



UD6.-Sincronización en Sistemas de Tiempo Real



Concurrencia y Sistemas Distribuidos



Objetivos de la Unidad Didáctica

- ▶ Comprender qué son los Sistemas de Tiempo Real (STR) y sus particularidades respecto a la sincronización de tareas.
 - ▶ Conocer las características de los STR
 - ▶ Comprender cómo realizar la planificación de las tareas en los STR y en qué consiste el análisis de planificación.
- ▶ Analizar los problemas de sincronización que aparecen en este tipo de sistemas.
 - ▶ Identificar el problema de la inversión de prioridades
 - ▶ Conocer estrategias de resolución para resolver los problemas de sincronización



Contenido

► Introducción a los STR

- ▶ Definición de STR
- ▶ Características de los STR
- ▶ Planificación de tareas en STR
- ▶ Análisis de STR

► Sincronización de tareas

- ▶ El problema de la inversión de prioridades
- ▶ Protocolo del Techo de Prioridad Inmediato
- ▶ Tiempo de respuesta con factores de bloqueo



Introducción a los STR

- ▶ Consideremos un sistema combinación de uno o varios computadores, dispositivos hardware de entrada/salida, y software de propósito especial, en la que:
 - ▶ hay una fuerte interacción con el entorno
 - ▶ el entorno cambia con el tiempo
 - ▶ el sistema controla o reacciona de forma simultánea a diferentes aspectos del entorno
- ▶ Como resultado:
 - ▶ se imponen requisitos temporales sobre el software
 - ▶ el software es de naturaleza concurrente
- ▶ Algunos requisitos de tiempo real frecuentes:
 - ▶ realizar actividades periódicas, solicitadas a intervalos periódicos
 - ▶ deben completar un trabajo antes de un plazo



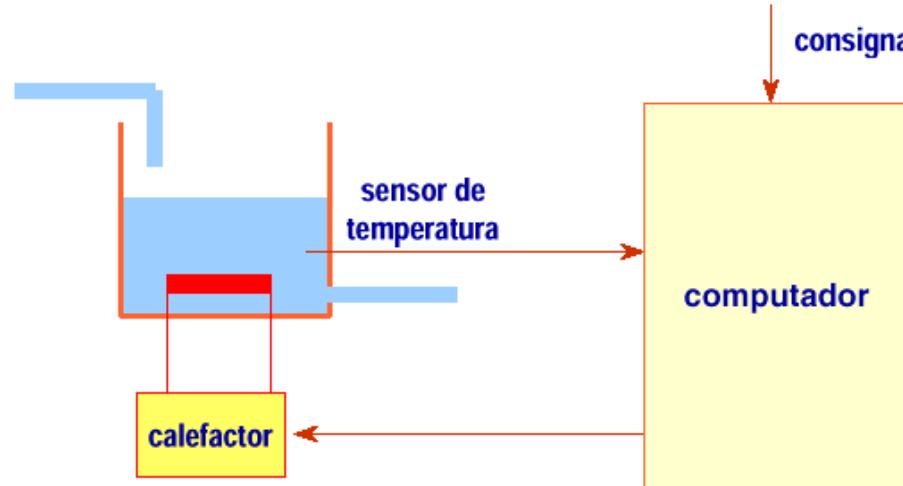
Definición de STR

- ▶ Un **sistema de tiempo real** (STR) es un sistema informático que
 - ▶ Interacciona repetidamente con su entorno físico
 - ▶ Responde a los estímulos que recibe del mismo dentro de un plazo de tiempo determinado
 - ▶ Para que el funcionamiento del sistema sea **correcto** no basta con que las acciones sean correctas, sino que tienen que ejecutarse dentro del intervalo de tiempo especificado

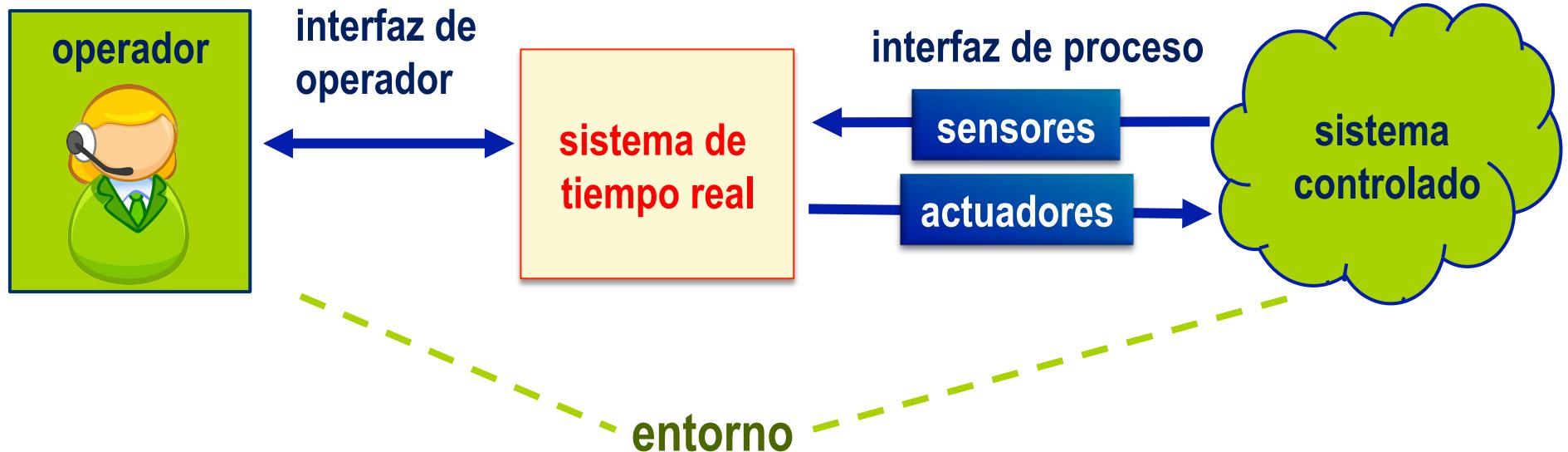
El tiempo en que se ejecutan las acciones del sistema es significativo

- ▶ Ejemplos:
 - ▶ control de procesos industriales, aviónica, control de tráfico aéreo, control de trenes, control de automóviles, telecomunicaciones, electrónica de consumo, sistemas multimedia.

Ejemplo: control de procesos

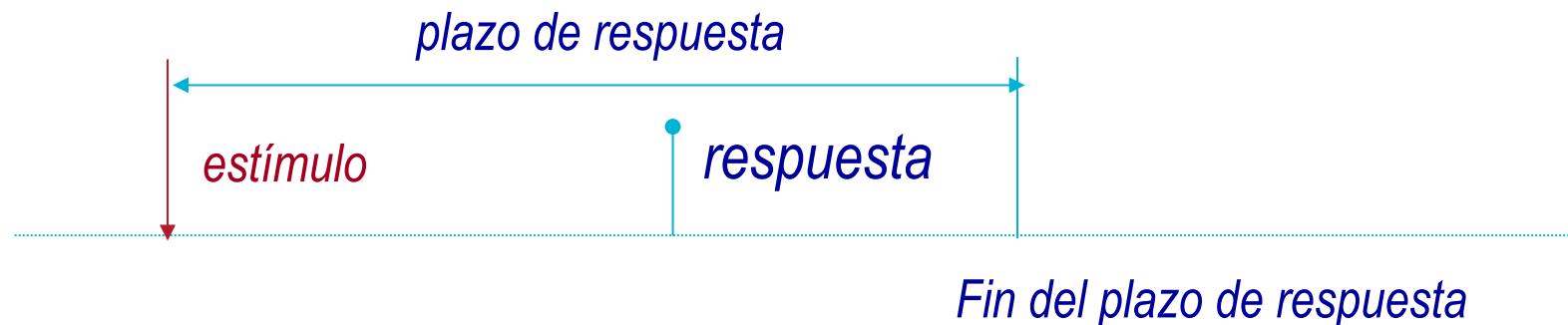


- ▶ El sistema de tiempo real actúa sobre el sistema controlado para conseguir que tenga un comportamiento determinado
- ▶ El comportamiento es muy regular (muestreo periódico)
- ▶ Normalmente se usa un esquema con realimentación
 - ▶ la acción de control es función de la desviación entre los valores de referencia y los valores medidos de las variables
 - ▶ el diseño de los algoritmos de control es un problema importante



- ▶ Un sistema de tiempo real debe responder a los **estímulos** de su entorno dentro de un **intervalo de tiempo** determinado por las propiedades dinámicas del entorno

- El intervalo de respuesta se caracteriza por un plazo (**deadline**)



- Estricto o crítico (**hard**) una respuesta fuera de plazo es inadmisible
 - (Ejemplo: UCI, central nuclear, control de frenado, airbag ...)
- firme (**firm**) una respuesta fuera de plazo no tiene utilidad
 - (Ejemplo: en sistemas multimedia (procesado de audio o vídeo) la pérdida de una trama de audio o vídeo)
- flexible (**soft**) una respuesta fuera de plazo tiene una utilidad reducida
 - (Ejemplo: adquisición de datos meteorológicos, contestador automático)



Contenido

- ▶ Introducción a los STR
 - ▶ Definición de STR
 - ▶ Características de los STR
 - ▶ Planificación de tareas en STR
 - ▶ Análisis de STR
- ▶ Sincronización de tareas
 - ▶ El problema de la inversión de prioridades
 - ▶ Protocolo del Techo de Prioridad Inmediato
 - ▶ Tiempo de respuesta con factores de bloqueo



Características de los STR

▶ Concurrencia

- ▶ Los componentes del sistema controlado funcionan simultáneamente
- ▶ El STR debe atenderlo y generar las acciones de control simultáneamente

▶ Interacción con dispositivos físicos

- ▶ Los STR interactúan con su entorno mediante diversos tipos de dispositivos no convencionales: convertidores A/D y D/A, entradas y salidas digitales, ... (interfaces con sensores, actuadores, periféricos especiales, ...)
- ▶ Los componentes del software que controlan el funcionamiento de estos dispositivos (manejadores, "drivers") son, en general, dependientes del sistema concreto



Características de los STR

► **Fiabilidad y seguridad**

- ▶ Un fallo en un sistema de control puede hacer que el sistema controlado se comporte de forma peligrosa o antieconómica
- ▶ Es importante asegurar que si el sistema de control falla lo haga de forma que el sistema controlado quede en un estado seguro
 - ▶ Hay que tener en cuenta los posibles fallos o excepciones en el diseño

► **Determinismo Temporal**

- ▶ Acciones en intervalos de tiempo determinados
- ▶ Es fundamental que el comportamiento temporal de los STR sea determinista
 - ▶ no hay que confundirlo con la necesidad de que sea eficiente
 - ▶ el sistema debe responder correctamente en todas las situaciones
 - ▶ hay que prever el comportamiento en el peor caso posible



Desarrollo de software para STR críticos

- ▶ Las características específicas de los **sistemas de tiempo real críticos** condicionan los métodos y herramientas que se utilizan para desarrollar el software
 - ▶ No todas las técnicas que se usan para construir otros tipos de sistemas sirven para el software de tiempo real crítico
 - ▶ suele haber problemas de fiabilidad y determinismo temporal
-
- ▶ La **predictibilidad** es uno de los objetivos principales en sistemas de tiempo real.
 - ▶ Se debe realizar un **análisis de la planificabilidad** de las tareas de un sistema de tiempo real para predecir si las tareas cumplirán sus requisitos temporales.



Contenido

- ▶ Introducción a los STR
 - ▶ Definición de STR
 - ▶ Características de los STR
 - ▶ Planificación de tareas en STR
 - ▶ Análisis de STR
- ▶ Sincronización de tareas
 - ▶ El problema de la inversión de prioridades
 - ▶ Protocolo del Techo de Prioridad Inmediato
 - ▶ Tiempo de respuesta con factores de bloqueo



Planificación de tareas en STR

- ▶ Objetivo: Planificar el uso de los recursos del sistema (en particular el procesador), para poder garantizar los requisitos temporales de las tareas *.
- ▶ Un paradigma de planificación consta de:
 - ▶ Un **algoritmo de planificación**, que determina el orden de acceso de las tareas a los recursos del sistema
 - ▶ Un **método de análisis** que permite calcular el **comportamiento temporal del sistema**
 - ▶ Para comprobar que los requisitos están garantizados en todos los casos
 - ▶ Se estudia siempre el peor caso
 - ▶ Es necesario conocer la duración de las tareas en el peor caso

* *Tarea = Actividad = Hilo*



Algoritmo: Planificación por prioridades fijas expulsivas

- ▶ Las prioridades son un mecanismo elemental para planificar la ejecución de un conjunto de tareas.
- ▶ La prioridad es un atributo de las tareas normalmente ligado a su importancia relativa en el conjunto de tareas.
- ▶ En STR la planificación expulsiva por prioridades fijas es la más popular:
 - ▶ El comportamiento temporal es más fácil de entender y predecir
 - ▶ El tratamiento ante sobrecargas es fácil de prever
 - ▶ Existen técnicas analíticas completas
 - ▶ Requerido por estándares de sistemas operativos y lenguajes concurrentes



Análisis de STR: Modelo de tareas

- ▶ Consideraremos inicialmente un modelo de tareas simple:
 - ▶ Conjunto de tareas estático, $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$
 - ▶ Todas las tareas son periódicas, con **periodo** T_i
 - ▶ Las tareas no cooperan entre sí (son independientes)
 - ▶ Se conoce el **tiempo de ejecución máximo** de cada tarea C_i
 - ▶ Cada tarea tiene un **plazo** $D_i \leq T_i$
- ▶ Respecto a la ejecución de las tareas, asumimos que :
 - ▶ La política de planificación es expulsiva por prioridades.
 - ▶ La máquina está dedicada a la aplicación
 - ▶ Los cambios de contexto tienen coste cero.
 - ▶ Las tareas no pueden suspenderse voluntariamente. Una vez una tarea está preparada, no puede demorar su ejecución.



Actividad: Cronograma de tareas periódicas

- ▶ Realice el cronograma de ejecución del siguiente conjunto de tareas periódicas:

Tarea	T_i	C_i	D_i	Pri *
τ_1	20	5	10	1
τ_2	40	10	15	2
τ_3	80	40	80	3

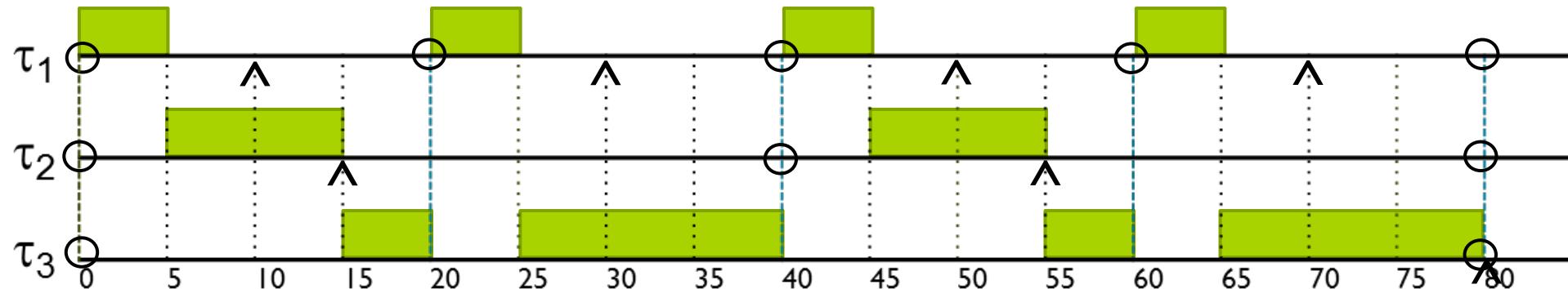
$$\text{pri}(1) > \text{pri}(2) > \text{pri}(3)$$

- ▶ Observar que el patrón de ejecución se repite cada cierto tiempo:
 - ▶ hiperperiodo = mínimo común múltiplo de los períodos de las tareas

Actividad: Cronograma de tareas periódicas

Tarea	T_i	C_i	D_i	Pri *
τ_1	20	5	10	1
τ_2	40	10	15	2
τ_3	80	40	80	3

- instante de activación
- ∧ plazo máximo de ejecución (deadline)



$$pri(1) > pri(2) > pri(3)$$

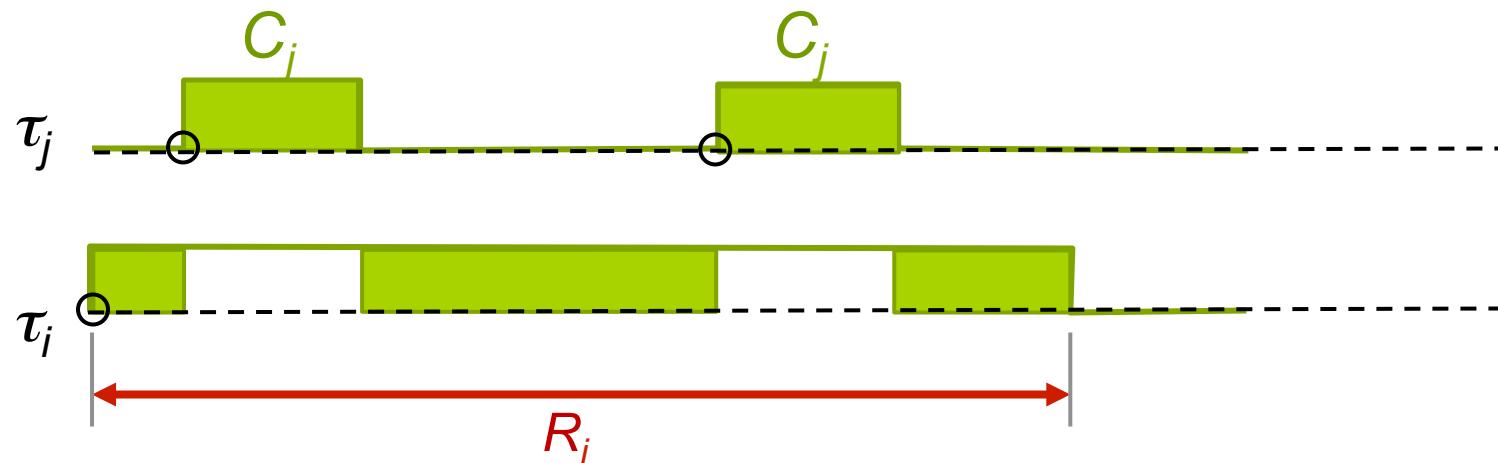


Análisis de STR: Principios básicos

- ▶ Dos conceptos ayudan a construir la situación de peor caso en el caso de tareas periódicas independientes:
 - ▶ **Instante crítico.** El tiempo de respuesta de peor caso de todas las tareas se obtiene si las activamos todas a la vez
 - ▶ **Basta comprobar el primer plazo.** Después de un instante crítico, si una tarea cumple su primer plazo cumplirá todos los plazos posteriores
- ▶ Basándose en estos conceptos se obtiene el test de planificabilidad:
 - ▶ **Test exacto, o de tiempos de respuesta**

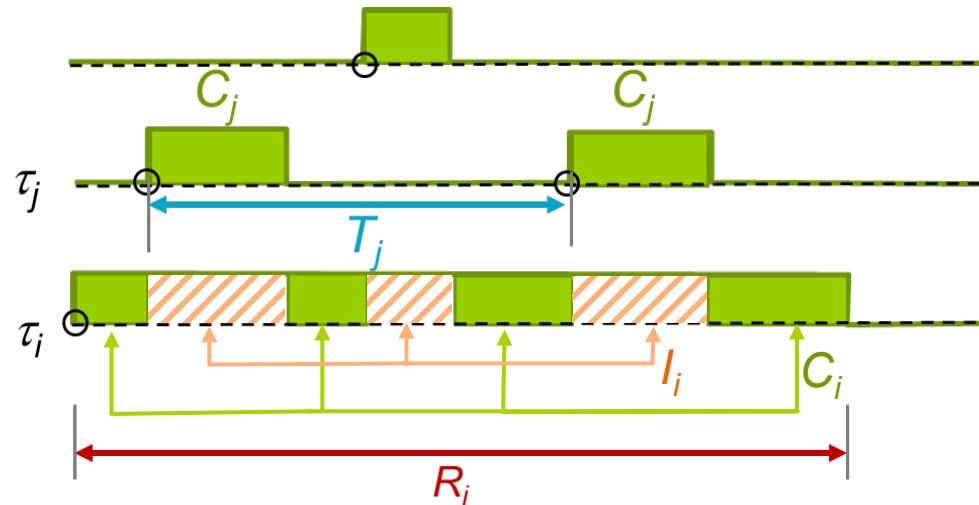
Ecuación del tiempo de respuesta

- El tiempo de respuesta de una tarea es la suma de su tiempo de cálculo más la interferencia que sufre por la ejecución de tareas más prioritarias



$$R_i = C_i + I_i$$

Cálculo de la interferencia máxima I_i



$\left\lceil \frac{R_i}{T_j} \right\rceil$: número de activaciones de la tarea j en R_i

$\left\lceil x \right\rceil$: $\text{techo}(x)$, menor número entero $\geq x$

- Para una tarea de prioridad superior $I_i^j = \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j$
- Para todas las tareas de prioridad superior

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

$hp(i)$: $high\ priority(i)$, conjunto de tareas más prioritarias que i



Test de planificabilidad

- ▶ En un sistema de tiempo real con n tareas independientes, planificadas mediante un esquema expansivo de prioridades fijas, y para una asignación concreta de prioridades a las tareas, **se cumplirán todos los plazos de respuesta si, y sólo si,**

$$\forall i, \quad 1 \leq i \leq n, \quad R_i \leq D_i$$

donde

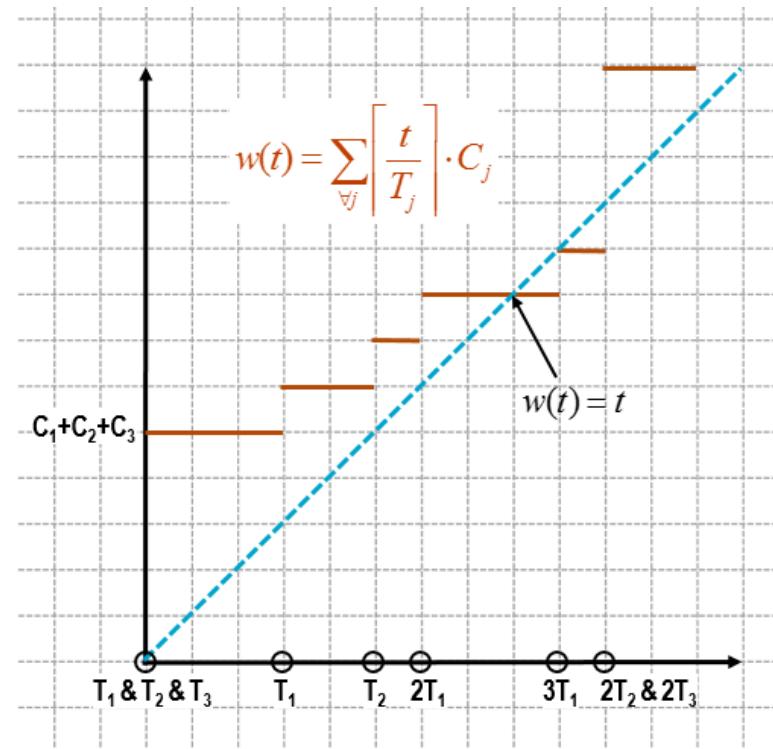
$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j$$

- ▶ R_i es la solución mínima de la ecuación

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

- ▶ La ecuación no es continua ni lineal
- ▶ No se puede resolver analíticamente

Tarea	T_i	C_i
τ_1	3	1
τ_2	5	1
τ_3	10	3





Cálculo del tiempo de respuesta

- ▶ Se puede resolver mediante la relación de recurrencia

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

- ▶ Un valor inicial aceptable es

$$w_i^0 = C_i + \sum_{j \in hp(i)} C_j$$

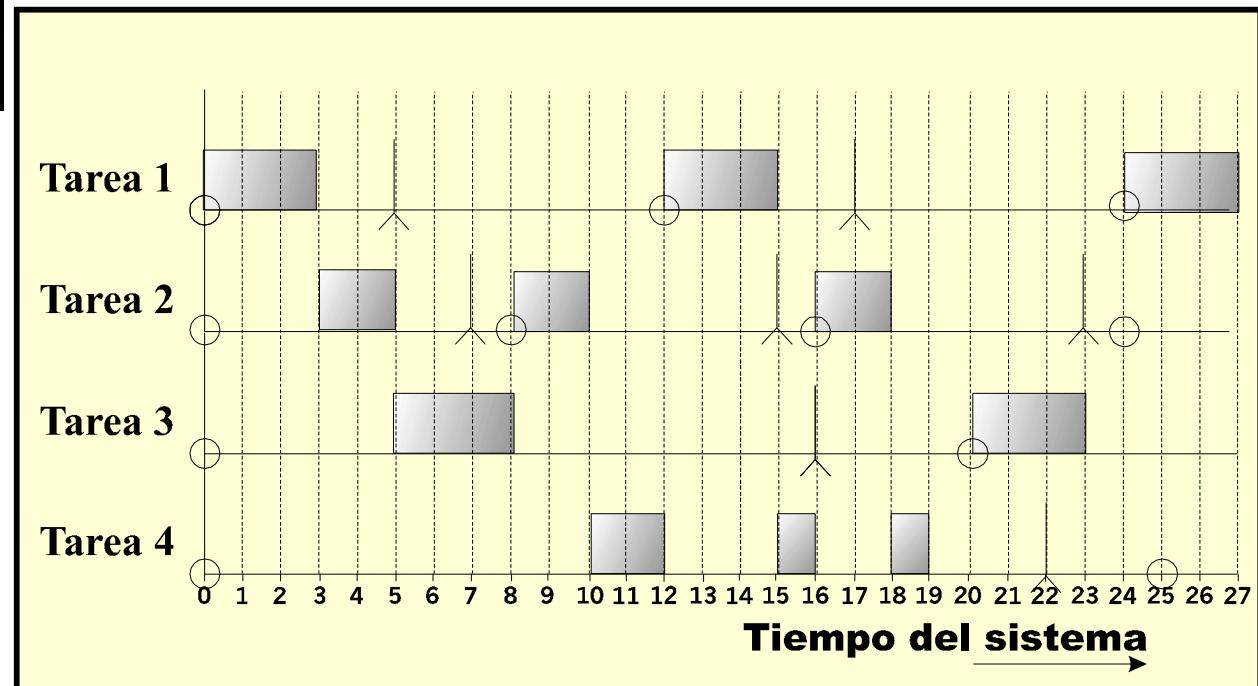
- ▶ Se termina cuando

- ▶ $w^{n+1} = w^n$ o bien
- ▶ $w^{n+1} > D_i$ (no se cumple el plazo)

Ejemplo de cálculo de los tiempos de respuesta

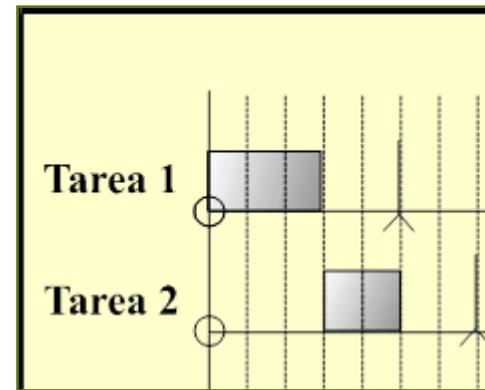
- Calcular los R_i para el siguiente conjunto de tareas:

Tarea	T_i	C_i	D_i	Prio	R_i
τ_1	12	3	5	1	
τ_2	8	2	7	2	
τ_3	20	3	16	3	
τ_4	25	4	22	4	



Ejemplo de cálculo de los tiempos de respuesta

Tarea	T_i	C_i	D_i	Prio	R_i
τ_1	12	3	5	1	3
τ_2	8	2	7	2	5
τ_3	20	3	16	3	
τ_4	25	4	22	4	



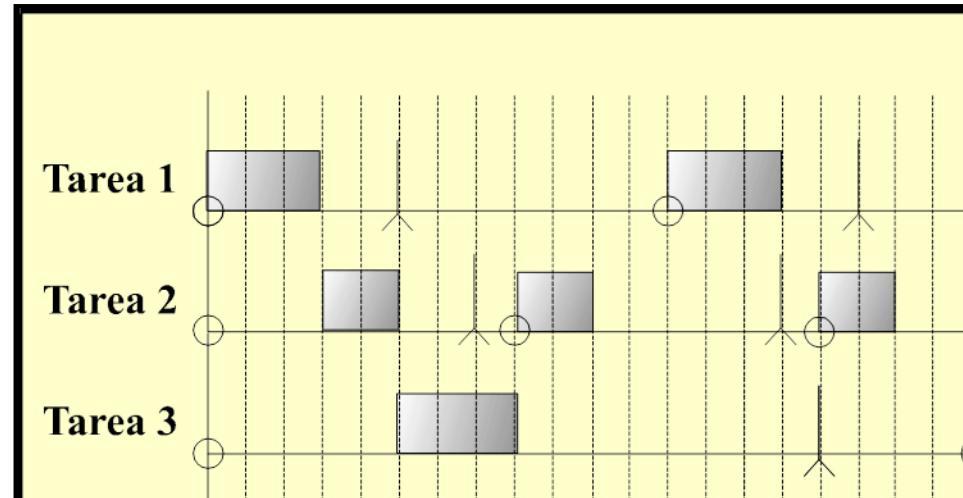
$$\tau_1 : w_1^0 = 3$$

$$\tau_2 : w_2^0 = 2 + 3 = 5$$

$$w_2^1 = 2 + \left\lceil \frac{5}{12} \right\rceil \cdot 3 = 5$$

Ejemplo de cálculo de los tiempos de respuesta

Tarea	T_i	C_i	D_i	Prio	R_i
τ_1	12	3	5	1	3
τ_2	8	2	7	2	5
τ_3	20	3	16	3	8
τ_4	25	4	22	4	



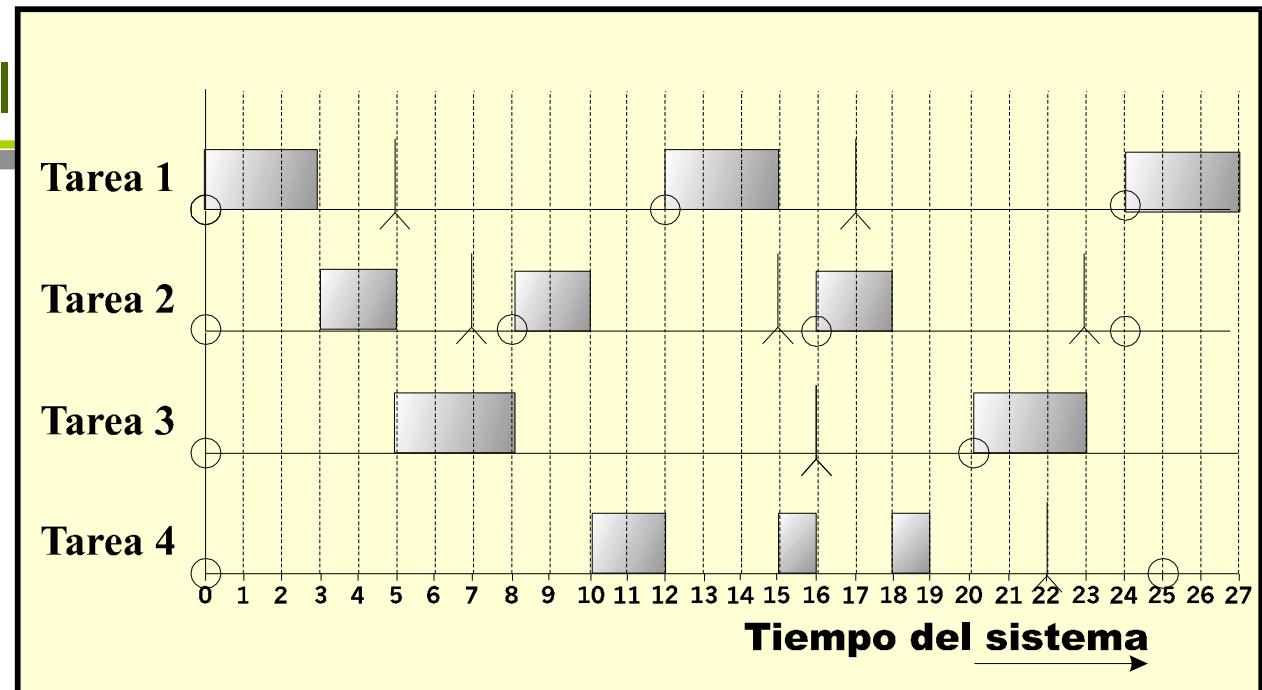
$$\tau_3 : w_3^0 = 2 + 3 + 3 = 8$$

$$w_3^1 = 3 + \left\lceil \frac{8}{12} \right\rceil \cdot 3 + \left\lceil \frac{8}{8} \right\rceil \cdot 2 = 8$$

Ejemplo de cálculo

Tarea	T_i	C_i	D_i	Prio	R_i
τ_1	12	3	5	1	3
τ_2	8	2	7	2	5
τ_3	20	3	16	3	8
τ_4	25	4	22	4	19

$$\tau_4 : w_4^0 = 2 + 3 + 3 + 4 = 12$$



$$w_4^1 = 4 + \left\lceil \frac{12}{12} \right\rceil \cdot 3 + \left\lceil \frac{12}{8} \right\rceil \cdot 2 + \left\lceil \frac{12}{20} \right\rceil \cdot 3 = 14$$

$$w_4^2 = 4 + \left\lceil \frac{14}{12} \right\rceil \cdot 3 + \left\lceil \frac{14}{8} \right\rceil \cdot 2 + \left\lceil \frac{14}{20} \right\rceil \cdot 3 = 17$$

$$w_4^3 = 4 + \left\lceil \frac{17}{12} \right\rceil \cdot 3 + \left\lceil \frac{17}{8} \right\rceil \cdot 2 + \left\lceil \frac{17}{20} \right\rceil \cdot 3 = 19$$

$$w_4^4 = 4 + \left\lceil \frac{19}{12} \right\rceil \cdot 3 + \left\lceil \frac{19}{8} \right\rceil \cdot 2 + \left\lceil \frac{19}{20} \right\rceil \cdot 3 = 19$$



Actividad sobre el test de planificabilidad

- ▶ Determinar la planificabilidad del siguiente conjunto de tareas, utilizando el test de los tiempos de respuesta, en los siguientes casos:
 - Usando la asignación de prioridades: $\tau_1 > \tau_2 > \tau_3$
 - Usando la asignación de prioridades: $\tau_1 < \tau_2 < \tau_3$

<i>Tarea</i>	<i>T_i</i>	<i>C_i</i>	<i>D_i</i>
τ_1	4	1	4
τ_2	5	2	5
τ_3	20	3	10



Solución caso a)

Tarea	T_i	C_i	D_i
τ_1	4	1	4
τ_2	5	2	5
τ_3	20	3	10

$$\tau_1 : w_1^0 = 1$$

$$\tau_2 : w_2^0 = 2 + 1 = 3$$

$$w_2^1 = 2 + \left\lceil \frac{3}{4} \right\rceil \cdot 1 = 3$$

$$\tau_3 : w_3^0 = 3 + 1 + 2 = 6$$

$$w_3^1 = 3 + \left\lceil \frac{6}{4} \right\rceil \cdot 1 + \left\lceil \frac{6}{5} \right\rceil \cdot 2 = 9$$

$$w_3^2 = 3 + \left\lceil \frac{9}{4} \right\rceil \cdot 1 + \left\lceil \frac{9}{5} \right\rceil \cdot 2 = 10$$

$$w_3^3 = 3 + \left\lceil \frac{10}{4} \right\rceil \cdot 1 + \left\lceil \frac{10}{5} \right\rceil \cdot 2 = 10$$

$$\begin{aligned} R_1 &= 1 &< D_1 \\ R_2 &= 3 &< D_2 \\ R_3 &= 10 \leq D_3 \end{aligned}$$

Todos los plazos están garantizados



Contenido

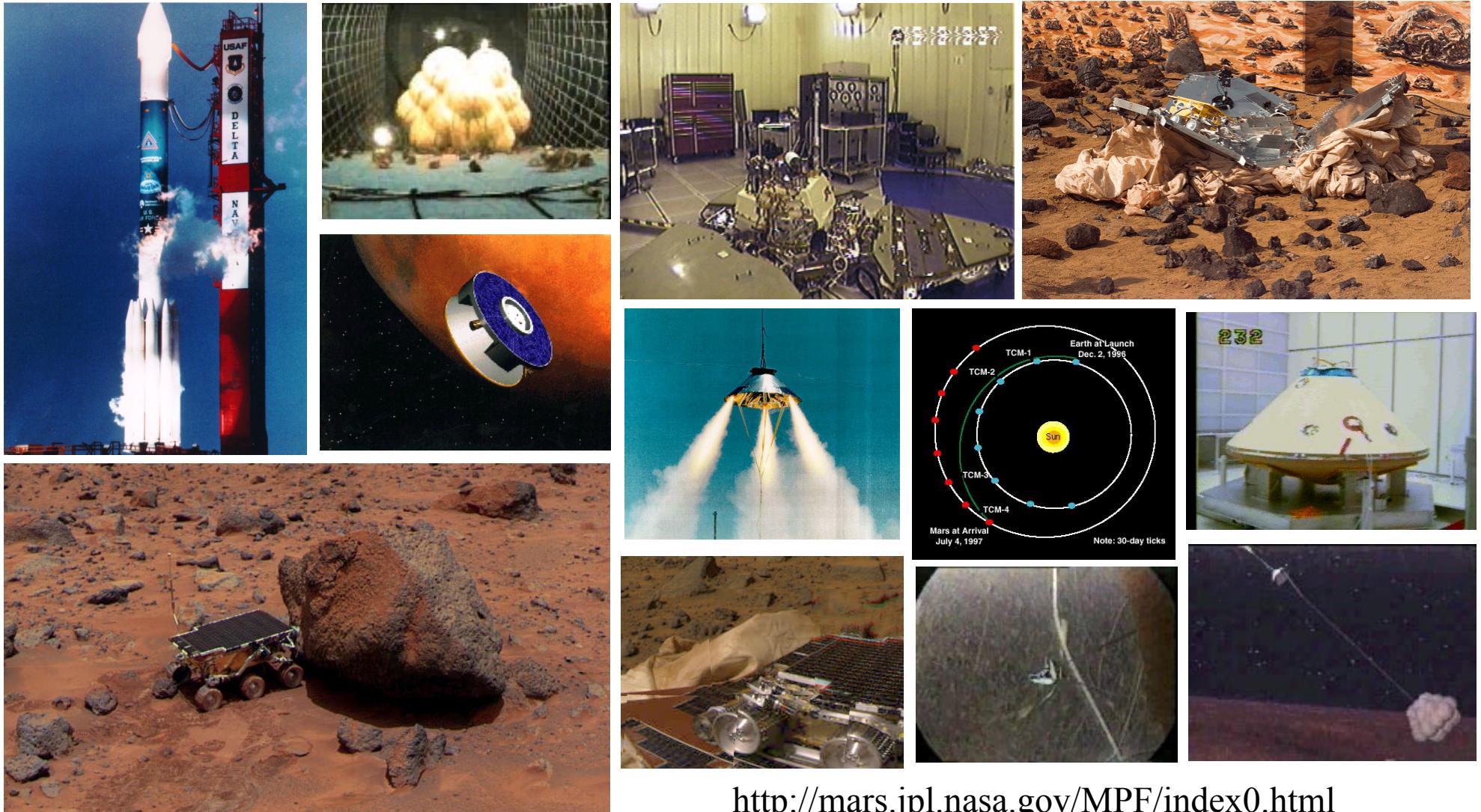
- ▶ Introducción a los STR
 - ▶ Definición de STR
 - ▶ Características de los STR
 - ▶ Planificación de tareas en STR
 - ▶ Análisis de STR
- ▶ Sincronización de tareas
 - ▶ El problema de la inversión de prioridades
 - ▶ Protocolo del Techo de Prioridad Inmediato
 - ▶ Tiempo de respuesta con factores de bloqueo



La sincronización entre tareas

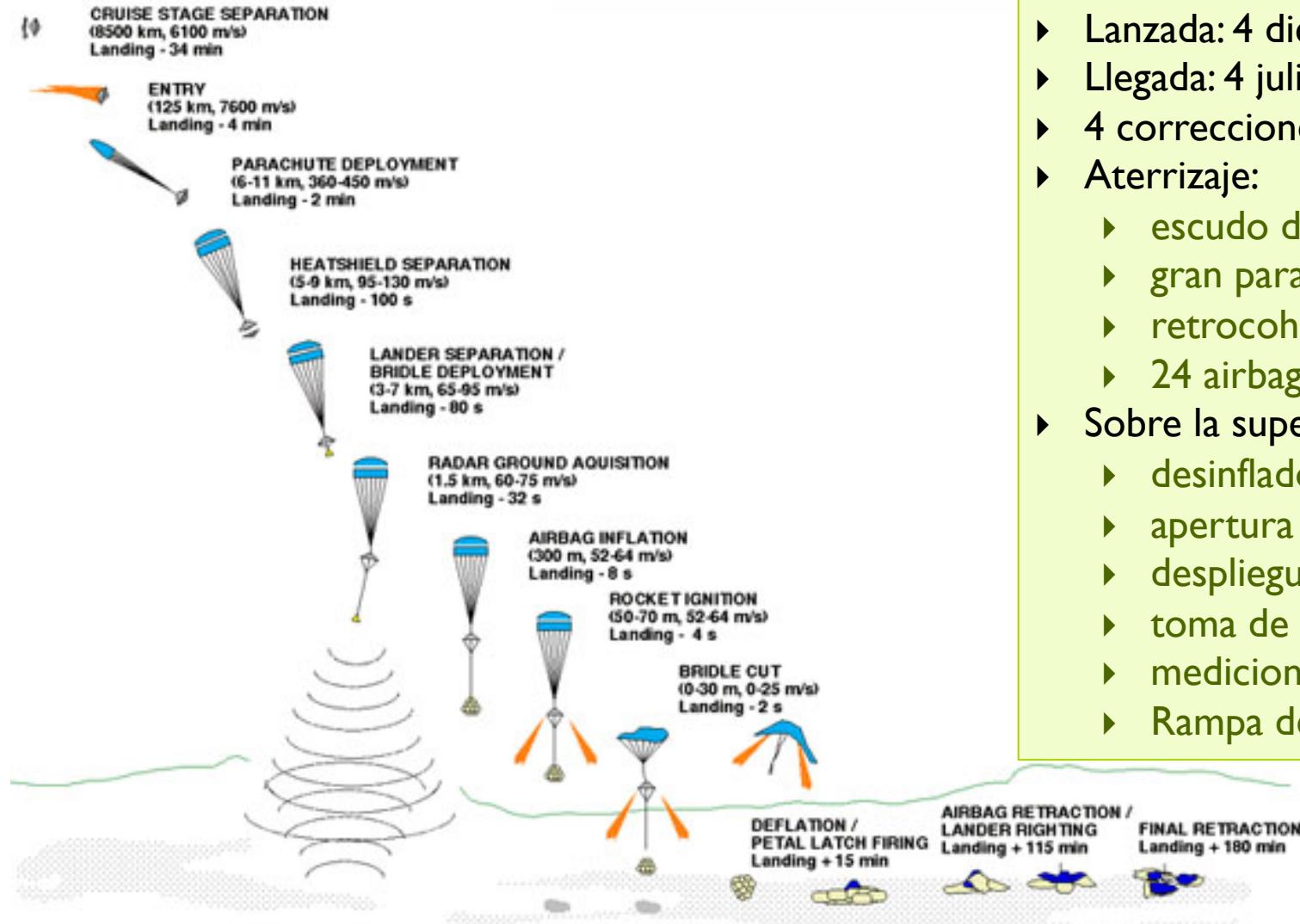
- ▶ Una de las restricciones más importantes del test de planificabilidad anterior es la de suponer que las tareas son **independientes**.
- ▶ En la mayoría de los sistemas de interés práctico las tareas necesitarán comunicarse.
 - ▶ Cuando la comunicación se realice mediante la compartición de datos comunes, ➔ la **sincronización** será necesaria para evitar las condiciones de carrera.
 - ▶ Asumiremos que las tareas sólo necesitan **exclusión mutua** en el acceso a sus secciones críticas, y que las guardan utilizando alguna primitiva de sincronización equivalente a los locks:
 - ▶ P(S) es equivalente a cerrar_lock (S);
 - ▶ V(S) es equivalente a abrir_lock(S);
- ▶ Sin embargo, en sistemas de tiempo real esta solución no es válida, ya que puede producir una situación denominada **inversión de prioridades**, que invalida el test de planificabilidad.

La misión Mars Pathfinder, Julio 1997



<http://mars.jpl.nasa.gov/MPF/index0.html>

Mars Pathfinder



- ▶ Lanzada: 4 diciembre 1996
- ▶ Llegada: 4 julio 1997
- ▶ 4 correcciones de vuelo
- ▶ Aterrizaje:
 - ▶ escudo de protección térmica
 - ▶ gran paracaídas
 - ▶ retrocohetes
 - ▶ 24 airbags
- ▶ Sobre la superficie:
 - ▶ desinflado de airbags
 - ▶ apertura del Lander
 - ▶ despliegue paneles solares
 - ▶ toma de imágenes
 - ▶ mediciones meteorológicas
 - ▶ Rampa de descenso del Sojourner

http://mars.jpl.nasa.gov/MPF/mpf/fact_sheet.html

REBUCHAR



Motivación: Tribute to the Mars Pathfinder Team

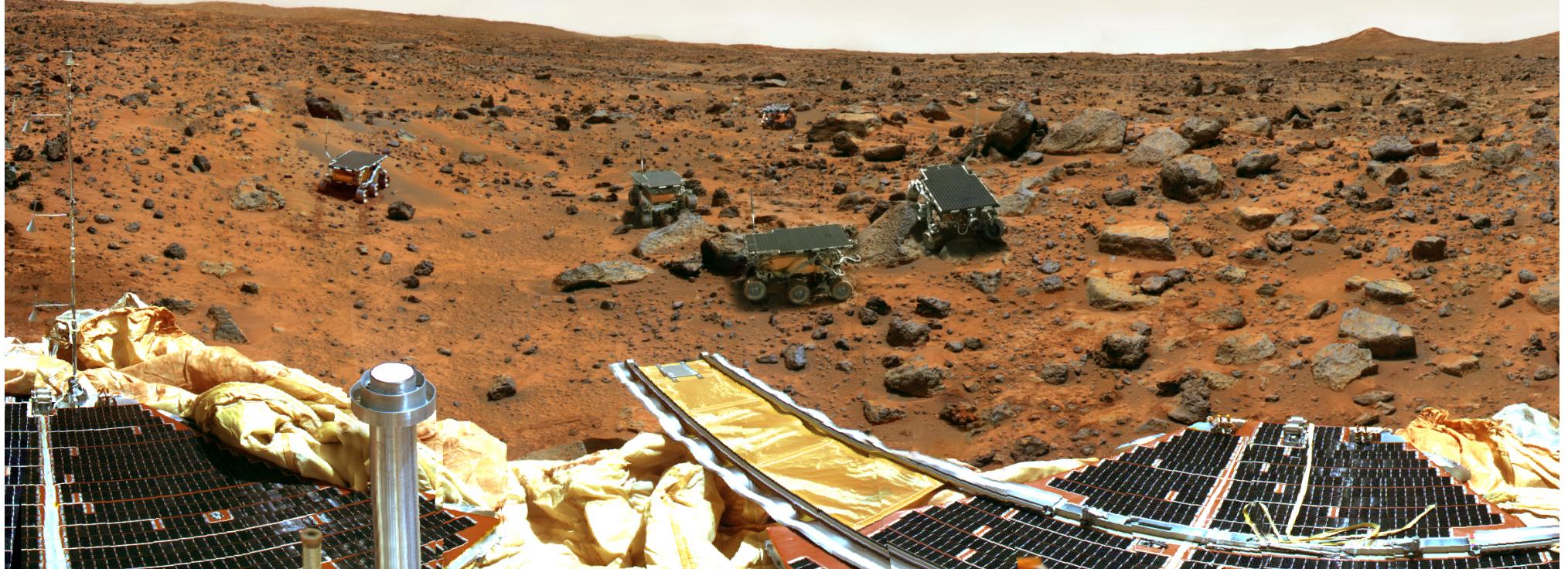


<https://www.youtube.com/watch?v=6jspO2CsFNA>



Incidente en el “Mars Pathfinder”

http://research.microsoft.com/~mbj/Mars_Pathfinder/Mars_Pathfinder.html



The Mars Pathfinder mission was widely proclaimed as "flawless" in the early days after its July 4th, 1997 landing on the Martian surface.

But a few days into the mission, not long after Pathfinder started gathering meteorological data, the spacecraft began experiencing total **system resets**, each resulting in losses of data. The press reported these failures in terms such as "**software glitches**" and "the computer was trying to do too many things at once".



Incidente en el “Mars Pathfinder”

http://research.microsoft.com/~mbj/Mars_Pathfinder/Mars_Pathfinder.html

The New York Times

www.nytimes.com

July 15, 1997

Mars Craft Again Halts Transmission

PASADENA, Calif., July 14— The computer aboard the Mars Pathfinder overloaded and reset itself early today for the second time in just over three days, interrupting the transmission of a full-color panoramic scene.

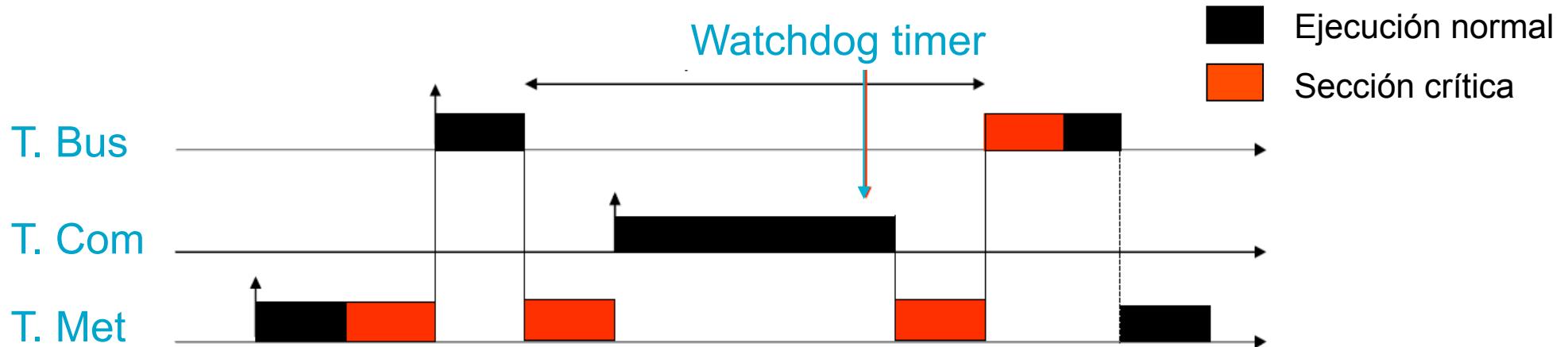
The mishap delayed chemical analysis of a tubby rock named Yogi, but no information was lost, and controllers will be able to resume transmission where it was left off, officials said.

Mary Beth Murrill, a spokeswoman for NASA's Jet Propulsion Laboratory, said transmission of the panoramic shot took "a lot of processing power." She likened the data overload to what happens with a personal computer "when we ask it to do too many things at once."

The project manager, Brian Muirhead, said that to prevent a recurrence, controllers would schedule activities one after another, instead of at the same time. It was the second time the Pathfinder's computer had reset itself while trying to carry out several activities at once.

data, the spacecraft began experiencing total **system resets**, each resulting in losses of data. The press reported these failures in terms such as "**software glitches**" and "the computer was trying to do too many things at once".

Incidente en el “Mars Pathfinder”



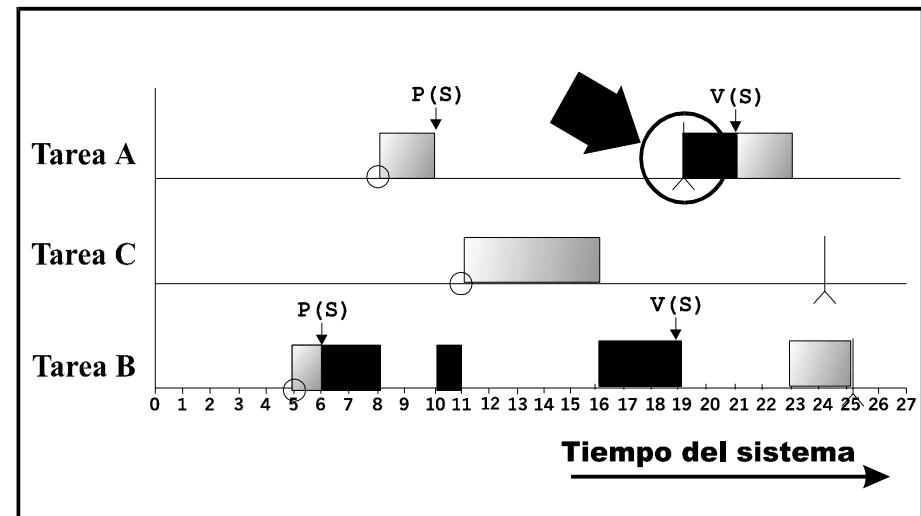
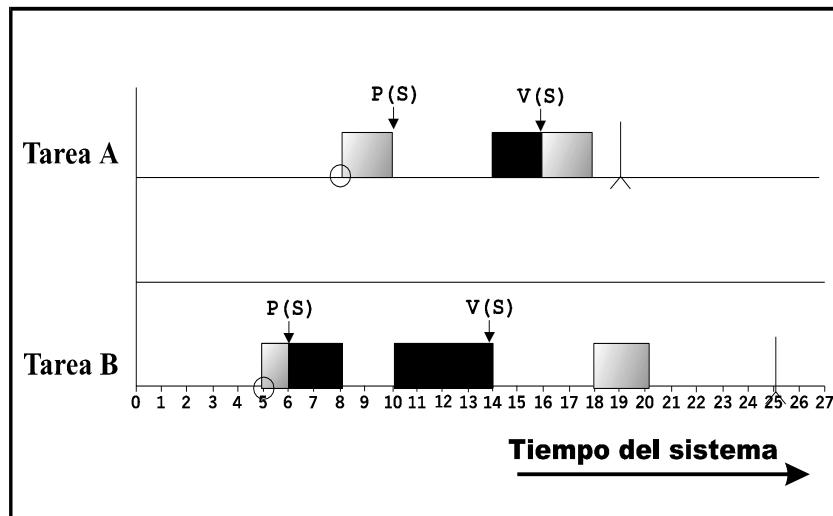
“ Most of the time this combination worked fine. However, very infrequently it was possible for an interrupt to occur that caused the (medium priority) communications task to be scheduled during the short interval while the (high priority) information bus thread was blocked waiting for the (low priority) meteorological data thread. In this case, the long-running **communications task**, having higher priority than the **meteorological task**, would prevent it from running, consequently preventing the blocked **information bus task** from running.

After some time had passed, a **watchdog timer** would go off, notice that the data bus task had not been executed for some time, conclude that something had gone drastically wrong, and initiate a total **system reset**.

This scenario is a classic case of **priority inversion**. ”

La inversión de prioridades

- ▶ La inversión de prioridades ocurre cuando una tarea menos prioritaria se ejecuta por delante de una más prioritaria.
- ▶ El tiempo total de **bloqueo** debido a una tarea menos prioritaria **no está acotado**, al poder intervenir tareas de prioridad intermedia.
- ▶ Ejemplo :





Protocolos de sincronización

- ▶ La inversión de prioridades no puede evitarse. Los bloqueos comprometen la planificabilidad del sistema.
 - ▶ Se trata de conseguir que el bloqueo esté acotado, se produzca el menor número de veces posible y que sea medible.
- ▶ Los protocolos de sincronización de tiempo real **evitan la inversión de prioridad no acotada**
- ▶ Los más importantes:
 - ▶ Herencia de prioridad
 - ▶ Techo de prioridad inmediato (o protección por prioridad)
- ▶ Ambos hacen que la inversión de prioridad, o retraso que una tarea sufre a causa de tareas de prioridad inferior:
 - ▶ sea función de la duración de una o varias secciones críticas
 - ▶ no sea función de la duración de tareas completas



Contenido

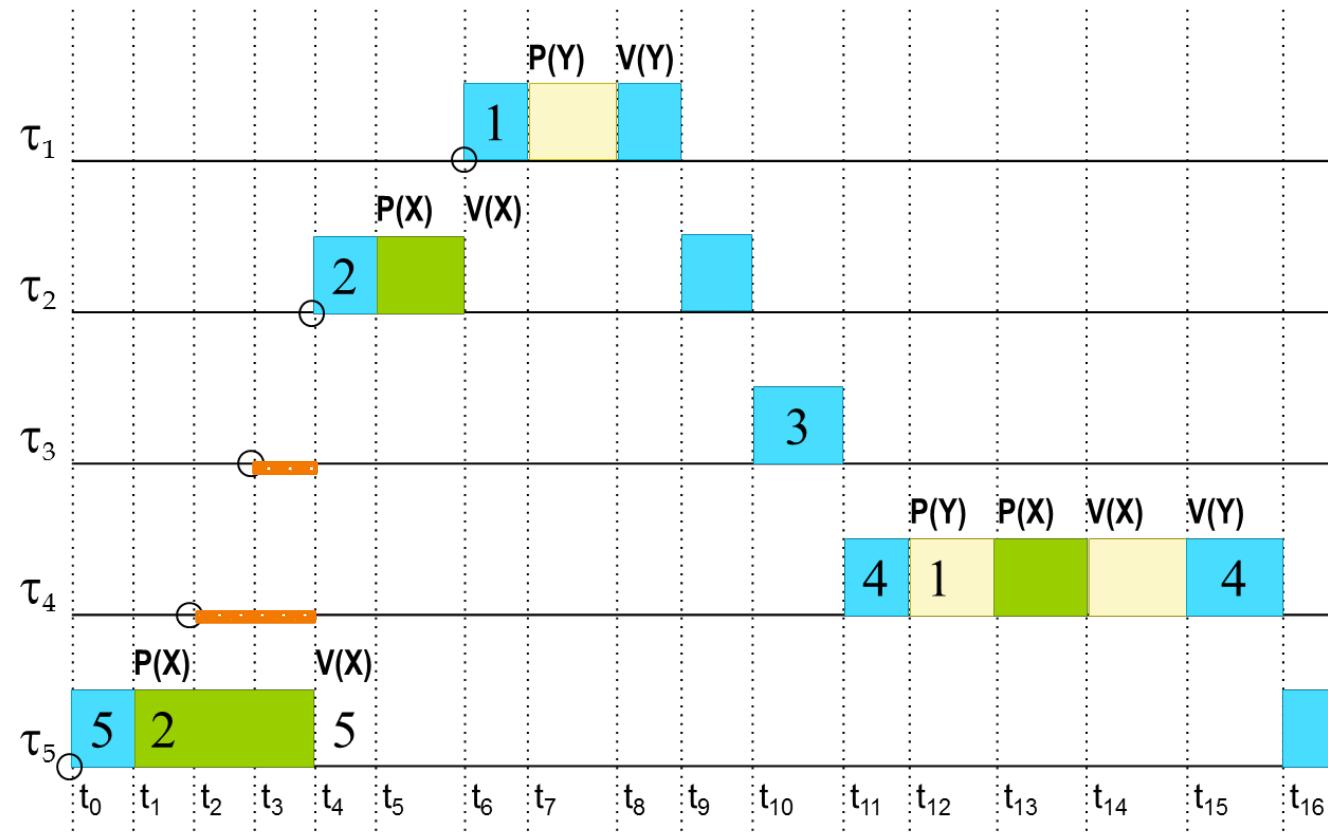
- ▶ **Introducción a los STR**
 - ▶ Definición de STR
 - ▶ Características de los STR
 - ▶ Planificación de tareas en STR
 - ▶ Análisis de STR
- ▶ **Sincronización de tareas**
 - ▶ El problema de la inversión de prioridades
 - ▶ Protocolo del Techo de Prioridad Inmediato
 - ▶ Tiempo de respuesta con factores de bloqueo



Protocolo del techo de prioridad inmediato

- ▶ Concepto de **techo de prioridad**: A cada semáforo le asociaremos un techo, equivalente a la mayor prioridad de las tareas que lo pueden usar.
- ▶ Con este protocolo, una tarea que accede a un semáforo hereda inmediatamente el techo de prioridad del semáforo.
 - ▶ La sección crítica se ejecuta con la prioridad del techo del semáforo que la guarda.
- ▶ Propiedades:
 - ▶ Cada tarea se puede bloquear una vez como máximo, en cada ciclo. Además si una tarea se bloquea, lo hace al principio del ciclo.
 - ▶ No puede haber interbloqueos.
- ▶ También es conocido por otros nombres
 - ▶ priority ceiling emulation (RT Java)
 - ▶ priority protect protocol (POSIX)
 - ▶ ceiling locking protocol (Ada)

Protocolo del techo de prioridad inmediato



$$\begin{aligned} \text{techo}(X) &= 2 \\ \text{techo}(Y) &= 1 \end{aligned}$$



Cálculo de los factores de bloqueo máximos

- ▶ Sea B_i el factor de bloqueo: tiempo máximo de retraso que sufre la tarea i.
- ▶ Para el protocolo de techo de prioridad inmediato:
 - ▶ B_i es la duración máxima de todas las secciones críticas cuyo techo es \geq prioridad de la tarea i, que son utilizadas por tareas de prioridad inferior
 - ▶ para la tarea de menor prioridad: $B_n = 0$
 - ▶ para el resto de tareas:
$$B_i = \max_{\{k,s \mid k \in lp(i) \wedge s \in usa(k) \wedge techo(s) \geq prio(i)\}} C_{k,s}$$
 - ▶ se consideran las tareas k de prioridad inferior a i, y entonces se miran los semáforos s que pueden cerrar, y se selecciona aquellos cuyo techo tenga prioridad $\geq i$, por último se miran los tiempos de cómputo de las secciones que guardan esos semáforos $C_{k,s}$ y se selecciona la de mayor duración.

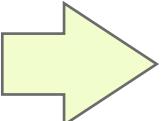


Incorporación del factor de bloqueo al test

- ▶ El **test inicial** en base a prioridades fijas, venía de resolver una relación de recurrencia como la siguiente :

$$R_i = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \quad \text{y averiguar si para cada tarea se cumplía que } R_i \leq D_i.$$

- ▶ Este test está basado en el peor caso de interferencia que las tareas más prioritarias ejercen sobre la tarea i .

Tenemos que incluir el **efecto del bloqueo** de las menos prioritarias

$$R_i = C_i + B_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$



Actividad: Test de planificabilidad con factores de bloqueo

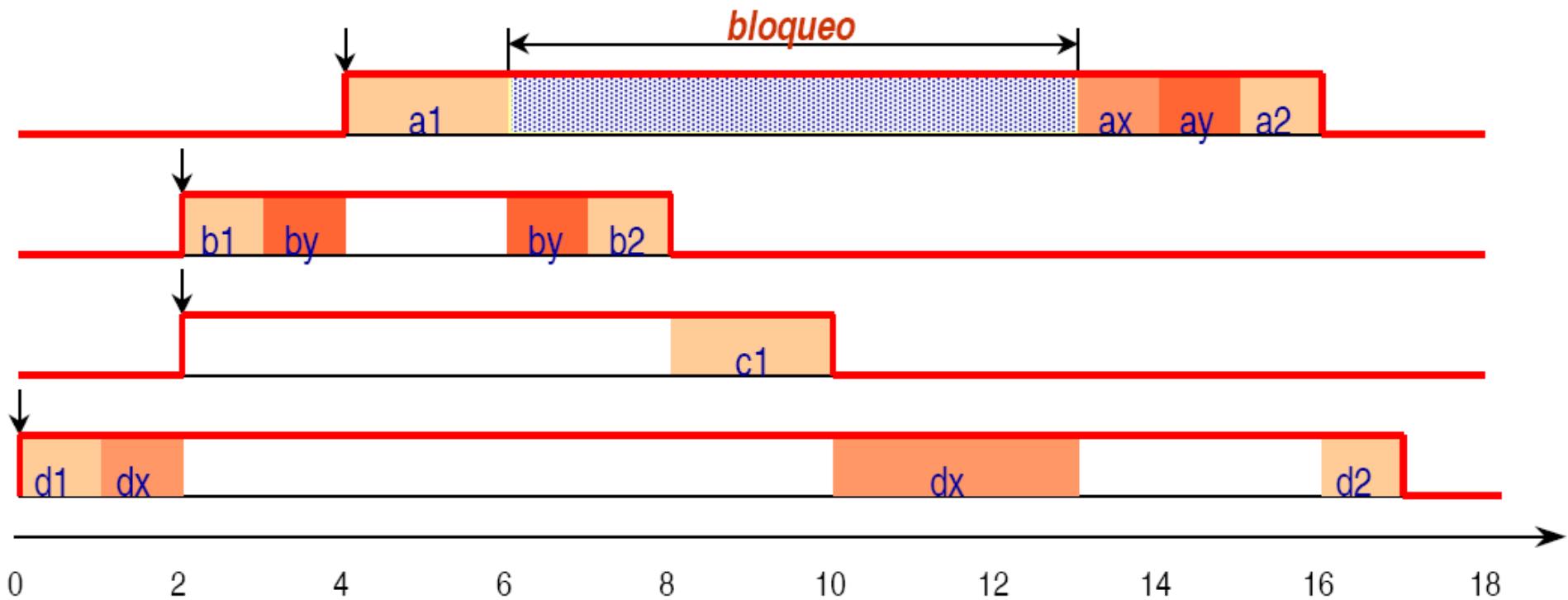
- ▶ Sea un sistema con 4 tareas y 2 semáforos que guardan las secciones críticas según tablas adjuntas:

Tarea	Acciones	Inst. de activación
τ_1	a1; ax; ay; a2	4
τ_2	b1; by; b2	2
τ_3	c1	2
τ_4	d1; dx; d2	0

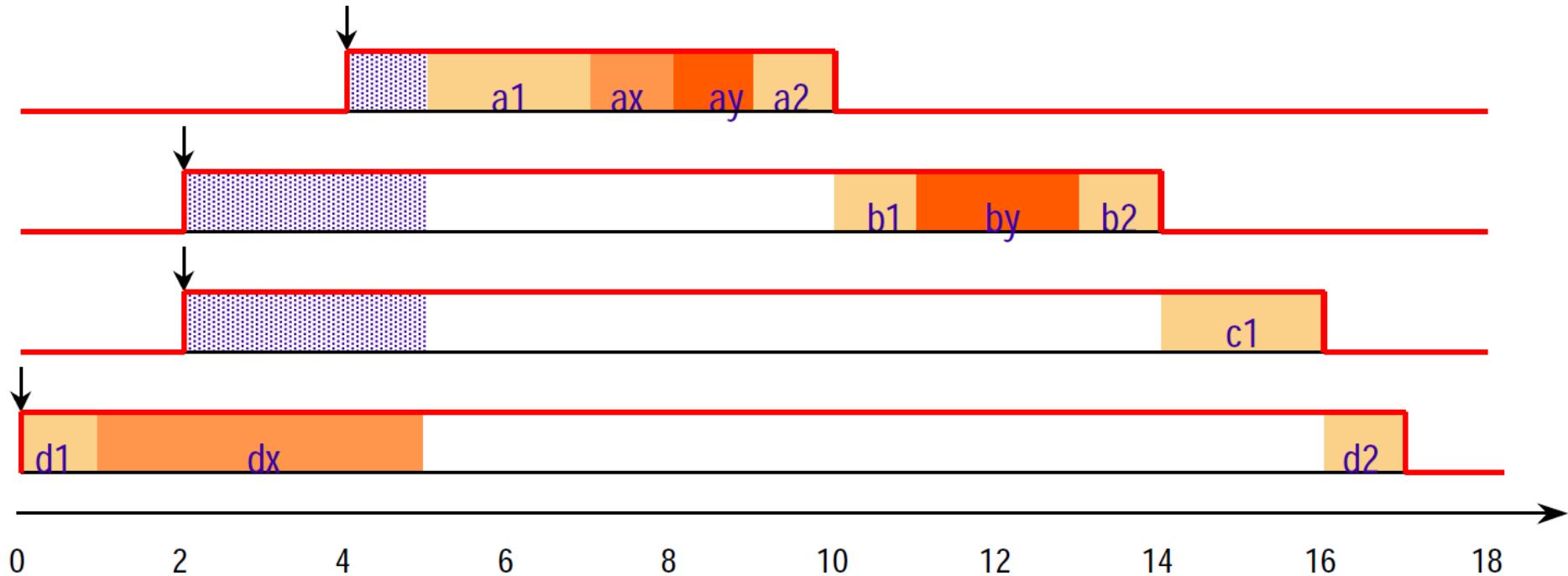
Acción	C	usa semáforo
a1	2	
ax	1	X
ay	1	Y
a2	1	
b1	1	
by	2	Y
b2	1	
c1	2	
d1	1	
dx	4	X
d2	1	

- ▶ Desarrollar los cronogramas para los siguientes casos:
 - Prioridades fijas expulsivas con $pri(1) > pri(2) > pri(3) > pri(4)$, y semáforos normales
 - Semáforos que utilizan el protocolo del techo de prioridad inmediato
- ▶ Calcular en el caso b), para cada tarea su factor de bloqueo máximo

Solución: a) inversión de prioridad



Solución: b) Techo de prioridad inmediato



Solución: cálculo de factores de bloqueo

- Para el protocolo de techo de prioridad inmediato:

 - para la tarea de menor prioridad: $B_n = 0$

 - para el resto de tareas:

$$B_i = \max_{\{k,s \mid k \in lp(i) \wedge s \in usa(k) \wedge techo(s) \geq prio(i)\}} C_{k,s}$$

 - se consideran las tareas k de prioridad inferior a i , y entonces se miran los semáforos s que pueden cerrar, y se selecciona aquellos cuyo techo tenga prioridad $\geq i$, por último se miran los tiempos de cómputo de las secciones que guardan esos semáforos $C_{k,s}$ y se selecciona la de mayor duración.

Acción	C	usa semáforo
a1	2	
ax	1	X
ay	1	Y
a2	1	
b1	1	
by	2	Y
b2	1	
c1	2	
d1	1	
dx	4	X
d2	1	



Tarea	B_i con Techo de prioridad inmediato
τ_1	4
τ_2	4
τ_3	4
τ_4	0



Resultados de aprendizaje de la Unidad Didáctica

- ▶ Al finalizar esta unidad, el alumno deberá ser capaz de:
 - ▶ Caracterizar adecuadamente los sistemas de tiempo real.
 - ▶ Identificar las dificultades que conlleva la planificación de tareas en sistemas de tiempo real.
 - ▶ Aplicar el análisis de viabilidad en diferentes supuestos.
 - ▶ Identificar el problema de inversión de prioridades y las consecuencias que puede tener.
 - ▶ Implantar el protocolo de techo de prioridad inmediato en aquellos sistemas que lo requieran.