

Universidad Industrial de Santander

Ingeniería de Sistemas

Simulación Digital

2018-2

Simulación Round Robin

Autor

Marianne Solangel Rojas Robles – 2150286

Profesor

Urbano Eliécer Gómez Prada

Round Robin

Este programa que simula la ejecución de n procesos en un procesador de un solo núcleo mediante el algoritmo Round Robin, en el que es posible establecer el quantum (en segundos) y el tiempo que demora cada proceso es un aleatorio entre 1 y 100 (segundos). Este muestra el tiempo de cada proceso, los turnos de atención en el procesador y el número de turnos totales.

Propósito

Un sistema de monoprogramación puede atender a un único proceso en determinado tiempo. Con el fin de emular multiprogramación con un sólo procesador es necesario porcionar el tiempo de CPU en un tamaño determinado llamado quantum y atender un proceso únicamente durante el quantum, a cuya terminación el programa es interrumpido y se atiende el siguiente proceso. Este procedimiento se repite hasta que se finalicen todos los procesos.

En algunos sistemas operativos es posible atender a los distintos procesos asignándoles prioridad a partir del tiempo de CPU que requieren para ser culminados, brindando la mayor prioridad a los procesos más *pequeños*, para evitar la inanición. Sin embargo, el algoritmo Round Robin hace posible esta emulación de multiprogramación sin la necesidad de asignar prioridad a ninguno de los procesos, sin riesgo de inanición.

Este programa se realizó con el fin de ilustrar el funcionamiento del algoritmo Round Robin, mostrando la forma en que se maneja la atención de todos los procesos mediante turnos y el tiempo total de procesamiento.

Programa

Se realizó el siguiente diagrama de Flujo Nivel en el programa Evolución 4.5. Con fines ilustrativos, este diagrama representa únicamente dos procesos, cuyo nivel corresponde al tiempo requerido por el proceso para completarse y el tiempo transcurrido es la suma de todo el tiempo requerido para culminar todos los procesos. Se agregó un nivel adicional llamado contador, mediante el cual se regula el turno de atención para cada proceso. Los niveles CONTADOR Y T_TRANSUCRRIDO comienzan en 0.

Sus ecuaciones son las siguientes:

Iter:

IF(AND(PROCESO_1 > 0, PROCESO_2 > 0), 1, 0)

Procesar:

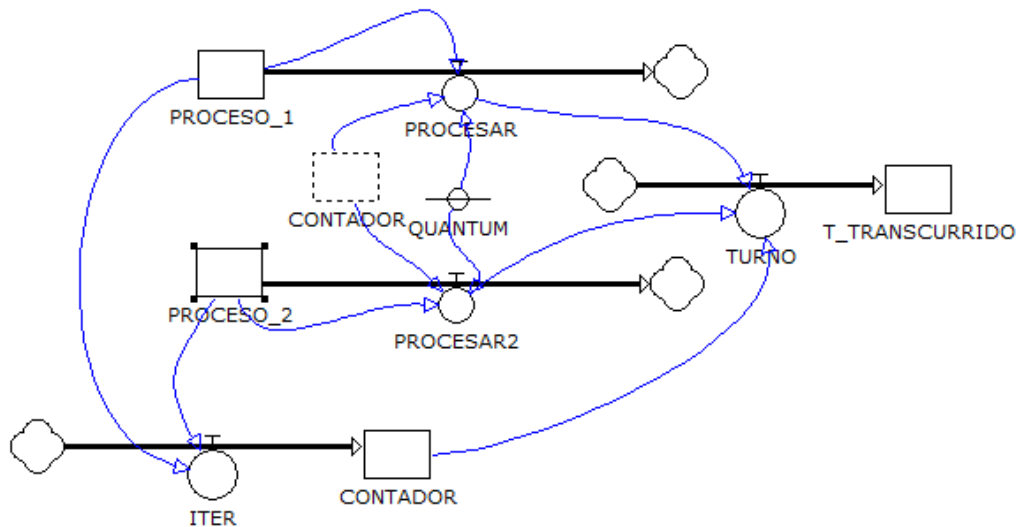
IF((MOD(CONTADOR,2) = 0), IF((PROCESO_1-QUANTUM) >= 0, QUANTUM, PROCESO_1), 0)

Procesar2:

$$\text{IF}((\text{MOD}(\text{CONTADOR}, 2) = 1), \text{IF}((\text{PROCESO_2} - \text{QUANTUM}) \geq 0, \text{QUANTUM}, \text{PROCESO_2}), 0)$$

Turno:

$$\text{IF}((\text{MOD}(\text{CONTADOR}, 2) = 0), \text{PROCESAR}, \text{PROCESAR2})$$



Nota: Para mayor interacción con el material ilustrado en este informe, favor referirse a los archivos adjuntos.

Para propósitos de este informe no se incluirá el código en el documento, pero se refiere como adjunto.

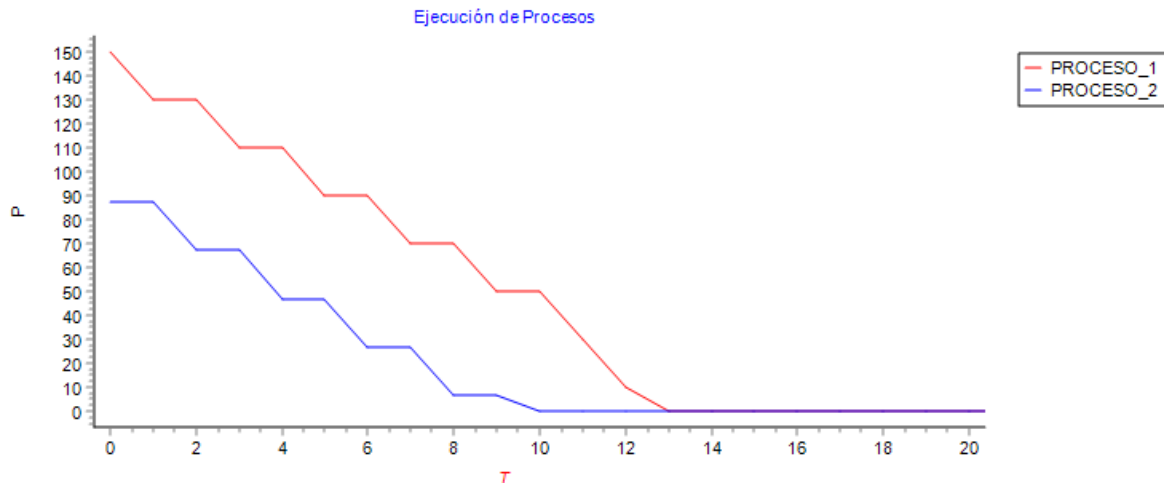
Resultado

Con el propósito de evaluar el funcionamiento de este programa se tomaron únicamente 2 procesos, con un quantum fijo de 20, y se tomó registro de 50 pruebas, donde se guardó el tiempo de cada uno de los procesos (que es asignado aleatoriamente) y el número de turnos totales. Del análisis de estos resultados, se obtuvieron los siguientes datos:

Proceso 1 min y max: 1, 98
Proceso 2 min y max: 1, 99
Tiempo total promedio: 102.580
Proceso 1 promedio: 51.120
Proceso 2 promedio: 51.290
Turnos promedio: 5.970

Se pudo evidenciar que en ninguno de los test se generó inanición en ninguno de los procesos, y el tiempo de ejecución total corresponde a la suma de los tiempos individuales de cada proceso. Esto

se puede observar más claramente en la siguiente gráfica generada en Evolución 4.5, realizada para un test con un proceso de 150 y otro de 87, y un quantum de 20.



Conclusión

Se puede concluir que el algoritmo Round Robin permite emular la multiprogramación en un sistema de monoprogramación de manera sencilla, dividiendo el tiempo de CPU en un quantum determinado, al final del cual el proceso siendo ejecutado se interrumpe y se atiende el siguiente proceso, hasta que todos se hayan terminado.

Fuentes de Información

1. Tanenbaum, Andrew S. Sistemas Operativos Modernos, Tercera Ed. Pearson Educacion, Mexico, 2009.
2. Stallings, William (2015). Operating Systems: Internals and Design Principles. Pearson. p. 409.