

# Tema 7

## Planificación de Tareas Periódicas



# Objetivos

1. Conocer y entender el funcionamiento de distintos algoritmos de planificación de sistemas con **tareas periódicas e independientes**
2. Conocer distintos métodos para predecir si estos sistemas cumplirán con sus **restricciones temporales**



# Índice

1. **Planificadores basados en prioridades**
2. *Tests de planificabilidad*
3. *Test de factor de utilización y Test de factor de carga*
4. *Test basado en el cronograma*
5. *Test de tiempos de respuesta*



# Recordemos el concepto de *scheduling*

- El **scheduling** proporciona:
  - un **PLANIFICADOR**
    - algoritmo que determina el orden del uso de recursos del sistema (en particular, la CPU)
    - restringe el indeterminismo de un sistema concurrente (sistema predecible)
  - un **TEST DE PLANIFICABILIDAD**
    - método para determinar si el sistema cumplirá con las restricciones temporales establecidas al usar un determinado planificador



# Recordemos el concepto de planificador óptimo

- En cualquiera de las categorías de planificadores, se dice que un planificador es óptimo dentro de esa categoría cuando es capaz de encontrar una planificación factible de cualquier conjunto de tareas planificable
- Consecuencias de utilizarlo:
  - Si un conjunto de tareas no es planificable con el planificador óptimo, podemos afirmar que no es planificable, ya que no lo será con ningún otro planificador (de esa misma categoría)
  - Si un conjunto de tareas no es planificable con un planificador no óptimo, no podemos garantizar que no sea planificable

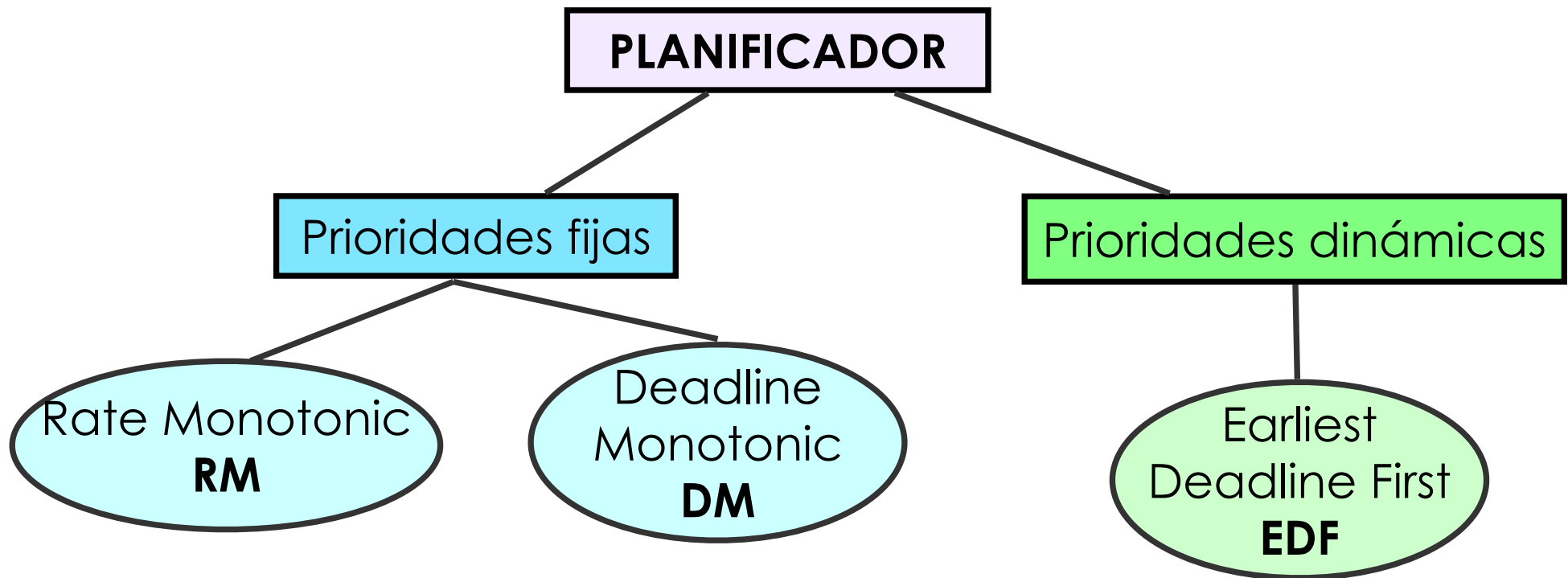


# Planificadores con prioridades

- En los sistemas de tiempo real la prioridad de un proceso se deriva de sus requerimientos temporales
- De todos los procesos listos para ejecución, se ejecuta el de mayor prioridad
- En un esquema **expulsivo** (**preemptive**), hay un cambio inmediato al proceso de mayor prioridad
- En un esquema **no expulsivo** (**non-preemptive**), el proceso de baja prioridad podrá continuar hasta que acabe
- Los esquemas expulsivos permiten a los procesos de mayor prioridad ser más reactivos y, por lo tanto, son los más adecuados



# Planificadores Óptimos



# Planificadores de prioridad fija (FPS)

- Cada proceso tiene una **prioridad fija (estática)**
  - Se basa en parámetros fijos de la tarea
  - Se establece antes de su ejecución
- Ejemplos:
  - Rate Monotonic (**RM**)
  - Deadline Monotonic (**DM**)





# Planificadores RM y DM

## RM : Rate Monotonic

- Se supone que  $D_i = T_i$
- La prioridad se asigna en función del **PERIODO**
- A menor periodo  $\Rightarrow$  mayor prioridad

$$T_i < T_j \Rightarrow P_i > P_j$$

## DM : Deadline Monotonic

- Se supone que  $D_i \leq T_i$
- La prioridad se asigna en función del **DEADLINE**
- A menor deadline  $\Rightarrow$  mayor prioridad

$$D_i < D_j \Rightarrow P_i > P_j$$



## 1. Planificadores basados en prioridades

### Asignación de prioridades

RM		
Tarea	T	P
a	25	5
b	60	3
c	42	4
d	105	1
e	75	2

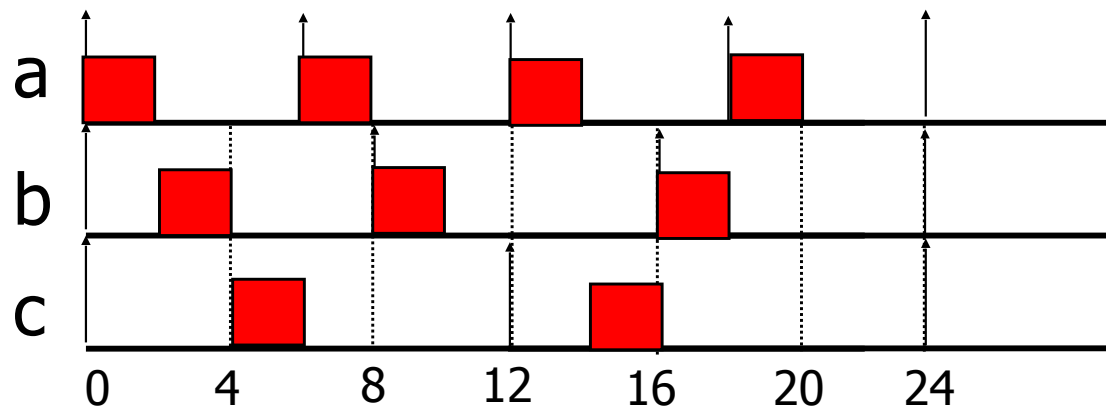
DM			
Tarea	T	D	P
a	20	5	4
b	15	7	3
c	10	10	2
d	20	20	1



## 1. Planificadores basados en prioridades

# Cronograma utilizando RM

Tarea	C	T=D
a	2	6
b	2	8
c	2	12



# Planificadores dinámicos

- Cada proceso tiene una **prioridad dinámica**
  - Se basa en parámetros dinámicos de la tarea
  - Cambia durante su ejecución
- Se obtienen secuenciaciones más flexibles
- Ejemplos:
  - Earliest Deadline First (**EDF**)
  - Least Slack Time First (**LSTF**)



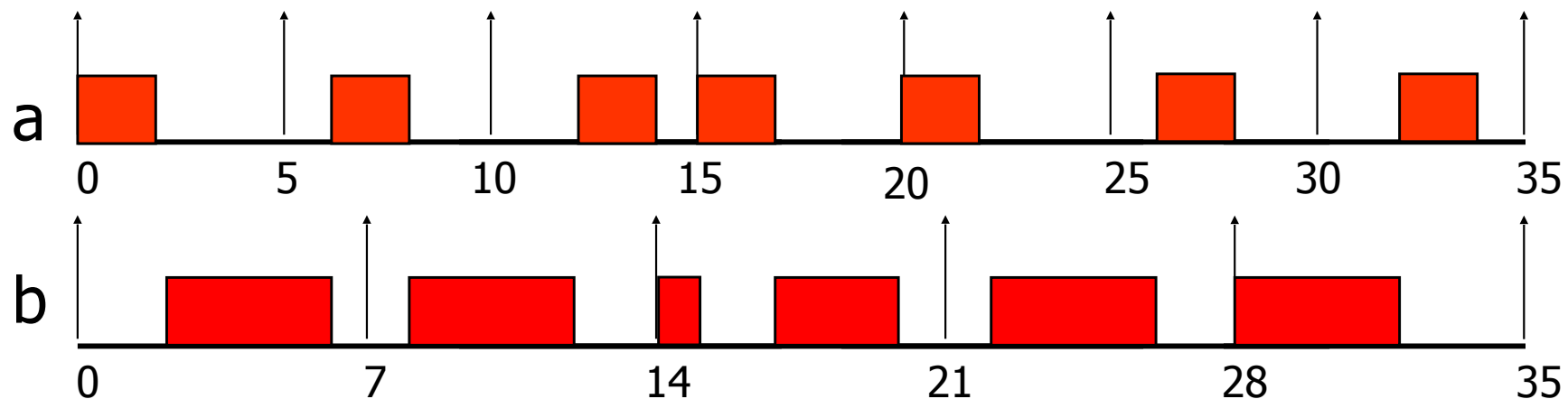
# Planificador EDF

- Selecciona la siguiente tarea a ejecutar dependiendo de su **deadline absoluto**
  - Se elige la tarea con *deadline* absoluto más cercano
- La decisión se realiza en tiempo de ejecución



# Cronograma utilizando EDF

Tarea	$T=D$	C
a	5	2
b	7	4



### EDF vs FPS

- EDF permite mayores utilizaciones que FPS
- EDF es más complejo de implementar y es ejecutado on-line
- En situaciones de sobrecarga, EDF es impredecible mientras que FPS es más predecible



# Planificador EDF en ADA: Política de *dispatching*

```
pragma Task_Dispatching_Policy(policy_identifier)
```

```
pragma Priority_Specific_Dispatching (policy, firts_priority,last_priority)
```

- Utilizar el identificador **EDF\_Across\_Priorities**
  - Se elige la tarea con menor deadline absoluto
- y en la especificación de las tareas el pragma:

```
pragma Relative_Deadline(plazo_ejecucion)
```





# Paquete Ada.Dispatching.EDF

```
with Ada.Real_Time;  
with Ada.Task_Identification;  
package Ada.Dispatching.EDF is  
  subtype Deadline is Ada.Real_Time.Time;  
  Default_Deadline : constant Deadline := Ada.Real_Time.Time_Last;  
  procedure Set_Deadline (D : in Deadline;  
                        T : in Ada.Task_Identification.Task_Id :=  
                        Ada.Task_Identification.Current_Task);  
  procedure Delay_Until_And_Set_Deadline (  
    Delay_Until_Time : in Ada.Real_Time.Time;  
    Deadline_Offset : in Ada.Real_Time.Time_Span);  
  function Get_Deadline (T : Ada.Task_Identification.Task_Id :=  
    Ada.Task_Identification.Current_Task) return Deadline;  
end Ada.Dispatching.EDF;
```



# Índice

1. *Planificadores basados en prioridades*
2. **Tests de planificabilidad**
3. *Test de factor de utilización y Test de factor de carga*
4. *Test basado en el cronograma*
5. *Test de tiempos de respuesta*



### Tipos de *tests* de planificabilidad

- Test basado en **factores de utilización**
- Test basado en **factores de carga**
- Test basado en el **cronograma** (diagrama de tiempo)
- Test basado en **tiempos de respuesta**



### Tests suficientes y/o necesarios

- Si un test de planificabilidad es **suficiente**, un resultado positivo del test garantiza que todos los *deadlines* siempre se cumplen
- Si un test de planificabilidad es **necesario**, un resultado negativo del test implica que algún *deadline* será incumplido



# Índice

1. *Planificadores basados en prioridades*
2. Tests de planificabilidad
3. **Test de factor de utilización y Test de factor de carga**
4. Test basado en el cronograma
5. *Test de tiempos de respuesta*



### 3. Test de factor de utilización y Test de factor de carga

## Recuerda que ...

- El **factor de utilización** de una tarea es el porcentaje de uso de procesador que requiere una tarea para su ejecución

$$u = \frac{C}{T}$$

- y es diferente del **factor de carga**:

$$ch = \frac{C}{D}$$



## Test de factores de utilización

■ Se puede utilizar sólo cuando  $T_i = D_i$

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq L_U$$

El **límite de planificabilidad**  $L_U$  es el valor máximo de  $U$  que puede alcanzar un conjunto de tareas garantizándose el cumplimiento de sus deadlines



## Test de factores de utilización con RM

Dado un conjunto  $\Gamma$  de  $N$  tareas periódicas secuenciadas según **RM**,  $\Gamma$  será planificable **SI**:

$$\sum_{i=1}^N \frac{C_i}{T_i} \leq N(2^{1/N} - 1) = L_U(N)$$

$N$	$L_U(N)$
1	100 %
2	82,8 %
3	77,9 %
4	75,7 %
...	...
$\infty$	69,3%

Se trata de una condición **SUFICIENTE**





### 3. Test de factor de utilización y Test de factor de carga

## Condición suficiente

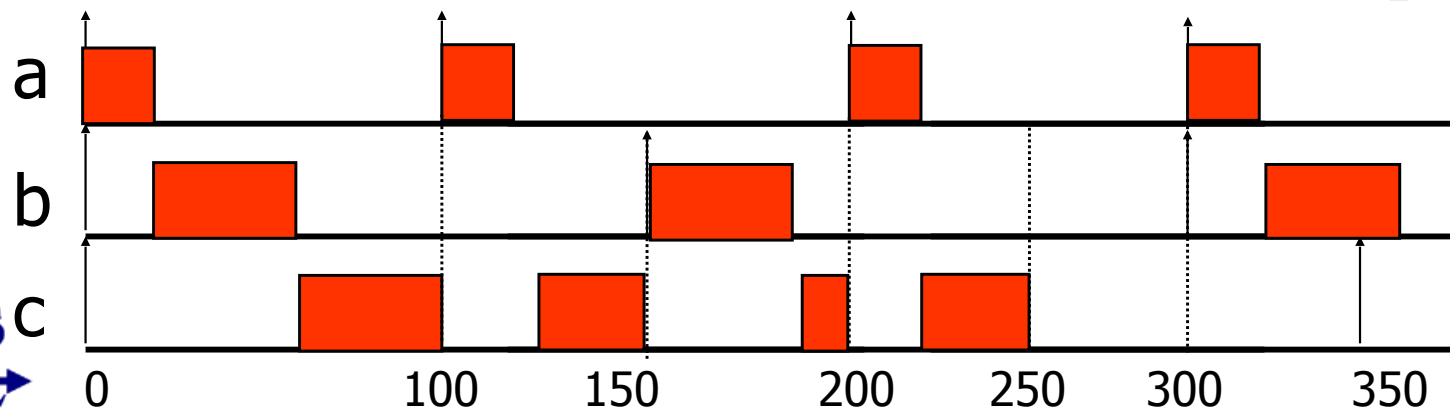
Tarea	C	T=D	P	U
a	20	100	3	0,2
b	40	150	2	0,267
c	100	350	1	0,286

**RM**

N	$L_U(N)$
1	1.000
2	0.828
3	0.779
4	0.756
5	0.743
...	...
$\infty$	0.693

$$\sum_{i=1}^N C_i / T_i = 0,753 \leq L_U(3) = 0,779$$

**Test OK !**



**OK !**



### 3. Test de factor de utilización y Test de factor de carga

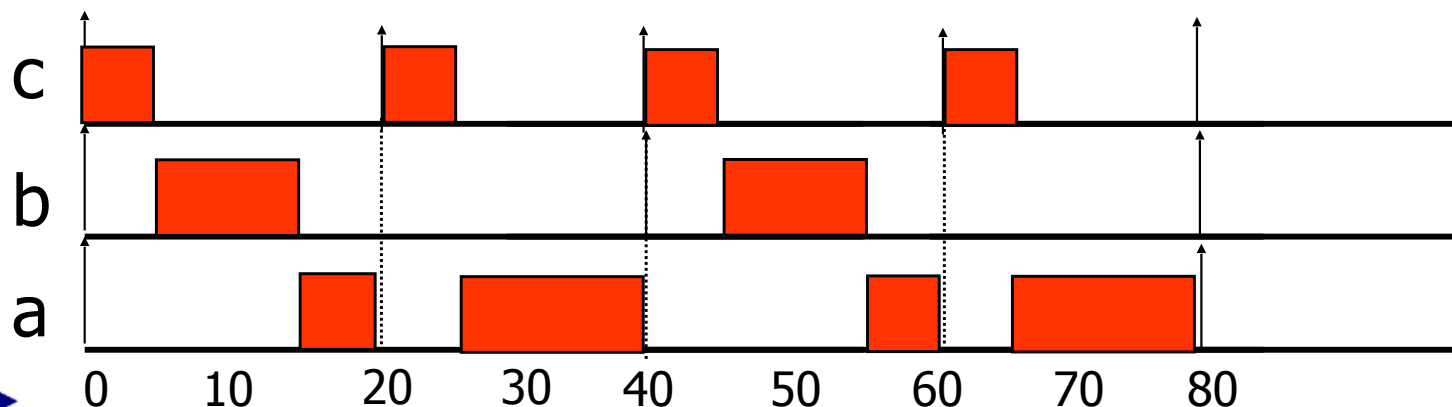
## Condición no necesaria

Tarea	C	T=D	P	U
a	40	80	1	0,50
b	10	40	2	0,25
c	5	20	3	0,25

**RM**

$$\sum_{i=1}^N C_i / T_i = 1 \leq L_U(3) = 0,779$$

**Falla test !**



**OK !**



## Test de factores de utilización con EDF

Dado un conjunto  $\Gamma$  de  $N$  tareas periódicas secuenciadas según **EDF**, con  $\mathbf{T_i = D_i}$ ,  $\Gamma$  será planificable **SI Y SOLO SI**:

$$\sum_{i=1}^N \frac{C_i}{T_i} \leq 1$$

$$L_u = 1$$

Se trata de una condición **NECESARIA y SUFICIENTE**



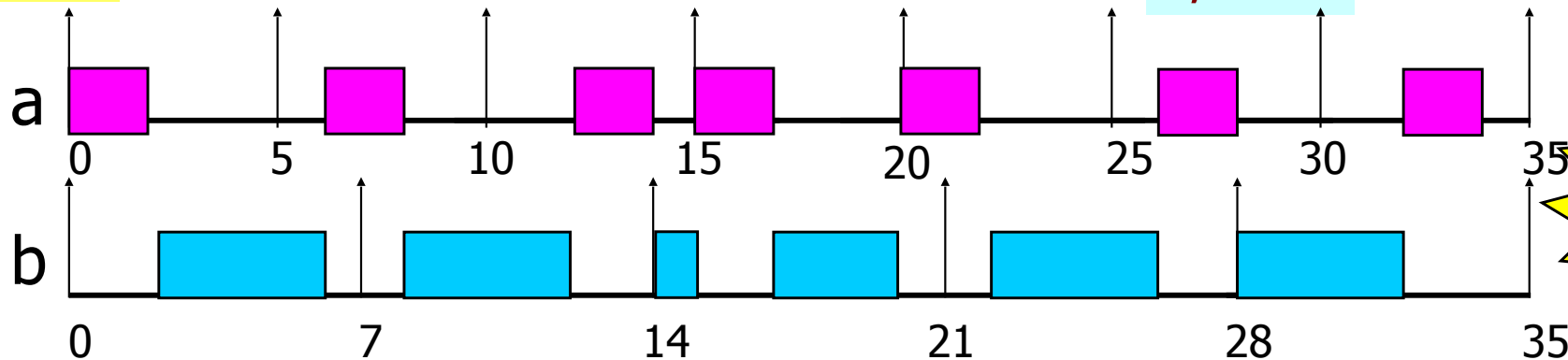
### 3. Test de factor de utilización y Test de factor de carga

## Ejemplo

**EDF**

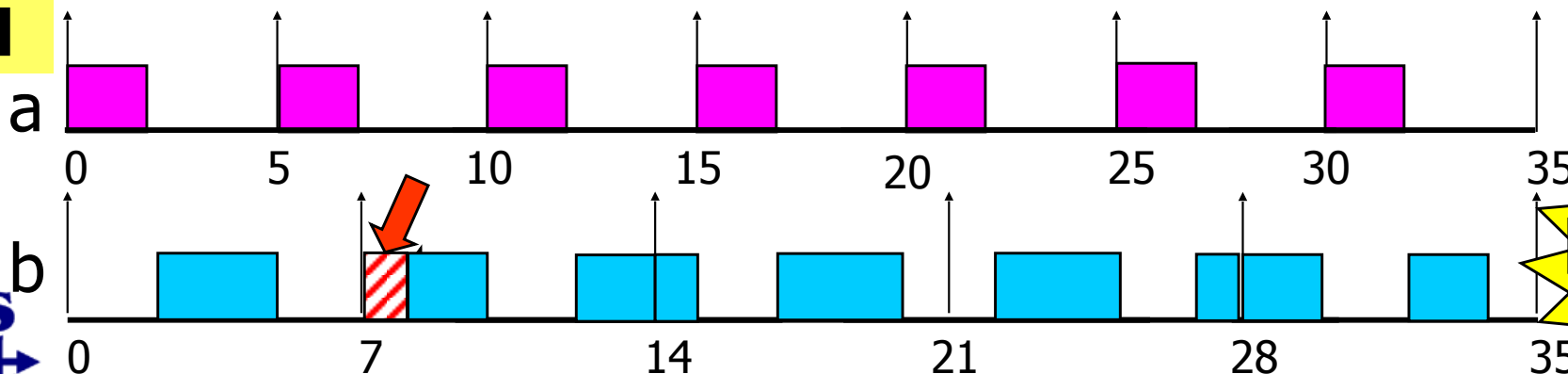
Tarea	C	$T=DL_u$
a	2	5
b	4	7

Uso total 97 %



**OK!**

**RM**



**Falla!**



## Test de factores de carga con DM

Dado un conjunto  $\Gamma$  de  $N$  tareas periódicas secuenciadas según **DM**,  $\Gamma$  será planificable **SI**:

$$\sum_{i=1}^N \frac{C_i}{D_i} \leq N(2^{1/N} - 1)$$

Se trata de una condición **SUFICIENTE**



## Test de factores de carga con EDF

Dado un conjunto  $\Gamma$  de  $N$  tareas periódicas secuenciadas según **EDF**, con  $\mathbf{D_i} < \mathbf{T_i}$ ,  $\Gamma$  será planificable **SI**:

$$\sum_{i=1}^N \frac{C_i}{D_i} \leq 1$$

Se trata de una condición **SUFICIENTE**



# Índice

1. *Planificadores basados en prioridades*
2. Tests de planificabilidad
3. *Test de factor de utilización y Test de factor de carga*
4. **Test basado en el cronograma**
5. *Test de tiempos de respuesta*

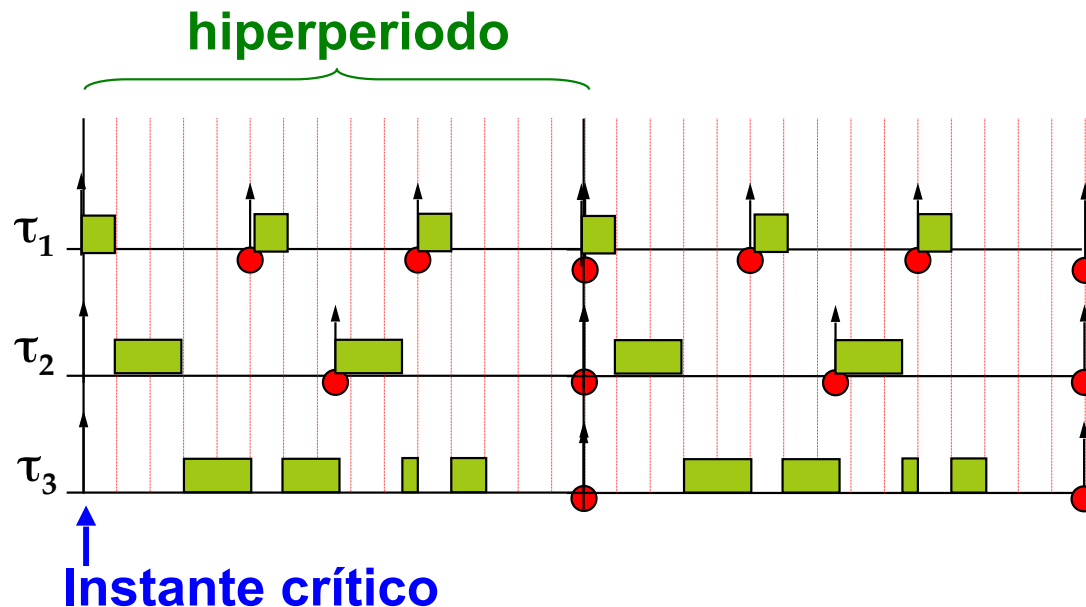


#### 4. Test basado en el cronograma

## Conceptos a tener en cuenta

- Se trata de obtener el peor tiempo de respuesta de cada tarea observando el cronograma
- **Instante crítico** : máxima carga del procesador
- **Hiperperiodo**: mínimo común múltiplo de los periodos de activación de todas las tareas

Tarea	C	T=D
$\tau_1$	2	10
$\tau_2$	4	15
$\tau_3$	10	30



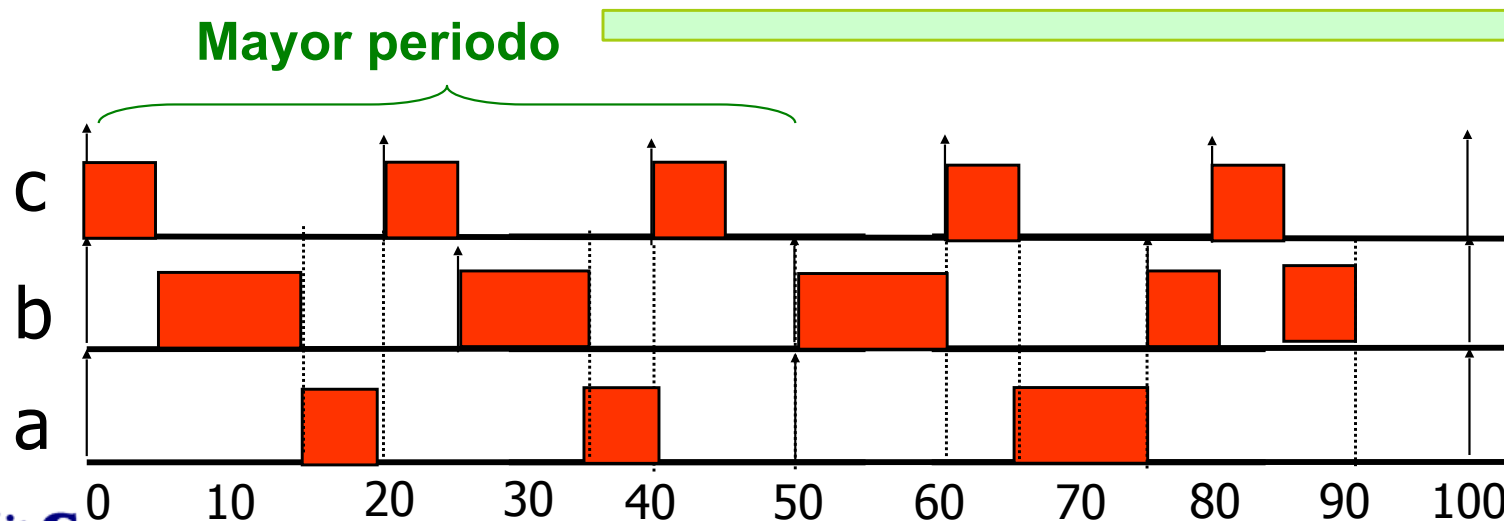


#### 4. Test basado en el cronograma

## Test basado en cronograma con prioridades fijas

Tarea	C	T=D	P
a	10	50	1
b	10	25	2
c	5	20	3

**RM**



$$R_a = 40$$

$$R_b = 15$$

$$R_c = 5$$

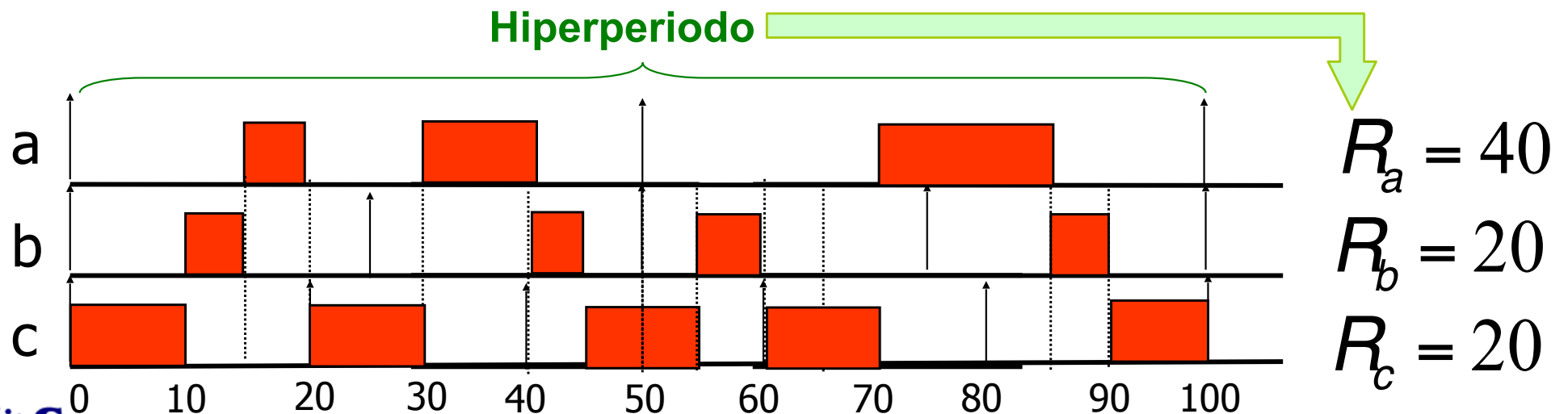


#### 4. Test basado en el cronograma

## Test basado en cronograma con prioridades dinámicas

Tarea	C	T=D
a	<b>15</b>	50
b	<b>5</b>	25
c	<b>10</b>	20

**EDF**



# Índice

1. *Planificadores basados en prioridades*
2. Tests de planificabilidad
3. *Test de factor de utilización y Test de factor de carga*
4. *Test basado en el cronograma*
5. **Test de tiempos de respuesta**



# Test de tiempos de respuesta

Dado un conjunto  $\Gamma$  de  $N$  tareas periódicas planificadas según **RM** o **DM**,  $\Gamma$  será planificable **SI Y SOLO SI**:

$$R_i \leq D_i, \forall i = 1..N$$

Donde  $R_i$  denota el **peor tiempo de respuesta** de la tarea  $i$

Se trata de una condición **NECESARIA y SUFICIENTE**



### Peor tiempo de respuesta

- ▣ **Instante crítico de una tarea  $\tau_i$ :** es el instante de activación de la tarea que da lugar a un tiempo de respuesta máximo para  $\tau_i$
- ▣ El **peor tiempo de respuesta  $R_i$**  para una tarea periódica  $\tau_i$  es su tiempo máximo de respuesta y se calcula, en su instante crítico, como:

$$R_i = C_i + I_i$$

Donde  $I$  es la interferencia de tareas de mayor prioridad



# Cálculo de interferencias

- Durante  $R_i$ , cada proceso  $\tau_j$  (de mayor prioridad que  $\tau_i$ ) se activará un cierto número de veces:

$$\text{Número de activaciones}_j = \left\lceil \frac{R_i}{T_j} \right\rceil$$

La función techo  $\lceil \cdot \rceil$  devuelve el menor entero mayor o igual que el número real sobre el que actúa

- Cada activación del proceso  $\tau_j$  ocasionará una interferencia de  $C_j$  y por lo tanto:

$$\text{Interferencia de } \tau_j \text{ sobre } \tau_i = \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$



## 5. Test de tiempos de respuesta

### Cálculo de $R_i$

- Cada proceso de mayor prioridad que  $\tau_i$  interferirá con dicho proceso y por lo tanto:

$$I_i = \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

$hp(i)$  es el conjunto de procesos con mayor prioridad que  $\tau_i$

$$R_i = C_i + I_i \longrightarrow R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$



# Resolución mediante relación de recurrencia

■ La solución a la ecuación  $R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$  no es sencilla

■ La forma más fácil de resolverla es mediante la siguiente relación de recurrencia:

$$R_i^{k+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot C_j$$

■ k indica el número de iteración

■ El conjunto  $\{R_i^0, R_i^1, \dots, R_i^k, \dots\}$  es monotónicamente no decreciente

■ Cuando  $R_i^k = R_i^{k+1}$ , entonces se ha encontrado la solución





## 5. Test de tiempos de respuesta

### Ejemplo de cálculo de $R_i$

RM	Tarea	C	T=D	P
	a	40	80	1
	b	10	40	2
	c	5	20	3

Tarea **c**

$$R_c^0 = 5$$

$$R_c^1 = 5$$

$$R_c^0 = R_c^1 = 5 \leq 20$$



## 5. Test de tiempos de respuesta

### Ejemplo de cálculo de $R_i$

**RM**

Tarea	C	T=D	P
a	40	80	1
b	10	40	2
c	5	20	3

Tarea **b**

$$R_b^0 = 10$$

$$R_b^1 = 10 + 5 \cdot \left\lfloor \frac{10}{20} \right\rfloor = 10 + 5 = 15$$

$$R_b^2 = 10 + 5 \cdot \left\lfloor \frac{15}{20} \right\rfloor = 10 + 5 = 15$$

$$R_b^1 = R_b^2 = 15 \leq 40$$



## 5. Test de tiempos de respuesta

### Ejemplo de cálculo de $R_i$

RM	Tarea	C	T=D	P
	a	40	80	1
	b	10	40	2
	c	5	20	3

Tarea a

$$R_a^0 = 40$$

$$R_a^1 = 40 + 10 \cdot \left\lceil \frac{40}{40} \right\rceil + 5 \cdot \left\lceil \frac{40}{20} \right\rceil = 40 + 10 + 10 = 60$$

$$R_a^2 = 40 + 10 \cdot \left\lceil \frac{60}{40} \right\rceil + 5 \cdot \left\lceil \frac{60}{20} \right\rceil = 40 + 20 + 15 = 75$$

$$R_a^3 = 40 + 10 \cdot \left\lceil \frac{75}{40} \right\rceil + 5 \cdot \left\lceil \frac{75}{20} \right\rceil = 40 + 20 + 20 = 80$$

$$R_a^4 = 40 + 10 \cdot \left\lceil \frac{80}{40} \right\rceil + 5 \cdot \left\lceil \frac{80}{20} \right\rceil = 40 + 20 + 20 = 80$$



# Test de tiempos de respuesta con EDF

- El instante crítico de cada tarea no se produce cuando todas las tareas se activan al mismo tiempo
- Determinar el tiempo de respuesta es mucho más complejo
  - Ejemplo: en un sistema pequeño de sólo 4 procesos, pero con periodos 24, 50, 73 y 101, habría que analizar todas las **activaciones** en las 4.423.800 primeras u.t. (mientras que con FPS sólo en las 101 primeras u.t.)



# Resumen Tests de Planificabilidad

	Prioridades Fijas		Prioridades Dinámicas	
	RM ( $D_i = T_i$ )	DM ( $D_i \leq T_i$ )	EDF ( $D_i = T_i$ )	EDF ( $D_i \leq T_i$ )
Factores de utilización	Suficiente	---	Necesario y suficiente	---
Factores de carga	---	Suficiente	---	Suficiente
Cronograma	Necesario y suficiente	Necesario y suficiente	Necesario y suficiente	Necesario y suficiente
Tiempos de respuesta	Necesario y suficiente	Necesario y suficiente	---	---



# Conclusiones

- Cuando una aplicación de control se compone de varias tareas periódicas concurrentes con restricciones de tiempo individuales, se debe garantizar que se activen regularmente con su apropiado periodo y se completan dentro de su deadline
- Los planificadores más utilizados son los basados en prioridad
- Los *tests* de planificabilidad se utilizan para determinar si el sistema cumplirá con las restricciones establecidas



# Bibliografía Recomendada

Sistemas de tiempo real y lenguajes de programación (**3ª edición**)

Alan Burns and Andy Wellings

Addison Wesley (2002)

 Capítulo 13 (Apartados: del 13.3 al 13.6; y 13.9)

Hard real-time computing systems (**Second edition**)

Giorgio C. Buttazzo


Kluwer Academic Publishers (2004)

 Capítulo 4

Scheduling in real-time systems

Cottet Francis and others.

Wiley (2002)

 Capítulo 2 (Apartado 2.1)



# Otras fuentes de información

Manual de Referencia de Ada2005

 Anexos: D.2.6

