

# Proyecto 2: Calendarización en Tiempo Real

## I. DESCRIPCIÓN

El propósito de este proyecto es simular el comportamiento de varios algoritmos de *scheduling* clásicos para Sistemas Operativos de Tiempo Real (RTOS). Se tendrá una interaz gráfica hecha con GTK y se generará una presentación Beamer como salida. Toda la programación debe realizarse en C sobre Linux.

## II. ALGORITMOS

Deben investigar e implementar los siguientes algoritmos de *scheduling* para tareas periódicas en un RTOS:

- **RM:** *Rate Monothonic*
- **EDF:** *Earliest Deadline First*
- **LLF:** *Least Laxity First*

## III. INTERACCIÓN

Usando una interfaz gráfica GTK el usuario indicará cuántas tareas se van a ejecutar. Para cada una de ellas se ingresará el tiempo de ejecución ( $e_i$ ) y su período ( $p_i$ ). Nótese que  $p_i \geq e_i \forall i$ . La interfaz no aceptará caracteres inválidos. La cantidad máxima de procesos debe ser tal que la salida sea agradable y legible. Mínimo se espera que esta cantidad esté entre 1 y 6 procesos.

Este conjunto de tareas podría ser “*schedulable*” o no. Se deben aplicar y reportar los resultados de los tests descritos en [1] y [2]. Aún cuando estos tests den resultado negativo, siempre se simulará la ejecución de cada algoritmo.

Con la interfaz gráfica, el usuario indicará cual mezcla de los tres algoritmos disponibles desea ejecutar con el conjunto de tareas ingresado. También indicará si desea que cada *slide* de la salida muestre el estado de todos los algoritmos seleccionados por el mismo punto de ejecución o si desea que cada uno aparezca en un *slide* independiente.

## IV. SALIDA

Para generar una presentación Beamer, su programa debe primero preparar un archivo “.tex” que será compilado con pdflatex o algún equivalente para que se produzca un archivo “.pdf” que debe ser desplegado usando evince o algún equivalente. La invocación de todos estos comandos debe realizarse internamente desde su programa, de manera transparente para el usuario. Deben quedar disponibles tanto el pdf como el fuente de L<sup>A</sup>T<sub>E</sub>X. Se espera que esta presentación sea de mucha calidad.

En su presentación deben aparecer los siguientes elementos:

1. **Portada:** se identifica claramente a los miembros del grupo, el curso, el semestre y el nombre del proyecto.
2. **Algoritmos:** por cada uno de los algoritmos seleccionados se debe generar un *slide* con una explicación general del algoritmo.

3. **Tests de *schedulability*:** se deben mostrar las fórmulas, el resultado particular y la interpretación correspondiente.

4. **Ejecución:** se presentarán tablas de tiempo similares a las usadas en clases donde se vea el avance del tiempo y cuál tarea está corriendo en cada momento después de haber sido seleccionada por el algoritmo de *scheduling* particular. Los casos de violación de *deadline* deben ser indicados de la mejor forma posible.

Dado un conjunto de tareas se calcula el mínimo común múltiplo de todos los períodos. Se debe simular la ejecución de las tareas durante esa cantidad de tiempo para garantizar que se regresa a la situación inicial. Cada grupo decidirá cuántas unidades de tiempo mostrará en cada *slide* usando como criterio la legibilidad pero al mismo tiempo el mínimo posible de *slides*.

Siguiendo las instrucciones del usuario cada *slide* podría mostrar el estado de todos los algoritmos solicitados, o se completa cada algoritmo de manera independiente. Se sigue prefiriendo la mínima cantidad posible de *slides*. Por supuesto, se espera el uso de colores y gráficos agradables, con toda la información adecuadamente etiquetada.

## V. FECHA DE ENTREGA

Demostraciones en **semana 11** en clase. Debe subir su código fuente al Tec-Digital antes de las 6 am el día de la presentación. Coloque todo lo necesario para compilar, documentar y ejecutar su proyecto en un directorio cuyo nombre esté formado por los apellidos de los miembros de cada grupo separados por guiones. Compacte este directorio en la forma de un .zip llamado igual que el directorio (e.g., rivera-alvarado-alfaro.tgz) y envíelo por correo.

## REFERENCIAS

- [1] C. L. Liu, J. Layland, “Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment”, Journal of the ACM, Vol. 20, No. 1, January 1973.
- [2] E. Bini, G. Buttazzo, G. Butazzo, “Rate Monotonic Analysis: the Hyperbolic Bound”, IEEE Trans. Computers, Vol. 52, 2003.