



Sistemas Embebidos

FreeRTOS – Planificación de Tareas



Laboratorio de
Sistemas Embebidos

<http://laboratorios.fi.uba.ar/lse/>

seminario-embebidos@googlegroups.com

66.48 & 66.66 Seminario de Electrónica: Sistemas Embebidos

<http://laboratorios.fi.uba.ar/lse/seminario/>

Ingeniería en Electrónica – FI – UBA

Buenos Aires, 3 de Octubre de 2012



Tareas de Tiempo Real

Las actividades de un sistema de tiempo real se llaman tareas

Tienen varios tipos de requisitos

Funcionales: qué hacen

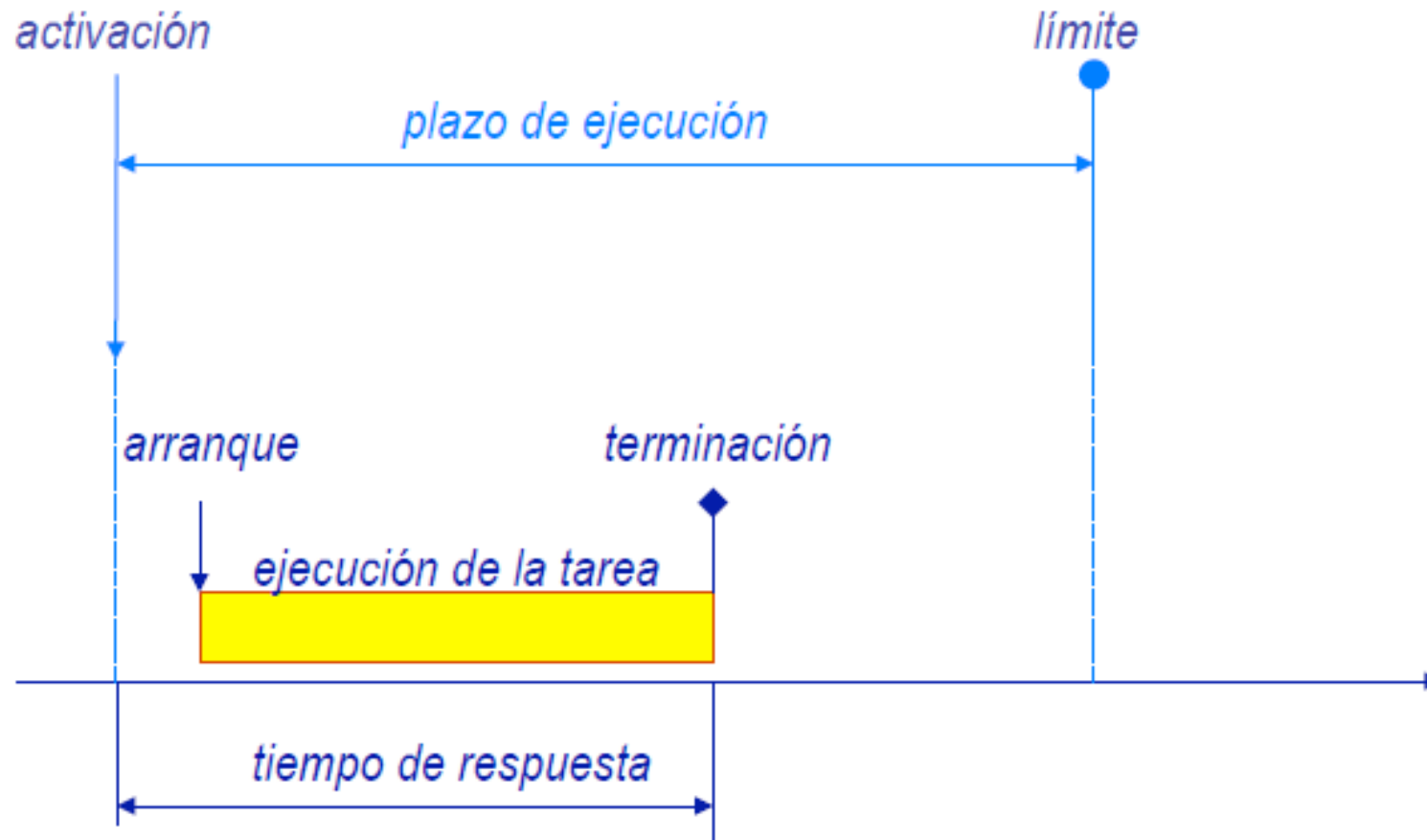
No funcionales: todos los demás

Temporales: cuándo se ejecutan, cuánto tardan, ...

Fiabilidad, seguridad

Costo, consumo de energía, etc.

Ejecución de Tarea de Tiempo Real





Atributos Temporales

Activación

Periódica: a intervalos regulares, con período T

Aperiódica: cada vez que ocurre un suceso determinado

Esporádica: separación mínima entre activaciones T
Estocástica, a rachas, irregular

Plazo de respuesta

Absoluto: tiempo límite para terminar

Relativo: intervalo desde la activación

Importante: Se trata de garantizar que la ejecución de cada tarea termina dentro de plazo



Tipos de Requisitos Temporales

Tiempo real **estricto** (hard real-time)

Todas las acciones deben terminar dentro d/plazo especificado.

Ejemplo: control de frenado

Tiempo real **flexible** (soft real-time)

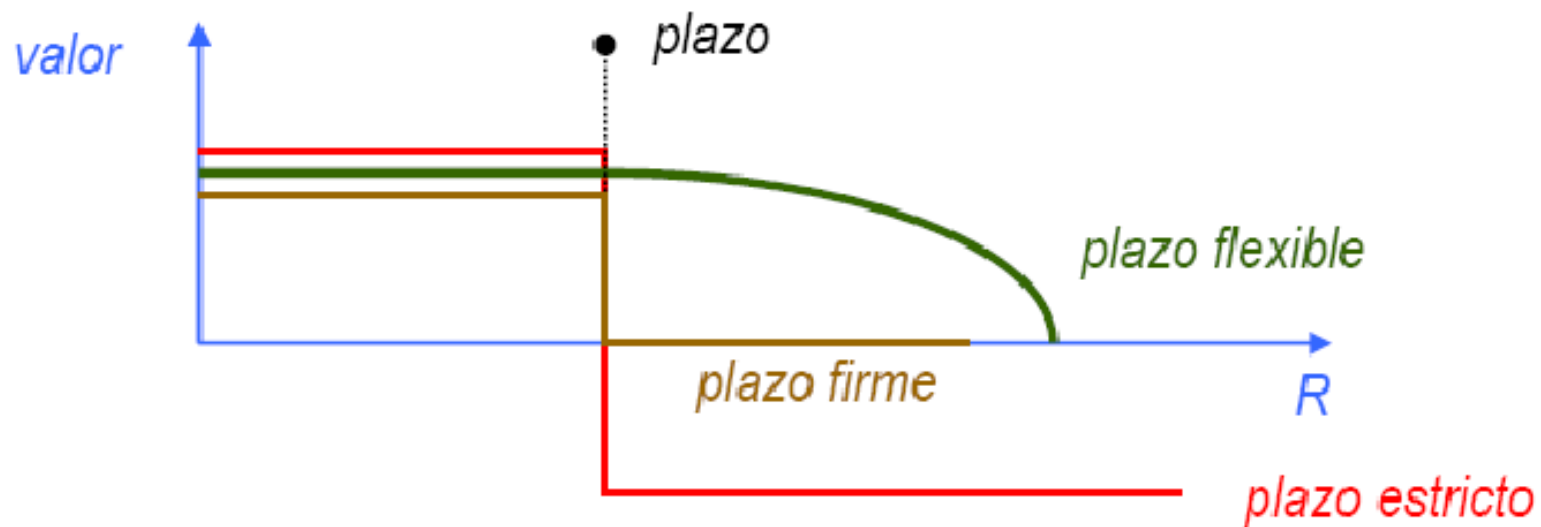
Se pueden perder plazos de vez en cuando. El valor de la respuesta decrece con el tiempo. Ej: adquisición de datos

Tiempo real **firme** (firm real-time)

Se pueden perder plazos ocasionalmente. Una respuesta tardía no tiene valor. Ejemplo: sistemas multimedia

Importante: En un mismo sistema puede haber tareas con distintos tipos de requisitos temporales

Tipos de Requisitos Temporales





Tareas de Tiempo Real

En un sistema de tiempo real se ejecutan una o más tareas

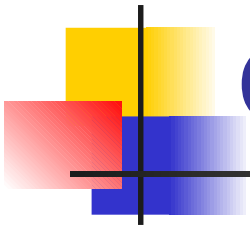
Cada tarea ejecuta una actividad de forma repetida

Cada vez que ejecuta la actividad se produce un ciclo de ejecución

Durante el ciclo de ejecución la tarea permanece activa

Cuando termina la actividad pasa a estar inactiva en espera de que comience el siguiente ciclo de ejecución



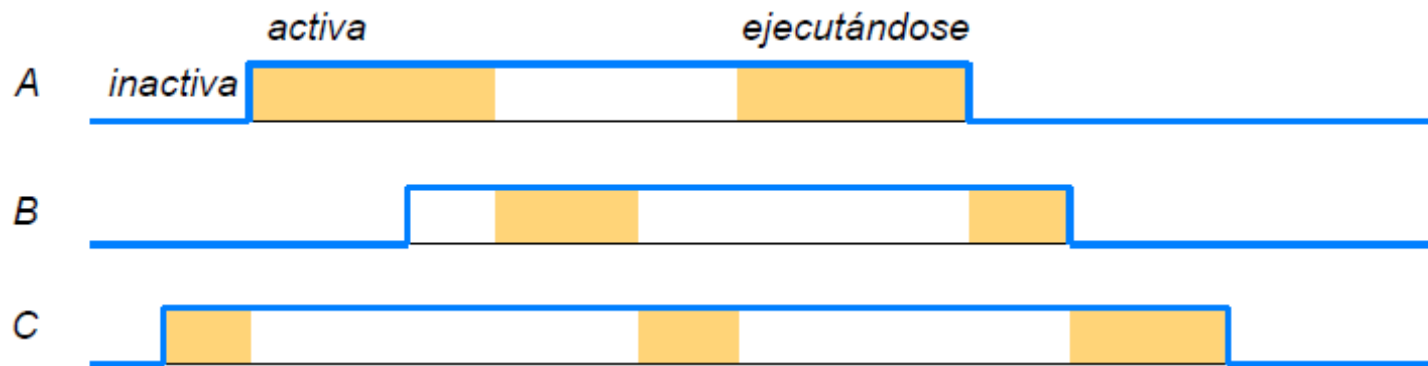


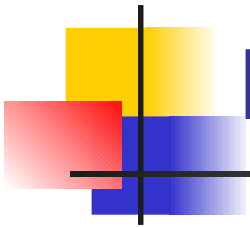
Concurrencia

Los sistemas de tiempo real controlan actividades del mundo exterior que son simultáneas

Para lo cual se deben ejecutar varias tareas en paralelo (concurrentemente)

La ejecución de las tareas se multiplexa en el tiempo en uno o varios procesadores





Requisitos Temporales

Los requisitos en tiempo real se refieren a:

Inicio del Ciclo de Ejecución (**esquema de activación**)

Tareas **periódicas**: se ejecutan a intervalos regulares

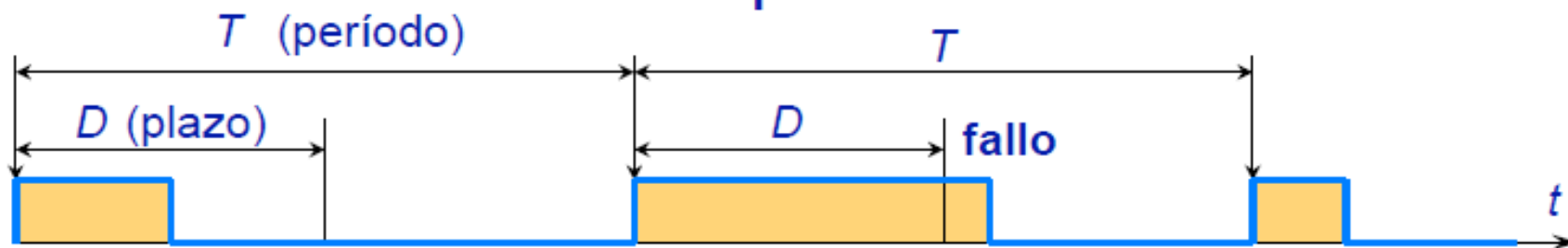
Tareas **esporádicas**: se ejecutan al ocurrir determinados sucesos, en instantes distribuidos irregularmente en el tiempo

Final del Intervalo de Ejecución: se suele especificar un **plazo** (relativo al instante de activación) para terminar la ejecución

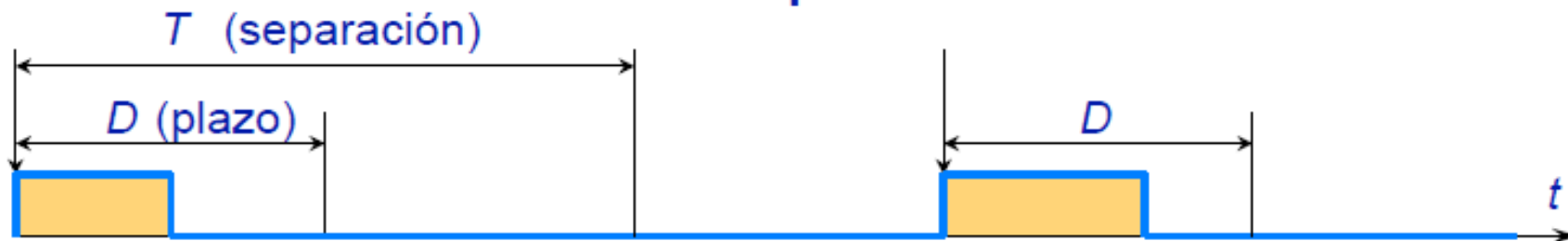


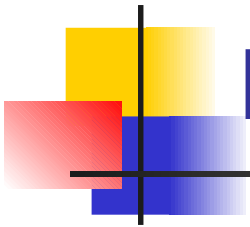
Tareas Periódicas y Espontáneas

Tarea periódica



Tarea esporádica





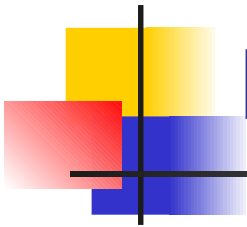
Planificación de Tareas

Se trata de repartir el tiempo de procesador entre varias tareas **de forma que se satisfagan los requisitos temporales**

La relación biunívoca entre acciones y procesadores es un **plan de ejecución** (schedule)

El componente del sistema que hace esto es el **planificador** (scheduler)

Para ello utiliza un **algoritmo de planificación**



Esquemas de Planificación

Planificación **dirigida por tiempo** (time/clock-driven)

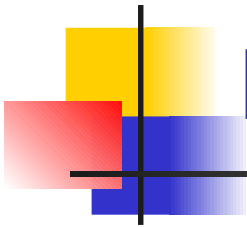
El planificador se ejecuta cada vez que llega una señal de reloj
ejemplo: planificación cíclica

Planificación por **turno circular** (round-robin)

Las acciones listas p/ejecutarse se agrupan en una cola FIFO
Cada acción se ejecuta durante una porción de tiempo y
después se pone al final de la cola
Variante: porciones de tiempo desiguales (ponderadas)

Planificación por **prioridades**

Cada acción tiene una prioridad
Se ejecuta siempre la acción de mayor prioridad entre las listas
La planificación está dirigida por sucesos (event-driven)



Planificación c/ y s/expropiación

Planificación **con expropiación** (preemptive scheduling)

Se le puede quitar el procesador a una acción que se está ejecutando para dar paso a otra (se combina normalmente con prioridades)

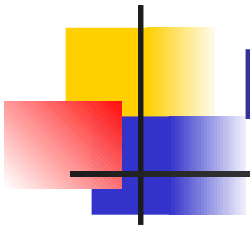
Planificación **sin expropiación** (non-preemptive scheduling)

Una acción que ha comenzado a ejecutarse sólo deja el procesador si:

- Termina su ejecución

- Necesita un recurso que no está disponible

- Abandona el procesador voluntariamente



Prioridades Fijas y Variables

Planificación con **prioridades fijas** (estáticas)

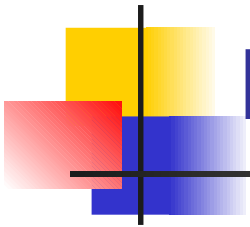
La prioridad de las acciones de una misma tarea es siempre la misma (puede variar si hay cambios de modo).

Ejemplo: prioridades monótonas en frecuencia (rate-monotonic scheduling): mayor prioridad a la tarea con período más corto

Planificación con **prioridades variables** (dinámicas)

La prioridad de una acción se decide en el momento de ejecutarla

Ejemplo: primero la más urgente (earliest deadline first): mayor prioridad a la acción que deba terminar antes



Modelos de Tareas

Un modelo de tareas especifica las características de las tareas de un sistema de tiempo real

Se restringen para poder analizar el sistema y garantizar los requisitos temporales

Ejemplos:

- Sólo tareas periódicas independientes

- Tareas periódicas y esporádicas independientes

- Tareas con comunicación y sincronización

- Tareas estáticas o dinámicas

Comenzar con modelos sencillos: tareas periódicas independientes



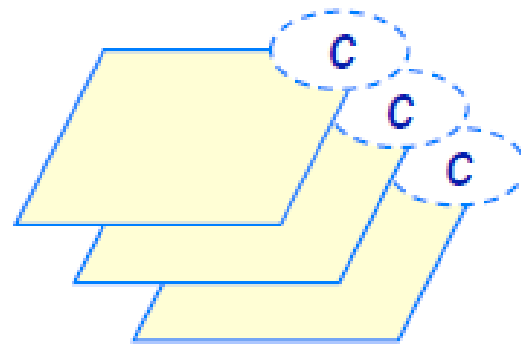
Modelo de Tareas Cíclico

Hay muchos sistemas de tiempo real que sólo tienen tareas periódicas

Son más fáciles de construir

Su comportamiento está completamente determinado

Inicialmente consideramos que no hay comunicación entre tareas (tareas independientes)

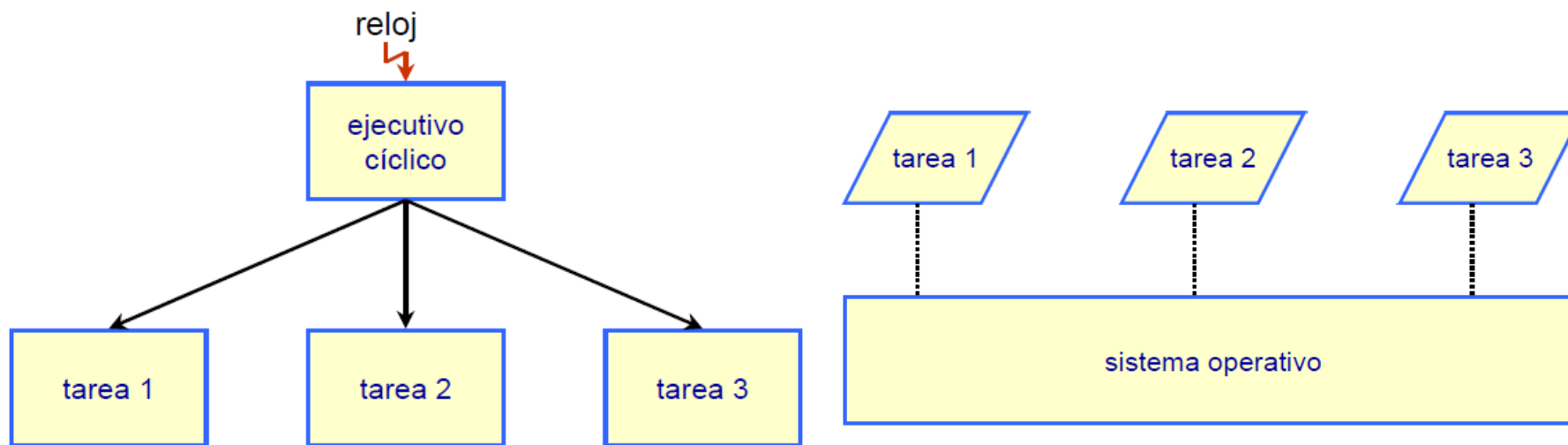




Arquitectura Sincrónica/Asincrónica

Las tareas se ejecutan según un **plan de ejecución** fijo (realizado por el diseñador)

El sistema operativo se reemplaza por un **ejecutivo cíclico**



Parámetros Temporales

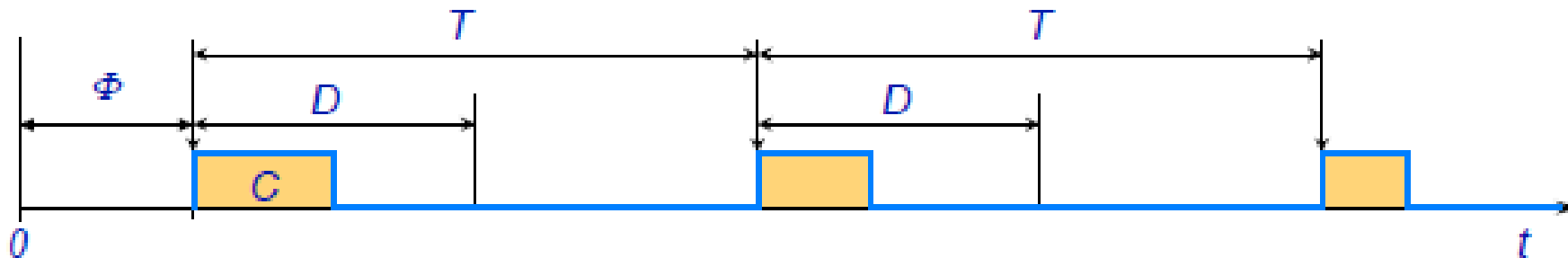
Una tarea periódica se define por sus parámetros (Φ , T , C , D)

Φ es la fase

T es el período de activación de la tarea (Minimum time between process releases - process period)

C es su tiempo de procesamiento en el peor caso (Worst-case computation time -WCET- of the process)

D es el plazo de respuesta relativo a la activación (Deadline of the process)



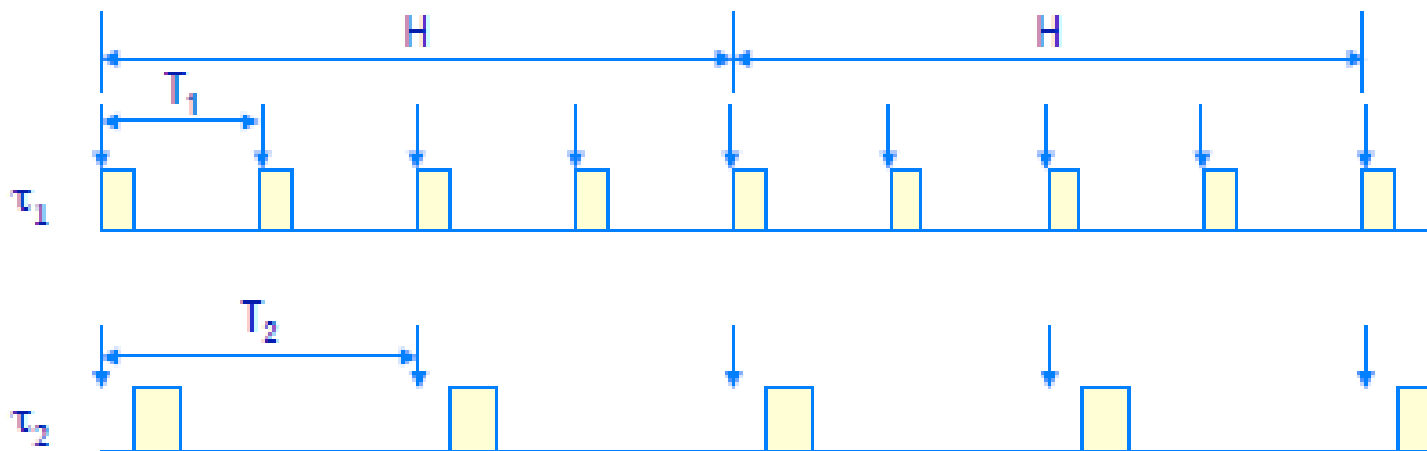


Hiperperíodo

En un sistema formado únicamente por tareas periódicas con períodos T_i , $i = 1 \dots N$, el comportamiento global se repite con un período:

$$H = \text{mcm}(T_i)$$

H es el **hiperperíodo** del sistema





Planificación Estática

Si todas las tareas son periódicas, se puede confeccionar un **plan de ejecución** fijo

Se trata de un esquema que se repite cada:

$$T_M = \text{mcm}(T_i) \text{ ciclo principal (hiperperíodo del sistema)}$$

El ciclo principal se divide en **ciclos secundarios**, con período:

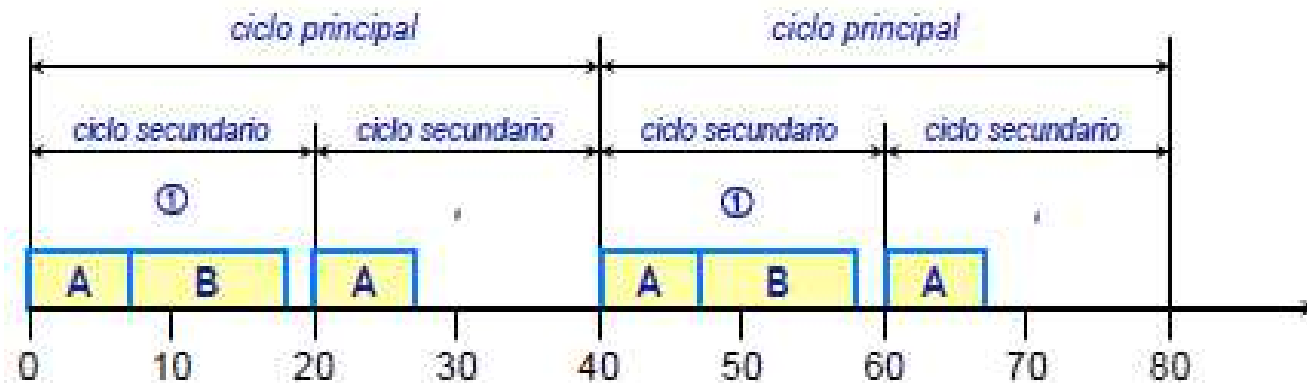
$$T_s (T_M = K T_s)$$

En cada ciclo secundario se ejecutan las actividades correspondientes a determinadas tareas

Planificación Estática: Ejemplo 1



Plan cíclico: $T_M = 40 \text{ ms}$; $T_S = 20 \text{ ms}$



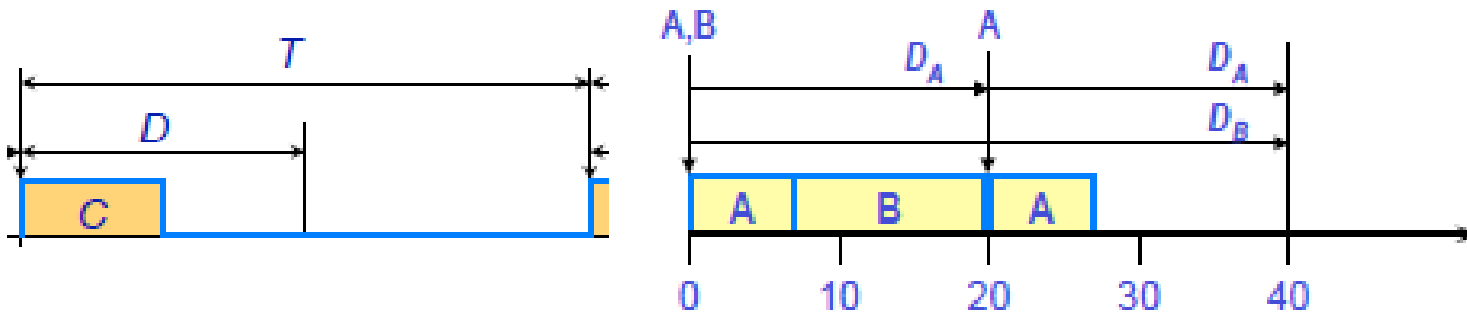
Planificación Estática: Ejemplo 1

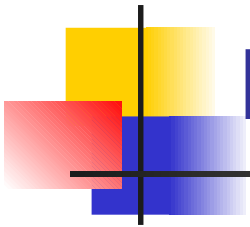
Plazos de Respuesta

Se comprueba que se cumplen los plazos directamente sobre el plan de ejecución

Para ello hace falta conocer el tiempo de cómputo de cada tarea

| | | | |
|-----|-----------|-----------|-----------|
| A : | T = 20 ms | D = 20 ms | C = 8 ms |
| B : | T = 40 ms | D = 40 ms | C = 12 ms |





Factor de Utilización

La cantidad:

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

se denomina **factor de utilización** del procesador (p/ D = T)

$$U \leq 0.69 \quad \text{as } N \rightarrow \infty$$

Es una medida de la carga del procesador para un conjunto de tareas

Para poder elaborar un plan de ejecución que garantice los plazos de todas las tareas, debe ser: **$U \leq 1$**



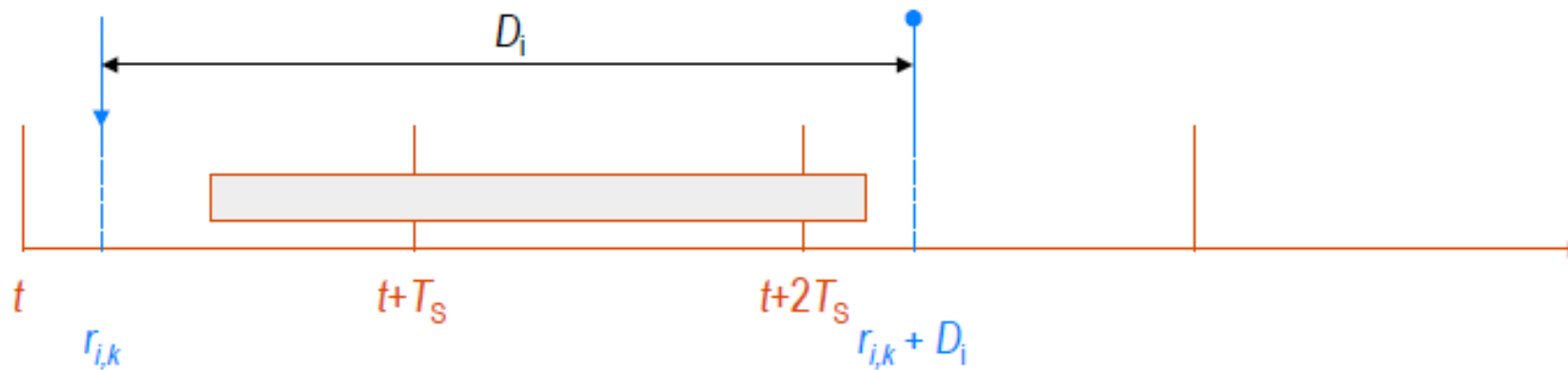
Parámetros del plan cíclico

$$\begin{aligned} (1) \quad & T_s \geq \max C_i \\ (2) \quad & \exists i : T_i/T_s - \lfloor T_i/T_s \rfloor = 0 \\ (3) \quad & \forall i : 2T_s - \text{mcd}(T_s, T_i) \leq D_i \end{aligned}$$

El período secundario debe cumplir ciertas condiciones (Baker & Shaw, 1989):

1. Todas las acciones deben caber en un marco
2. El período secundario divide al menos al período de una tarea (y por tanto al período principal)
3. Entre el instante de activación y el tiempo límite de cada acción debe haber al menos un marco completo (para poder comprobar si la acción termina a tiempo). Si la acción se activa al comienzo de un marco, basta que $T_s \leq D_i$

Condición 3



$$\begin{aligned}
 t + 2T_s &\leq r_{i,k} + D_i \\
 2T_s - (r_{i,j} - t) &\leq D_i \\
 r_{i,j} - t &\geq \text{mcd}(T_i, T_s) \\
 2T_s - \text{mcd}(T_i, T_s) &\leq D_i
 \end{aligned}$$

Si $r_{i,k} = t$, basta con que sea
 $T_s \leq D_i$

Planificación Estática: Ejemplo 2

| <i>Tarea</i> | <i>C</i> | <i>T</i> | <i>D</i> |
|--------------|----------|----------|----------|
| <i>T1</i> | 10 | 40 | 40 |
| <i>T2</i> | 18 | 50 | 50 |
| <i>T3</i> | 10 | 200 | 200 |
| <i>T4</i> | 20 | 200 | 200 |

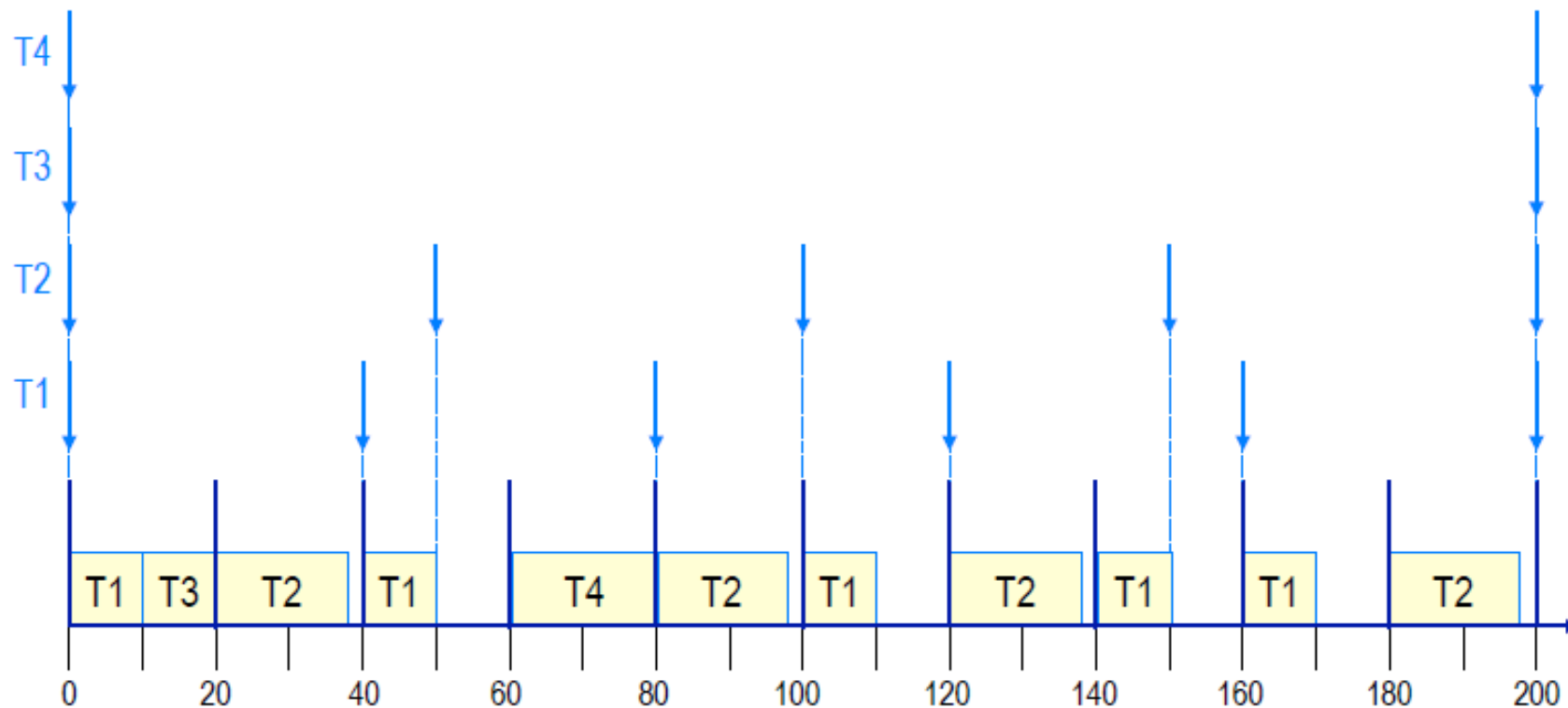
$$U = 0,76$$

$$T_M = 200$$

- 1) $T_s \geq 20$
- 2) $T_s \in \{20, 25, 40, 50, 100, 200\}$
- 3) $2T_s - \text{mcd}(T_s, 40) \leq 40$
 $2T_s - \text{mcd}(T_s, 50) \leq 50$
 $2T_s - \text{mcd}(T_s, 200) \leq 200$

$$T_s = 20$$

Planificación Estática: Plan Cíclico

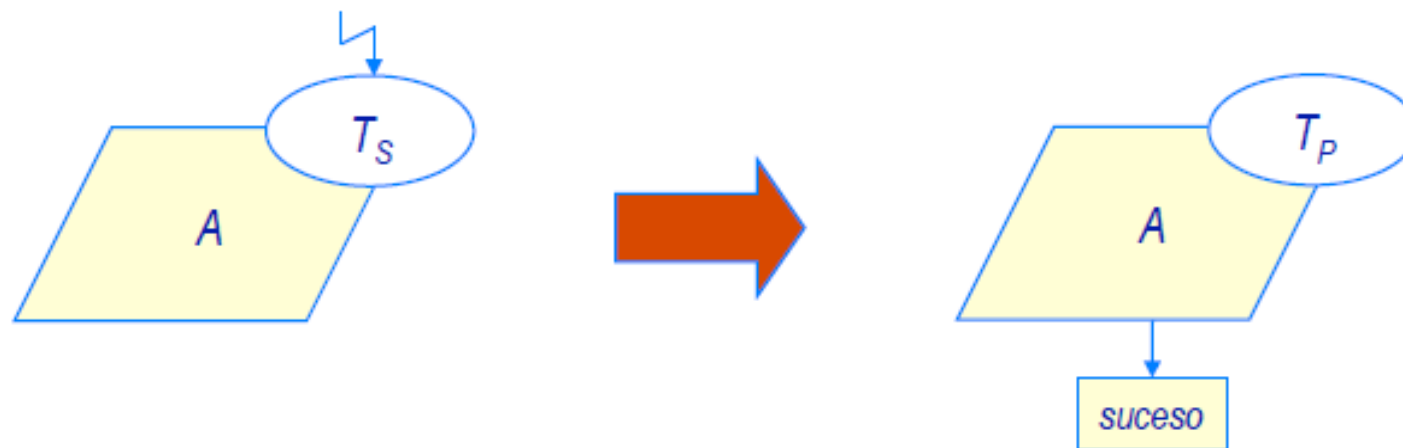




Tareas Esporádicas

El ejecutivo cíclico sólo permite ejecutar tareas periódicas

Las tareas esporádicas se ejecutan con un **servidor de consulta** (polling server). Se trata de una tarea periódica que consulta si se ha producido el suceso esporádico o no, donde el período depende de la separación mínima entre eventos y del plazo de respuesta

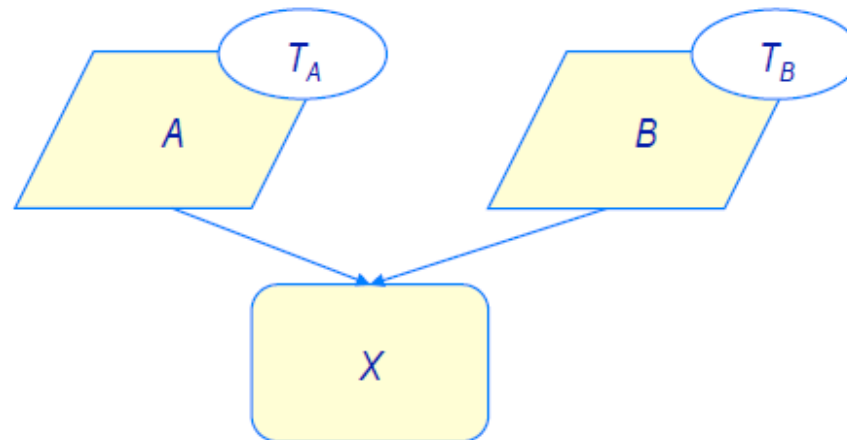




Recursos Compartidos

Una tarea (o segmento) se ejecuta sin interrupción hasta que termina

No es necesario proteger los recursos compartidos
La exclusión mutua es automática





Segmentación de Tareas/Problemas

A veces no es posible confeccionar un plan cíclico que garantice los plazos

Si $U \leq 1$, es posible planificar la ejecución segmentando una o más tareas (los segmentos son secuencias de instrucciones de la tarea con un tiempo de cómputo conocido)

A veces es difícil ajustar el tiempo de cómputo de los segmentos

Si hay recursos compartidos, cada sección crítica debe estar incluida en un solo segmento

Si se modifica una sola tarea hay que rehacer la planificación completa (y posiblemente volver a segmentar de otra manera)



Planificación Estática: Ejemplo 3

| <i>Tarea</i> | <i>C</i> | <i>T</i> | <i>D</i> |
|--------------|----------|----------|----------|
| τ_1 | 10 | 40 | 40 |
| τ_2 | 20 | 100 | 100 |
| τ_3 | 50 | 200 | 200 |

$$U = 0,95$$

$$T_M = 200$$

- 1) $T_s \geq 50$
- 2) $T_s \in \{50, 100, 200\}$
- 3) $2T_s - \text{mcd}(T_s, 40) \leq 40$
 $2T_s - \text{mcd}(T_s, 100) \leq 100$
 $2T_s - \text{mcd}(T_s, 200) \leq 200$

- ◆ Ningún valor cumple (1) y (3)
- ◆ No hay solución aceptable

Planificación Estática: Segmentación

| <i>Tarea</i> | <i>C</i> | <i>T</i> | <i>D</i> |
|--------------|----------|----------|----------|
| τ_1 | 10 | 40 | 40 |
| τ_2 | 20 | 100 | 100 |
| $\tau_{3.1}$ | 10 | 200 | 200 |
| $\tau_{3.2}$ | 30 | 200 | 200 |
| $\tau_{3.3}$ | 10 | 200 | 200 |

- (1) $T_s \geq 30$
- (2) $T_s \in \{40, 50, 100, 200\}$
- (3) $2T_s - \text{mcd}(T_s, 40) \leq 40$
 $2T_s - \text{mcd}(T_s, 100) \leq 100$
 $2T_s - \text{mcd}(T_s, 200) \leq 200$

$T_s = 40$ cumple todas las condiciones





Construcción del Plan de Cíclico

Tres tipos de decisiones interdependientes:

- Ajustar el tamaño de los marcos

- Segmentar acciones & Colocar los segmentos en marcos

En general, el problema es NP-duro

- No hay algoritmos eficientes que resuelvan todos los casos

Se usan algoritmos heurísticos

- Se construye un árbol de soluciones parciales

- Se puede empezar colocando las tareas más urgentes

- Se podan las ramas según algún criterio heurístico

Es más fácil cuando el sistema es armónico

- Pero esto puede forzar una mayor utilización del procesador

Cuando los períodos son muy dispares es más difícil

- Muchos ciclos secundarios en cada ciclo principal



Plan de Cíclico: Conclusiones

Los sistemas cíclicos, con arquitectura síncrona, tienen muchas ventajas

- Implementación sencilla y robusta
- Determinismo temporal
- Es posible certificar que son seguros

Pero tienen inconvenientes importantes

- Mantenimiento difícil y costoso
 - Si se cambia algo hay que empezar desde el principio
 - La segmentación añade mucha complejidad

- Es difícil incluir tareas esporádicas

En general, es un método de bajo nivel

- Sólo es apropiado para sistemas que no se modifican una vez construidos



Plan de Cíclico: Conclusiones

Considere de qué modo configurar FreeRTOS para planificar tareas según el modelo cíclico.



Referencias

Real-Time Systems and Programming Languages (Fourth Edition)
Ada 95, Real-Time Java and Real-Time POSIX by Alan Burns and
Andy Wellings

<http://www.cs.york.ac.uk/rts/books/RTSBookFourthEdition.html>

Sistemas de Tiempo Real, Juan Antonio de la Puente,
Departamento de Ingeniería de Sistemas Telemáticos, Universidad
Politécnica de Madrid <http://web.dit.upm.es/~jpuente/str/>

Introducción al tiempo real en sistemas empotrados (Master Oficial
en Ingeniería de Sistemas Empotrados), Alberto Lafuente,
Universidad del País Vasco,

<http://www.sc.ehu.es/acwlaroa/ITRSE.htm>

Sistemas en Tiempo Real, Escuela Universitaria de Ingeniería de
Gijón <http://isa.uniovi.es/docencia/TiempoReal/>

The Cheddar project: a free real time scheduling analyzer

<http://beru.univ-brest.fr/~singhoff/cheddar/>