

## 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 29, 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.

## 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*

## Tipo y manejo de prioridades

Es un algoritmo donde las prioridades se manejan de forma dinámica. Por cada unidad de tiempo se deben de evaluar las prioridades para así conocer el Laxity de cada tarea, y tomar una decisión sobre el scheduling de las mismas.

# Teoremas de Scheduling

## Theorem (Parámetros a tomar en cuenta)

*Laxity =  $D_i - T_i + C_i$  Donde  $D$  es el deadline próximo de la tarea,  $T$  es el tiempo actual de ejecución y  $C$  es el tiempo computación al faltante de la tarea*

## Theorem (Criterio de Scheduling)

*Se toma el  $L_i$  menor entre todas las tareas en la cola de Ready y se ejecuta la tarea para cada tiempo de ejecución*

# RM Scheduling Results

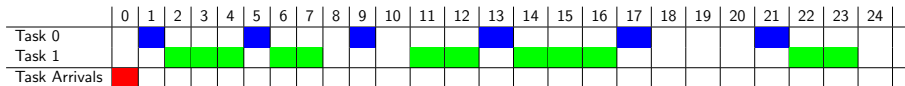


Table: RM Simulation results



# EDF Scheduling Results

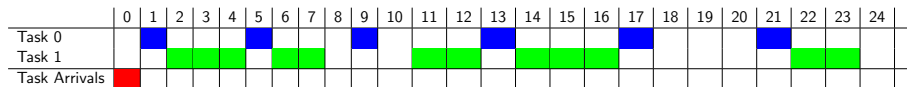


Table: EDF Simulation results

# LLF Scheduling Results

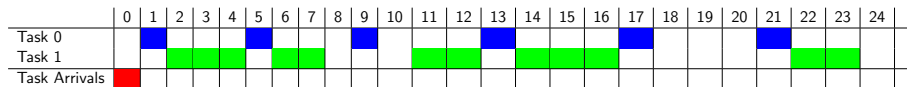


Table: LLF Simulation results