

Proyecto 2: Calendarización en Tiempo Real

Vargas A, Camacho A, Morales V

Tecnológico de Costa Rica *avargas@gmail.com, acamacho@gmail.com,*
verny.morales@gmail.com 3er Cuatrimestre

November 30, 2019

Rate Monotonic

Tipo

Algoritmo de Scheduling Dinámico, utilizado para la resolución de problemas caóticos, como por ejemplo el problema de los carros autónomos.

Manejo de prioridades

Algoritmo de prioridades estáticas, esto quiere decir que ninguna tarea puede cambiar su prioridad. Donde la prioridad de una tarea siempre es igual a su período. Período mas corto, mayor la prioridad.

Supuestos

Todas las tareas críticas son periódicas, e independientes. El tiempo de computación se conoce a priori, y el cambio de contexto es igual a cero, o ya esta considerado en el tiempo de computación.

Teoremas de Scheduling

Theorem (Parámetros a tomar en cuenta)

$\mu = \sum \frac{C_i}{P_i}$ Utilización del CPU $U_n = n2^{\frac{1}{n}} - 1$ donde n es la cantidad de tareas

Theorem (Condiciones de suficiencia)

$\mu \leq U_n$ Tareas calendarizables $\mu \geq U_n$ Debido a que es una condición de suficiencia podría ser calendarizable $\mu > 1$ Tareas no calendarizables

Early Deadline First

Tipo

Algoritmo de Scheduling Dinámico, en donde las tareas son periódicas. Se considera un algoritmo óptimo para algoritmos de prioridades dinámicas. Es un algoritmo de tipo expropiativo.

Manejo de prioridades

El nombre del algoritmo indica la política de prioridad. La prioridad es inversamente proporcional al tiempo que falta para el deadline. El deadline de cada tarea es igual al período de la misma.

Supuestos

Todas las tareas críticas son periódicas, e independientes. El tiempo de computación se conoce a priori, y el cambio de contexto es igual a cero, o ya esta considerado en el tiempo de computación.

Teoremas de Scheduling

Theorem (Parámetros a tomar en cuenta)

$$\mu = \sum \frac{C_i}{P_i} \text{ Utilización del CPU}$$

Theorem (Condiciones de suficiencia)

$\mu \leq 1$ *Tareas calendarizables, ya que es una condición necesario y de suficiencia*

Prueba de Schedulability

$$\mu = 1,000000 \leq U(n) = 0,828427$$

Las pruebas son calendarizables

Prueba de Schedulability

$$\mu = 1,000000 \leq 1$$

Las pruebas son calendarizables

All Resultados de Simulador

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Task 0		■	■			■	■			■	■			■	■			■	■			■	■		
Task 1				■	■			■	■			■	■			■	■			■	■			■	
Task Arrivals	■				■				■				■			■	■				■				■

Table: RM Simulation results

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Task 0		■	■			■	■			■	■			■	■			■	■			■	■		
Task 1				■	■			■	■			■	■			■	■			■	■			■	
Task Arrivals	■				■				■				■			■	■				■				■

Table: EDF Simulation results