systems, algorithms

1 Introducción

Existen múltiples aplicaciones y sistemas que necesitan obtener una respuesta antes de un determinado tiempo límite (denominado comúnmente, *deadline*). A dichos sistemas se les denomina Sistemas con restricciones de Tiempo Real o simplemente Sistemas en Tiempo Real. En estos sistemas se admite que el resultado numérico no sea el óptimo, siempre que el resultado sea suficientemente aceptable y se proporcione a tiempo.

En general, estos sistemas se dividen en un conjunto de tareas, cada una de ellas encargada de dar respuesta a variaciones en el entorno o en el propio estado lógico del sistema. Existe un límite en el número de tareas que se puede ejecutar simultáneamente (dicho límite es igual al número de procesadores o núcleos de procesamiento existentes en el sistema). Cuando se supera dicho máximo ya no es posible la ejecución simultánea real y por tanto, es necesario organizar de alguna manera las tareas, para que se pueda establecer el orden de ejecución de cada una de ellas en cada momento entre los diferentes procesadores. Las reglas que permiten hacer dicho ordenamiento constituyen los algoritmos de planificación.

El sistema operativo y, en particular, el planificador, es quizá el componente más importante de un Sistema de Tiempo Real [1]. Los algoritmos de planificación establecen prioridades a las tareas que permiten al sistema operativo escoger las tareas más relevantes en cada instante. Dichas prioridades pueden ser fijas y no variar a lo largo de la ejecución, o pueden ir variando conforme se van ejecutando las tareas.

Dada la importancia del procesamiento en tiempo real, es materia de estudio en algunas asignaturas. En particular, se puede destacar la asignatura Sistemas en Tiempo Real, perteneciente a los planes de estudios de la titulación de Ingeniería en Informática, Ingeniería en Automática e Ingeniería Industrial y Graduado en Ingeniería Informática de la Universidad de Córdoba (España).

El resto del artículo está estructurado como sigue. En el apartado 2 se hace una breve introducción a la planificación en tiempo real. Continúa el apartado 3 mostrando otros planificadores que podemos encontrar en la literatura. Seguidamente, el apartado 4 nos muestra la aplicación desarrollada junto con sus principales características para finalmente mostrar las conclusiones del trabajo en el apartado 5.

2 Planificación de Tiempo Real

En el momento en que hay más de dos tareas que se ejecutan en un determinado sistema hay que plantear algún mecanismo para poder intercambiar las tareas. El planificador es un módulo del S.O. que se encarga de controlar las tareas existentes en el sistema y proporcionarles tiempo de CPU en un determinado instante, gestionando el estado de ejecución en el que se encuentra y decidiendo en cada caso qué tarea debe ejecutarse en cada momento [2].

La planificación en un sistema operativo de propósito general intenta garantizar un uso extensivo de la CPU, equidad en la distribución de acceso al tiempo de ejecución y bajo *overhead*. La planificación de un conjunto de tareas es viable si todas las tareas comienzan su ejecución después de su instante de lanzamiento y todas terminan antes de su *deadline*. La planificación de las tareas tiene dos mecanismos principales:

- **Planificación cíclica no apropiativa:** Las tareas se ejecutan repetitivamente con un determinado periodo. El sistema operativo permite ejecutar cada tarea sin interrupción por parte de las otras tareas hasta que finalice, tras lo cual se ejecuta otra tarea.
- **Planificación apropiativa:** El sistema operativo asigna la CPU a una tarea hasta que la interrumpe y le proporciona la CPU a otra tarea distinta. Se dice que es apropiativa (o expropiativa) porque el S.O. se apropia de la CPU, que estaba en posesión de una tarea. La asignación de una tarea a una CPU puede realizarse por un cierto tiempo conocido a priori o hasta que suceda un bloqueo o llamada al sistema operativo.

La asignación de prioridades a las tareas se puede realizar con planificaciones apropiativas y con no apropiativas, aunque lo más habitual es que se realice con planificaciones apropiativas. Las prioridades pueden asignarse de manera estática (Algoritmo de la Razón Monótona, RMA) o de manera dinámica (EDF).

2.1 Planificación cíclica no apropiativa

La planificación cíclica sin prioridad es un mecanismo de planificación sencillo de implementar y calculable a priori. Requiere un conocimiento muy concreto de los tiempos de ejecución de las rutinas y es muy eficiente en términos de bajo *overhead* pero no maneja bien eventos esporádicos ni interrupciones. Su desventaja es que es muy restrictivo, ya que todas las tareas deben tener tiempos de ejecución similares.

Esta planificación se utiliza principalmente en pequeños sistemas empujados donde se conocen completamente las tareas que requiere el sistema. Para desarrollar un planificador basta con diseñar un bucle que vaya ejecutando una tras otra las tareas.

Hay que definir un ciclo mayor, que coincide con el periodo de repetición de toda la planificación (y es el mínimo común múltiplo de todos los periodos) y un ciclo menor, que es el máximo común divisor de los periodos de todas las tareas.

2.2 RMA (Algoritmo de la Razón Monótona)

El Algoritmo de la Razón Monótona (RMA) es uno de los algoritmos de asignación de prioridad más estudiados y es el más utilizado en la práctica. En RMA todas las tareas son periódicas y el *deadline* (relativo, es decir, en cada iteración de la tarea) de cada tarea es igual a su periodo. Las tareas de mayor prioridad pueden interrumpir a las tareas de menor prioridad.

Este algoritmo asigna prioridades (P_i) a las tareas inversamente proporcionales a su periodo (T_i). La prioridad de cada tarea se calcula de la siguiente forma:

$$\text{Prioridad}_i = \frac{100}{T_i}$$

El esquema de prioridades de RMA es óptimo, en tanto que si un conjunto de tareas es planificable mediante cualquier mecanismo de prioridades fijas, entonces lo será mediante RMA.

Test de planificabilidad

El algoritmo de la Razón Monótona posee un mecanismo matemático que permite determinar si un conjunto de tareas es viable utilizando RMA. En 1973, Liu y Layland demostraron que un conjunto de tareas es planificable utilizando RMA si:

Para N procesos, se cumple que:

$$\sum_{i=1}^N \left(\frac{C_i}{T_i} \right) < N \cdot \left(2^{\frac{1}{N}} - 1 \right)$$

Donde C_i es el tiempo de ejecución en el peor caso del proceso i .

T_i es el periodo de repetición del proceso (llamado periodo de lanzamiento).

La sumatoria representa el grado de utilización total de la CPU por el conjunto de procesos.

Este test es solo una condición suficiente y no necesaria. Si se cumple el test, entonces se cumplirán todos los *deadline*. Si el test falla, los *deadline* podrían cumplirse o no, por lo que se deberá hacer el gráfico de *timeline* para comprobar el comportamiento de las tareas y observar si son planificables o no.

2.3 EDF (Earliest Deadline First)

El Algoritmo EDF (Earliest Deadline First) es un algoritmo de asignación dinámica de prioridades. Las prioridades de las tareas van modificándose, proporcionando mayor prioridad a las tareas que tienen más cercano su *deadline*. EDF es un algoritmo óptimo en uniprosesores: si un conjunto de tareas no es viable con EDF, no existe ningún otro algoritmo de planificación que lo haga viable.

Con EDF no es necesario realizar la suposición de que las tareas son periódicas. El resto de suposiciones realizadas para RMA se toman para EDF.

La ventaja de las prioridades dinámicas es que aumentan la flexibilidad, permiten aumentar la prioridad de una tarea en una situación de alerta. Sin embargo, también

tiene algunas desventajas como que el cambio dinámico de prioridades es peligroso porque el comportamiento del sistema es mucho más complejo de predecir y, además, existe el riesgo de bloquear algunas tareas durante demasiado tiempo. [3]

El test de planificación para EDF cuando todas las tareas son periódicas y sus *deadlines* coinciden con sus periodos, es simple: Si la utilización total del conjunto de tareas es menor que 1, el conjunto de tareas se puede planificar en un procesador usando EDF. Sin embargo, si no se cumplen que las tareas sean periódicas o que los *deadlines* no coinciden con los periodos, no existe ningún mecanismo para comprobar la planificabilidad: hay que ejecutarlo y comprobarlo experimentalmente.

3 Antecedentes

Existen diversas herramientas de apoyo al análisis de planificabilidad en este tipo de sistemas, destinadas a usuarios profesionales. Las hay tanto comerciales como de libre distribución. A continuación se hará un breve resumen de las principales:

PerfoRMAx: Esta herramienta facilita tanto el análisis de planificabilidad como la simulación del comportamiento del planificador. El sistema se especifica de manera tabular y se emplea el modelo de eventos/acciones/recursos para describir las tareas. Estas se especifican mediante su prioridad, tiempo de ejecución, periodo y los recursos compartidos que requieren.

Rapid RMA [4]: Conocida originalmente como PERTS, proporciona uno de los entornos de análisis de planificabilidad más completos. Incluye técnicas que admiten políticas del tipo Deadline Monotonic Analysis (DMA), Earliest Deadline First (EDF) o ejecutivos cíclicos, permite el análisis End-to-End monoprocesador y distribuido, Off-line scheduling para CORBA de tiempo real y da soporte para el protocolo de acceso a recursos comunes orientado a objetos DASPCP.

TimeWiz: Es una herramienta bastante completa basada en el modelo de eventos/acciones/recursos. Admite entrada de datos tabular y parcialmente gráfica y dispone de una variedad de protocolos para el acceso mutuamente exclusivo a recursos compartidos. Proporciona un diagrama de eventos de salida sobre una línea de tiempos basada en eventos de entrada o en los nodos del sistema.

TIMES: Se presenta como una herramienta para el análisis simbólico de planificabilidad y la generación de código ejecutable para sistemas de tiempo real. Se basa en la utilización de un autómata temporizado de comportamiento temporal predecible como bloque básico para la construcción de sistemas de tiempo real. Permite generar el código C ejecutable para el sistema.

MAST (Modeling and Analysis Suite for Real-Time Applications): Se trata de un conjunto de herramientas de libre distribución y código abierto, que facilitan análisis de planificabilidad mono-procesador y distribuido, cálculo de holguras, asignación óptima de prioridades, simulación de eventos discretos, etc.

En general, todas las herramientas presentan problemas para su utilización dentro del ámbito docente. Son complicadas de usar, no están adaptados para la comprensión de los alumnos, los entornos de trabajo son complejos, muy pesados computacionalmente y no están adaptados a su uso remoto. Muchos de ellos son de pago y por tanto, no están disponibles de manera gratuita a los alumnos, lo que limita en gran manera su uso docente.

Para un uso educacional es necesario que la herramienta sea sencilla de utilizar, gratuita, con pocos requerimientos computacionales, con posibilidad de uso remoto y adaptada a los contenidos de los cursos de Tiempo Real.

4 Simulador

En esta sección se presenta el simulador de planificadores de Sistemas en Tiempo Real que se ha desarrollado, que tiene como principal objetivo facilitar el aprendizaje de los principales algoritmos de planificación. Mediante este simulador, el alumno puede ejecutar el algoritmo de planificación cíclica no apropiativa, RMA y EDF, así como crear conjuntos de procesos, editar sus datos y observar los resultados obtenidos tras aplicar los distintos algoritmos. También tiene la posibilidad de importar y guardar en sus propios archivos los datos relativos a los procesos y dispone de una opción para exportar los resultados obtenidos para un análisis posterior de los mismos.

4.1 Características

Mediante el simulador se modelará el comportamiento parametrizado de planificadores tanto apropiativos como no apropiativos habituales en Sistemas en Tiempo Real. Consta de una aplicación web [5] y una aplicación móvil bajo plataforma Android [6]. Sus principales características son las siguientes:

- Su funcionamiento no presenta dificultad en sí mismo, ya que persigue la máxima simplicidad y facilidad de uso. La interfaz es accesible y amigable.
- Ofrece la posibilidad de crear conjuntos de procesos y realizar operaciones con sus datos, como su exportación e importación y, además, permite exportar los resultados obtenidos tras ejecutar un algoritmo (esta opción no está disponible para Android).
- Realiza el cálculo de los tests de planificabilidad.

- Para el algoritmo EDF incluye una opción que permite visualizar las prioridades de los procesos en cada instante de tiempo.
- Permite visualizar el contenido de la página web en dos idiomas: inglés y español.
- El programa es fácilmente ampliable, por lo que se pueden introducir nuevos planificadores en el futuro.
- Permite consultar contenido teórico acerca de los algoritmos de planificación y ejemplos ilustrativos de cada uno de ellos.
- La aplicación web no requiere conexión a Internet, ya que se ha codificado en Javascript como aplicación con ejecución local.

4.2 Ejemplo de ejecución

En este ejemplo se muestran los resultados que se han obtenido al ejecutar el algoritmo de planificación RMA en el simulador. En la Tabla 1 se presentan los datos que se han considerado para los procesos.

Tabla 1. Datos de los procesos

Proceso	Inicio	Periodo (T)	T. ejecución (C)	Prioridad (P)	Utilización (U=C/T)
A	0	5	1	20	0,2
B	0	10	3	10	0,3
C	0	15	2	6,67	0,13
D	0	10	2	10	0,2
Utilización global de la CPU					0,833

Test de planificabilidad

Se ha obtenido que el grado de utilización global de la CPU por el conjunto de procesos es de 0,833, mientras que $N(2^{1/N} - 1)$ es $4(2^{1/4} - 1) = 0,757$; por lo que no se cumple el test de planificabilidad. Si el test falla, los *deadline* podrían cumplirse o no. En este caso se debe realizar el gráfico de *timeline* para comprobar el comportamiento de las tareas y ver si son planificables o no.

En la Figura 1 se muestra la interfaz de la aplicación web, que es muy similar a la de la aplicación móvil. En la parte superior se sitúa la barra de menús, que permite acceder a las diferentes secciones y cambiar el idioma, y debajo se localiza el área central de la página, donde aparece el contenido de la sección a la que se haya accedido.

El simulador es la página de inicio y la parte más importante de la aplicación web, donde el usuario puede interactuar con el sistema y obtener unos resultados a partir de los datos que introduce. En primer lugar, se selecciona el algoritmo que se desea ejecutar. Una vez elegido, se desplegará debajo el segundo paso, en el que se mostrará

una tabla con los datos de los procesos. Debajo de la tabla se incluyen distintos botones para importar o exportar los datos de los procesos, así como borrar toda la tabla y ejecutar el algoritmo seleccionado.

The screenshot shows the web simulator interface with the following elements:

- Navigation menu: Planificadores STR, Simulador (selected), Algoritmos, Acerca de, Español, English.
- Step 1: "1. Selecciona el algoritmo" with buttons for "Cíclico", "RMA" (selected), and "EDF".
- Step 2: "2. Completa la tabla con los datos de los procesos" with a table for process data.

Proceso	Inicio	Periodo	Tiempo de ejecución
A	<input type="text" value="0"/>	<input type="text" value="5"/>	<input type="text" value="1"/>
B	<input type="text" value="0"/>	<input type="text" value="10"/>	<input type="text" value="3"/>
C	<input type="text" value="0"/>	<input type="text" value="15"/>	<input type="text" value="2"/>
D	<input type="text" value="0"/>	<input type="text" value="10"/>	<input type="text" value="2"/>

Below the table, there is an "Añadir" button and three action buttons: "Importar datos", "Exportar datos", and "Borrar todo". At the bottom, there is a large green "Ejecutar algoritmo" button.

Figura 1. Interfaz del simulador de la aplicación web.

En la Figura 2 se muestra el tercer paso, que aparece al ejecutar el algoritmo. Para RMA se obtiene una tabla con la prioridad de cada proceso y el grado de utilización de la CPU, la explicación teórica y el resultado del test de planificabilidad, además de la línea temporal obtenida. También es posible exportar los resultados de la ejecución.

Para este ejemplo, el test de planificabilidad no se cumple, por lo que el *deadline* de los procesos podría cumplirse o no, pero al realizar el gráfico de *timeline* se puede ver que el conjunto de procesos es planificable y así se indica en el resultado.

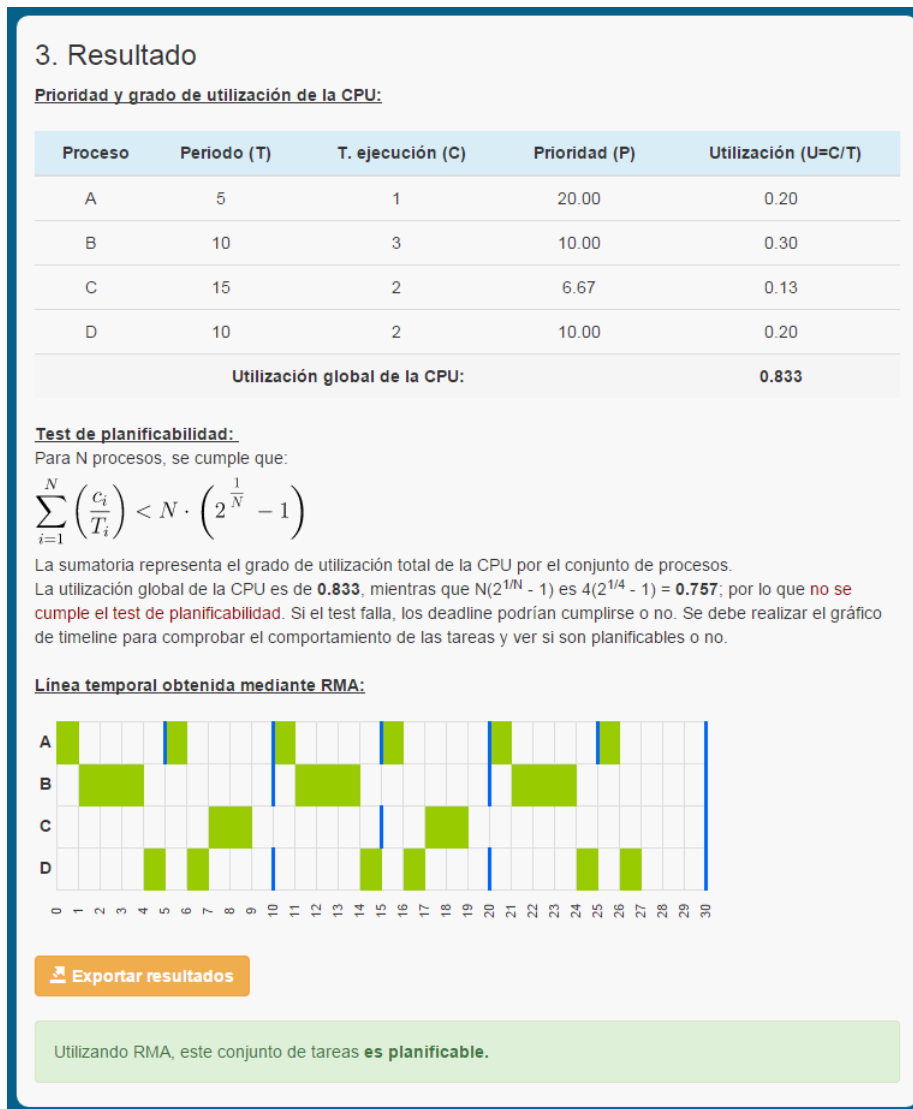


Figura 2. Visualización de resultados al ejecutar un algoritmo.

5 Conclusiones

En este artículo se han descrito las características novedosas que presenta el simulador desarrollado, que lo diferencian de otras herramientas que se pueden encontrar en el mercado. Esta herramienta es idónea para ser utilizada en el ámbito

académico, ya que facilita al profesor su labor docente, los alumnos pueden usarla para corregir sus propios ejercicios y los resultados obtenidos se pueden interpretar fácilmente.

La aplicación web y la aplicación Android son similares en cuanto a resultados aunque tienen pequeñas variaciones. La versión web, al estar codificada como una aplicación con ejecución local mediante lenguaje Javascript, permite su ejecución en múltiples entornos sin necesidad de conexión con Internet.

La exportación de la simulación permite un análisis más pormenorizado de los resultados utilizando otros mecanismos externos (paquetes estadísticos, hojas de cálculo, etc.) que consiguen realizar el cálculo de diferentes parámetros de interés, como puede ser el *jitter* de las tareas entre repeticiones, holgura de los periodos para interrupciones o eventos, etc.

Dentro de la línea seguida en el desarrollo, se puede contemplar la introducción de nuevos planificadores en el futuro, ya que se ha tenido en cuenta que la actualización y el mantenimiento del software no resulte difícil. También se ha estimado interesante la creación de componentes adicionales que permitan analizar los resultados obtenidos, así como incluir distintos estilos que modifiquen mediante CSS el diseño de la web para mejorar su accesibilidad.

Referencias

1. Stallings, W. Sistemas Operativos. Traducido por Dodero Beardo, J.; Torres Franco, E; Katrib Mora, M. Revisión Técnica de Joyanes Aguilar, L. 2ª ed. Madrid. Prentice-Hall. 1997. 732 p. ISBN: 84-89660-22-0.
2. Palomares Muñoz, J.M. Tema 6: Planificación de Tiempo Real. Sistemas en Tiempo Real. Universidad de Córdoba.
3. Aldea Rivas, M. Planificación de Tareas en Sistemas Operativos de Tiempo Real Estricto para Aplicaciones Empotradas. Universidad de Cantabria [en línea]. [Última consulta: 4 de abril 2015]. Disponible en: <http://marte.unican.es/documentation/tesis-mario.pdf>
4. Tri-Pacific Software Inc. Rapid RMA [sitio web]. [Última consulta: 4 de abril 2015]. Disponible en: <http://www.tripac.com/rapid-rma>
5. Aliaga García, F.J.; Aliaga García, I.M.; Palomares Muñoz, J.M.; Gámez Granados, J.C. Simulador de Planificadores de Sistemas en Tiempo Real [sitio web]. [Última consulta: 6 de abril 2015]. Disponible en: <http://www.uco.es/~el2pamuj/simuladorSTR/>
6. Aliaga García, F.J.; Aliaga García, I.M.; Palomares Muñoz, J.M.; Gámez Granados, J.C. App Android de Simulador de Planificadores de Sistemas en Tiempo Real [App Android]. [Última consulta: 6 de abril 2015]. Disponible en: <http://www.uco.es/~el2pamuj/simuladorSTR/SimuladorSTR.apk>