Sistemas Operativos
Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico 1: Scheduling

Adrian Vinocur	adrian.vinocur@gmail.com
Alejandro Albertini	ale.dc@hotmail.com
Raul Brum	brumraul@gmail.com

Resumen

En este trabajo estudiaremos el problema de planificar (scheduling) la ejecución de tareas. Analizaremos de manera experimental algunos de los algoritmos clasicos como FCFS, Round Robin. Además analizaremos otros que utilizan prioridades para resolver el problema de planificación en tiempo real como RM (Rate Monotonic) y EDF (Earlier Deadline First).

Palabras Claves: Scheluder, Scheduling, FCFS, Round Robin, RM, Rate Monotonic, EDF, Earlier Deadline First, Quantum

Índice

1. Introducción

En el presente trabajo nos proponemos a implementar la lógica de de varios planificadores de tareas.

Para implementar los algoritmos utilizaremos el simulador provisto por la catedra, nuestro trabajo consistira en implementar ciertos algoritmos de scheduler en lenguaje c++ y realizar experimentos empíricos tratando de encontrar en particular para Round Robin el Quantum que nos de mejores resultados en las pruebas tratando de satisfacer métricas que nos son compatibles entre sí (Es decir encontrar un punto de equilibrio entre ellas).

Además analizaremos un paper sobre scheduling basados en prioridades para procesos en tiempo real con ciertos requerimientos de deadlines obligatorios. Implementaremos estos últimos y buscaremos ejemplos para ilustrar el comportamiento de estos.

La trabajo se encuentra dividido en 10 ejercicios cuya meta de estos es abarcar lo anteriormente comentado. Se decidio omitir una sesión resultados agregando los gráficos directamente en la sesión desarrollo, lo que se busca con esta estructuración del informe es mantener cada ejercicio con explicaciones y graficos juntos para comodidad del lector.

2. Desarrollo

2.1. Ejercicio 1

TODO: AGREGAR ALGO

2.2. Ejercicio 2

TODO: AGREGAR ALGO

2.3. Ejercicio 3

TODO: AGREGAR ALGO

2.4. Ejercicio 4

TODO: AGREGAR ALGO

2.5. Ejercicio 5

TODO: AGREGAR ALGO

2.6. Ejercicio 6

TODO: AGREGAR ALGO

2.7. Ejercicio 7

TODO: AGREGAR ALGO

2.8. Ejercicio 8

TODO: AGREGAR ALGO

2.9. Ejercicio 9

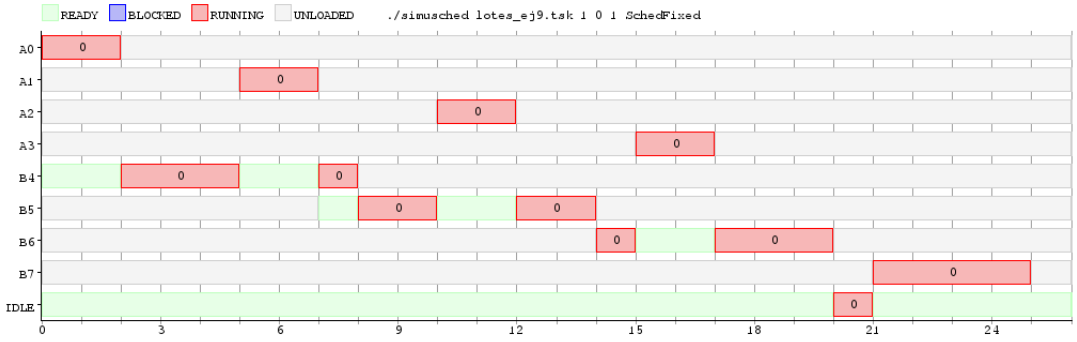
Para mostrar con un ejemplo un caso que no sea factible para el scheduler RM y si lo sea para EDF nos basto el siguiente lote de 2 tareas periodicas:

Nota: Para simplicar el ejemplo consideramos en este caso despreciable el tiempo de cambio de contexto.

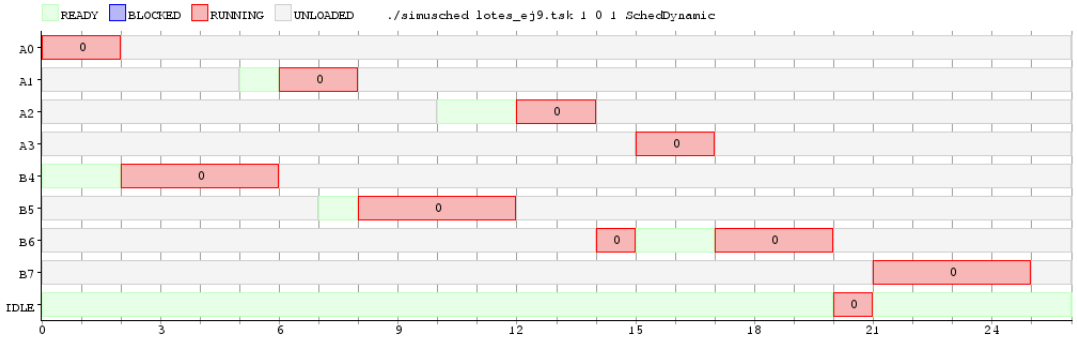
Lote:

```
@0:
&A4,5,1
@0:
&B4,7,3
```

Produciendo los siguientes gráfico:
Para RM:



Para EDF:



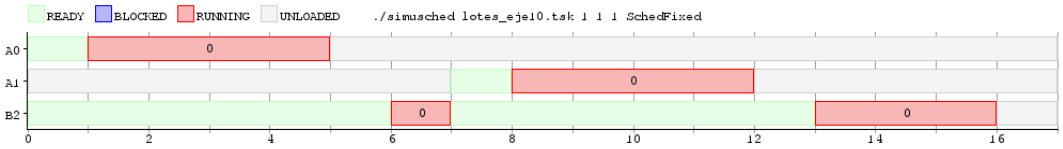
La tarea de la familia A tiene un periodo menor que el de la familia B por lo cual según el scheduler RM tiene mayor prioridad asignada. Esta elección de priorizar a la familia A provoca un perjuicio en la familia B por lo que la familia B no puede cumplir su deadline en $T = 7$ (Se produce un overflow).

En el caso del gráfico de EDF se puede observar que le da el CPU un poco más a la familia B por tener el deadline más próximo logrando evitar así el inclumplimiento del deadline (en $T = 5$ deja esperando a la tarea de la familia A).

Algo interesante para destacar es que en este caso ambos scheduler ocuparon el mismo tiempo el CPU con trabajos, lo que cambia es que el scheduler RM a diferencia de EDF tiende a gastar más tiempo de computo en tareas de mayor prioridad dejandolo poco procesamiento a las de menor prioridad.

2.10. Ejercicio 10

//TODO: AGREGAR TEXTO



//TODO: AGREGAR TEXTO

3. Conclusiones

TODO: agregar algo!