

Planificación multiprocesador y de tiempo real



1.

Planificación multiprocesador

Clasificación de los sistemas multiprocesador

- Debilmente acoplado o multiprocesador distribuído.
- Procesadores de funcionalidad especializada.
- **Procesamiento fuertemente acoplado**

Caracterizar un multiprocesador:

Granularidad

Tamaño del Grano	Descripción	Intervalo de sincronización (Instrucciones)
Fino	Paralelismo inherente en un único flujo de instrucciones	<20
Medio	Procesamiento paralelo o multitarea dentro de una única aplicación	20-200
Grueso	Multiprocesamiento de procesos concurrentes en un entorno multiprogramado	200-2000
Muy grueso	Procesamiento distribuido entre nodos de una red para conformar un único entorno de computación	2000-1M
Independiente	Múltiples procesos no relacionados	(N/D)

“

En un multiprocesador la planificación involucra tres aspectos interrelacionados:

- La asignación de procesos a procesadores.
- El uso de la multiprogramación en cada procesador individual.
- La activación del proceso, propiamente dicha.

Planificación: Asignación de procesos a procesadores

A TENER EN CUENTA:

La **información de contexto** de todos los procesos estará **disponible** para todos los procesadores.



El **coste de planificación** de un proceso será **independiente** de la identidad del procesador sobre el cual se planifica.

Planificación: Asignación de procesos a procesadores

MAESTRO/ESCLAVO:

- Ciertas funciones claves del núcleo se ejecutan siempre en un procesador concreto. (maestro)
- El resto **solo** puede ejecutar **programas de usuario**.(esclavo)
- El maestro es el responsable de la planificación de trabajos.



ENFOQUE SIMPLE



SE SIMPLIFICA LA RESOLUCIÓN DE CONFLICTOS



FALLO EN MAESTRO

=

FALLO DEL SISTEMA

Planificación: Asignación de procesos a procesadores

CAMARADA:

- El núcleo puede ser ejecutado desde cualquier procesador.
- C/ procesador se auto-planifica desde la colección de procesos disponibles.



COMPLICA AL S.O

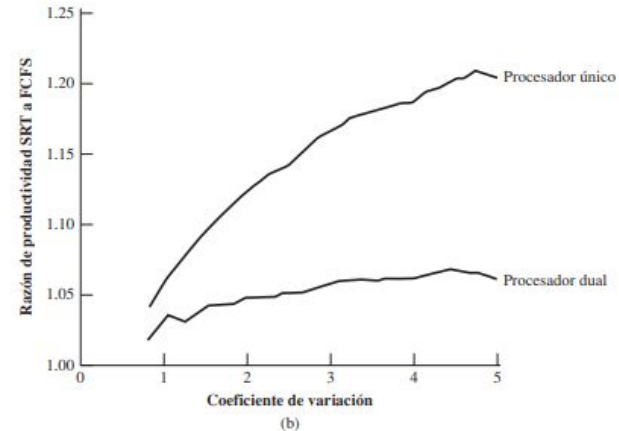
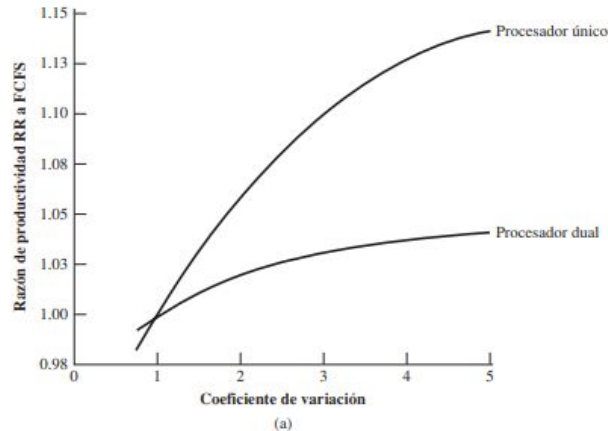


REQUIERE TÉCNICAS PARA RESOLVER Y
SINCRONIZAR LA COMPETENCIA DE DEMANDA

Planificación de procesos

Los procesos no se vinculan a los procesadores → Se utilizan colas

La disciplina de planificación específica es mucho menos importante con dos procesadores que con uno.



Planificación de hilos

El poder completo de los hilos se vuelve evidente en un sistema multiprocesador

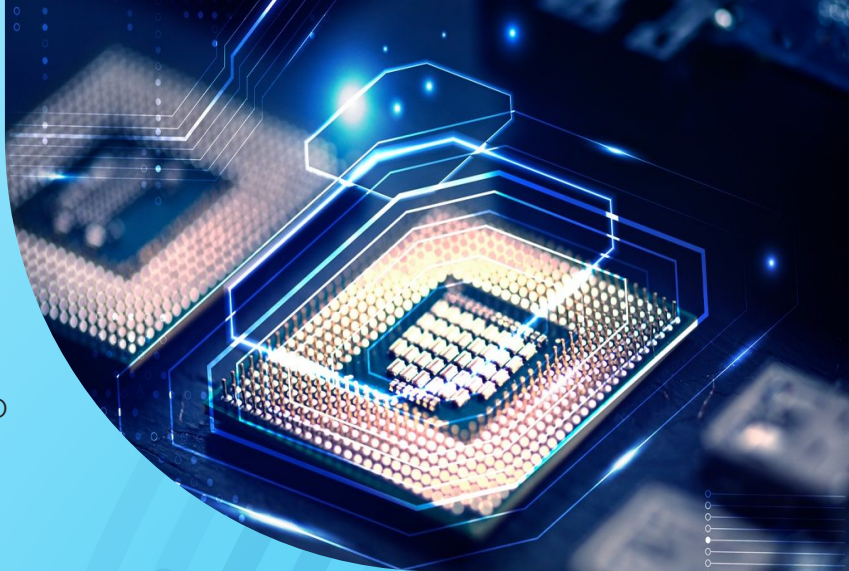


Los hilos pueden explotar el paralelismo real dentro de una aplicación

Existen 4 enfoques generales para la planificación multiprocesador de hilos:

- Compartición de carga
- Planificación en pandilla
- Asignación de procesador dedicado
- Planificación dinámica

Compartición de carga



Los procesos no se asignan a un procesador particular:

- Cola global de hilos listos
- C/ procesador cuando está ocioso, selecciona un hilo



Enfoque + **simple**



La carga se **distribuye uniformemente** entre los procesadores

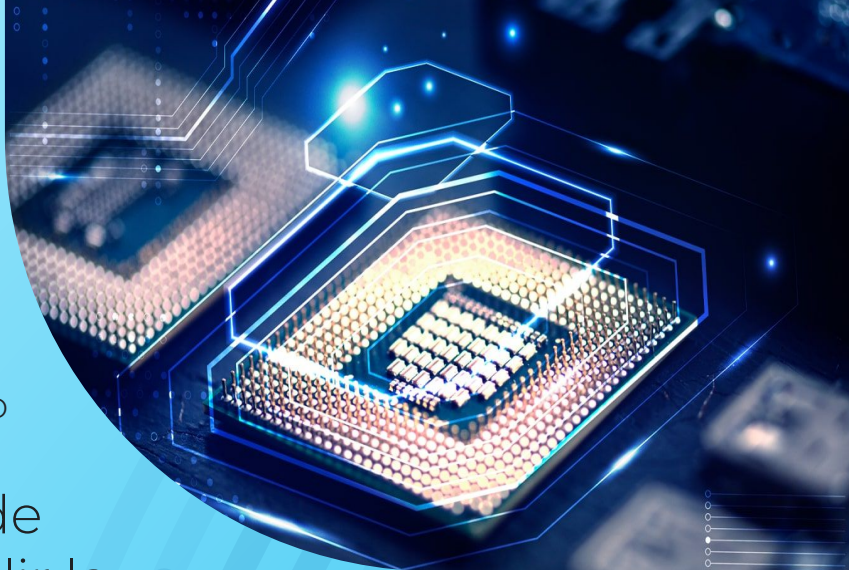


No se precisa un planificador centralizado



La cola global puede organizarse y ser **accesible** usando **cualquiera de los esquemas**

Compartición de carga



Los procesos no se asignan a un procesador particular:

- Cola global de hilos listos
- C/ procesador cuando está ocioso, selecciona un hilo

✗ La cola central ocupa una región de memoria en la cual se debe cumplir la **exclusión mutua**.

✗ Se **desfavorece la caché local**: es poco probable que los hilos expulsados retomen en el mismo procesador.

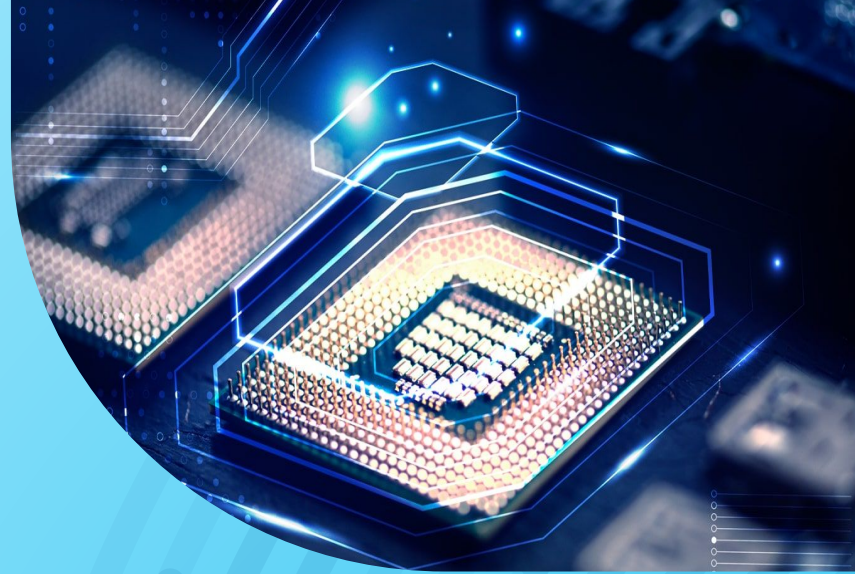
✗ Los cambios de procesos necesarios pueden **comprometer seriamente el rendimiento**.

Compartición de carga

Se analiza usando el siguiente sistema de compartición de carga:

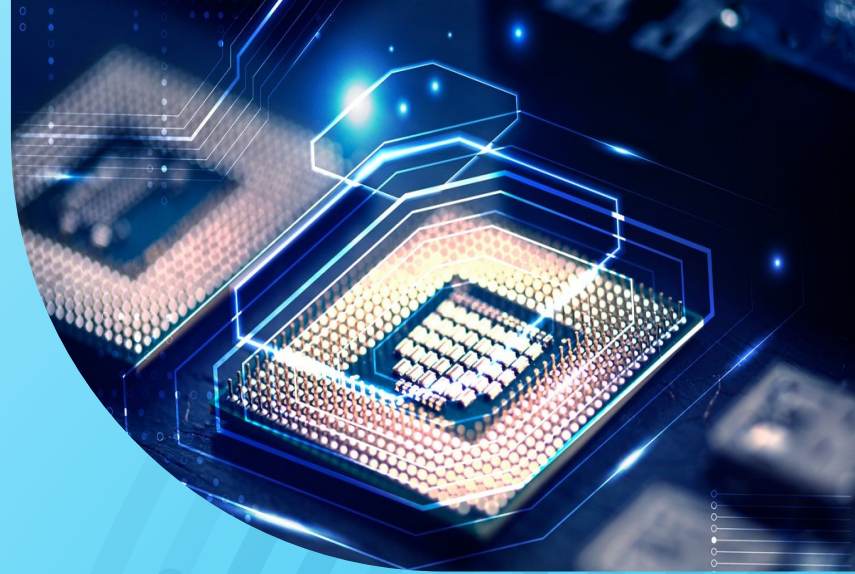
Primero en llegar, primero en ser servido (FCFS):

1. Llega un trabajo.
2. Los hilos se disponen consecutivamente al final de la cola compartida.
3. Un procesador pasa a estar ocioso.
4. Se elige el siguiente hilo listo, que ejecuta hasta que se completa o se bloquea.



Planificación en pandilla

Un conjunto de hilos relacionados que se planifica para ejecutar sobre un conjunto de procesadores al mismo tiempo, en una relación uno-a-uno.



La planificación en pandilla es para aplicaciones paralelas de grano medio o fino cuyo rendimiento se ve degradado de forma importante cuando cualquier parte de la aplicación no está ejecutando mientras otras partes están listas para ejecutar.

La necesidad de planificación en pandilla está ampliamente reconocida, y existen implementaciones en variedad de sistemas operativos multiprocesador.

Planificación en pandilla

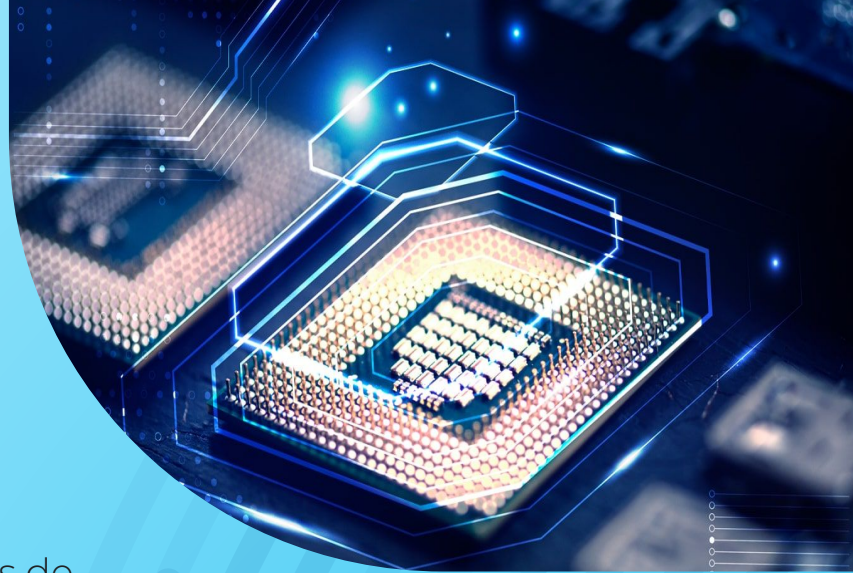
Se ha aplicado a la planificación simultánea de los hilos que componen un proceso individual



Si se ejecutan en paralelo procesos estrechamente relacionados, puede reducirse el bloqueo por sincronización, pueden necesitarse menos cambios de proceso y las prestaciones aumentarán.



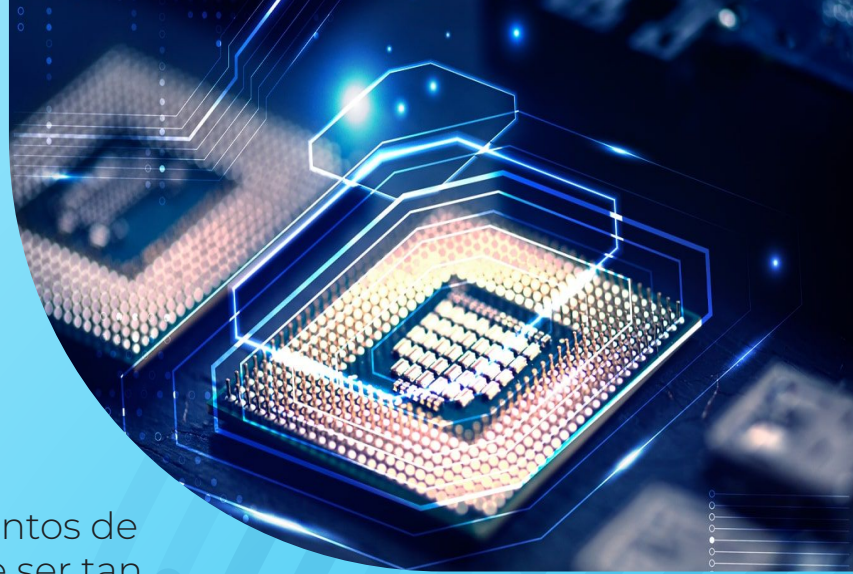
La sobrecarga de planificación puede reducirse dado que una decisión única afecta a varios procesadores y procesos a la vez.



Asignación de procesador dedicado

Forma extrema de la planificación en pandilla

Dedicar un grupo de procesadores a una aplicación durante toda la duración.



En Sistema **altamente paralelo** con decenas o cientos de procesadores, la utilización del procesador deja de ser tan importante como medida de eficacia

Evitar totalmente el cambio de proceso durante la vida de un programa

=

Sustancial mejora de velocidad de ese programa



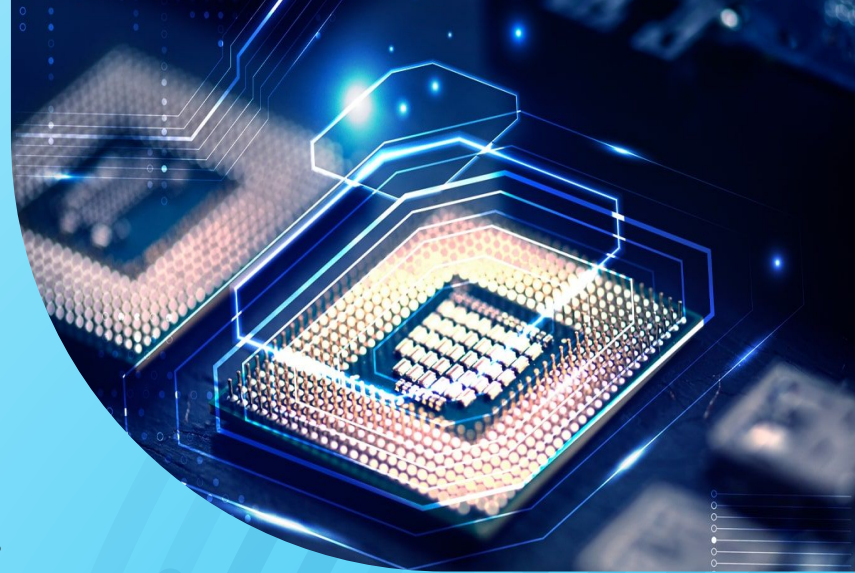
Planificación dinámica

Se utilizan herramientas que permiten que el número de hilos del proceso pueda ser alterado dinámicamente, mientras se está ejecutando..

Sistema operativo ajusta la carga → mejorar la utilización

Cuando un trabajo solicita uno o más procesadores...

1. Si hay procesadores ociosos, utilizarlos para satisfacer la solicitud.
2. En otro caso, si el trabajo que realiza la solicitud acaba de llegar, ubicarlo en un único procesador quitándoselo a cualquier trabajo que actualmente tenga más de un procesador.
3. Si no puede satisfacerse cualquier parte de la solicitud, mantenerla pendiente hasta que un procesador pase a estar disponible, o el trabajo rescinda la solicitud
4. Cuando se libera uno o más procesadores, hay que examinar la cola actual de solicitudes de procesador no satisfechas y asignar un único procesador a cada trabajo en la lista que no tenga actualmente procesadores. Luego volver a examinar la lista, volviendo a asignar el resto de los procesadores siguiendo la estrategia FCFS.





2.

Planificación de tiempo real

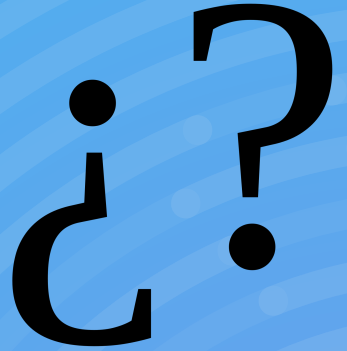
Distintos Algoritmos

- **Estáticos dirigidos por tabla**
- **Estáticos expulsivos dirigidos por prioridad**
- **Dinámicos basados en un plan**
- **Dinámicos de mejor esfuerzo**

Planificación por plazos

Se tienen en consideración las siguientes dimensiones:

- ¿Qué tarea a ejecutar a continuación?**
- ¿Qué tipo de expulsión se permite?**



La politica de planificar la tarea de plazo mas cercano minimiza la cantidad de tareas que fallan

Planificación por plazos: Tareas periódicas

Tabla 10.2. Perfil de ejecución de dos tareas periódicas.

Proceso	Tiempo de llegada	Tiempo de ejecución	Plazo de conclusión
A(1)	0	10	20
A(2)	20	10	40
A(3)	40	10	60
A(4)	60	10	80
A(5)	80	10	100
•	•	•	•
•	•	•	•
•	•	•	•
B(1)	0	25	50
B(2)	50	25	100
•	•	•	•
•	•	•	•
•	•	•	•

Planificación por plazos: Tareas periódicas

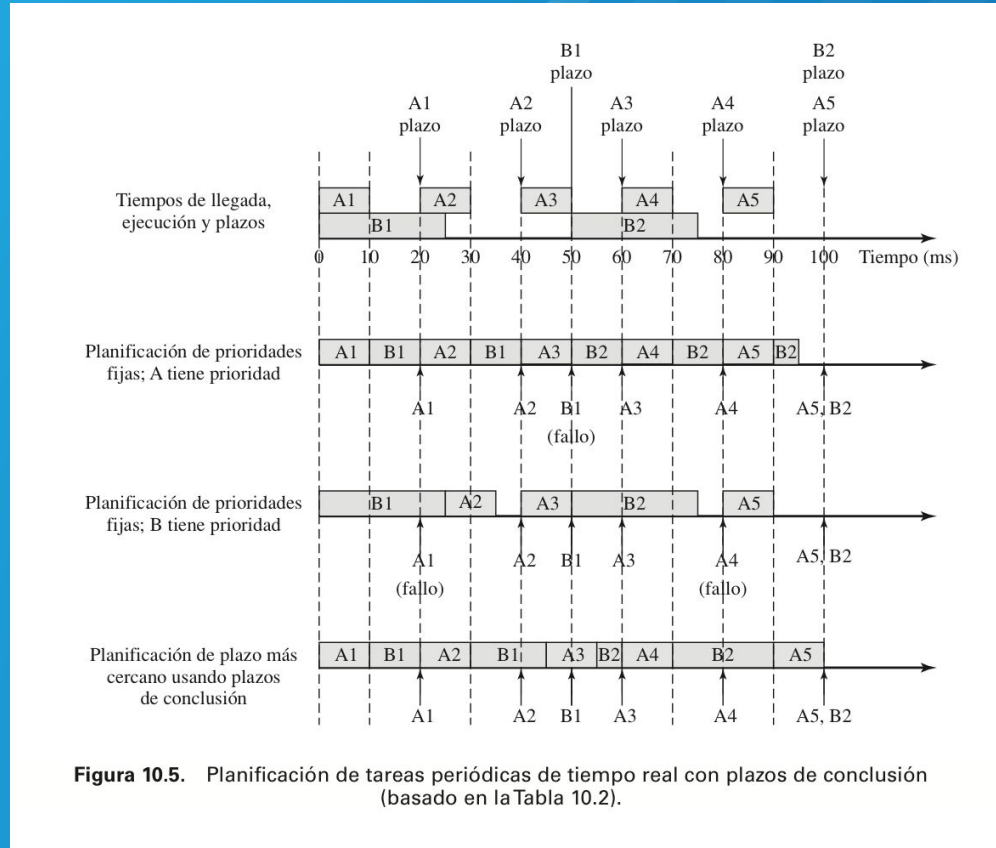


Figura 10.5. Planificación de tareas periódicas de tiempo real con plazos de conclusión (basado en la Tabla 10.2).

Planificación por plazos: Tareas aperiódicas

Tabla 10.3. Perfil de ejecución de cinco tareas aperiódicas.

Proceso	Tiempo de llegada	Tiempo de ejecución	Plazo de comienzo
A	10	20	110
B	20	20	20
C	40	20	50
D	50	20	90
E	60	20	70

Planificación por plazos: Tareas aperiódicas

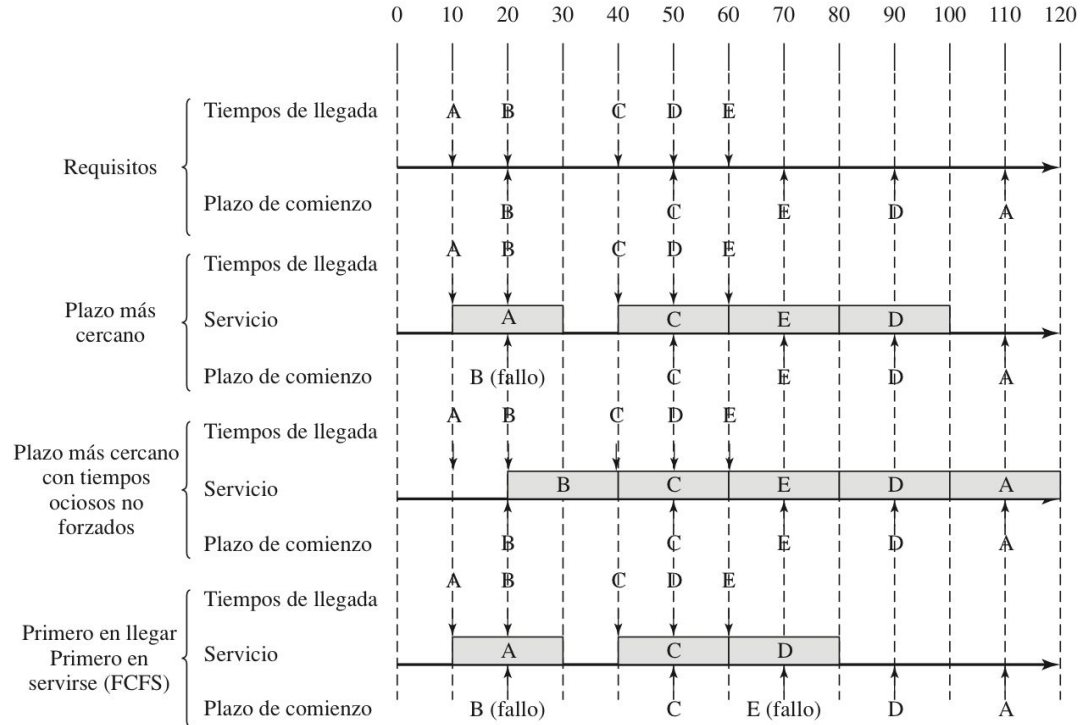


Figura 10.6. Planificación de tareas aperiódicas de tiempo real con plazos de comienzo.

Planificación de Tasa Monótona(RMS)

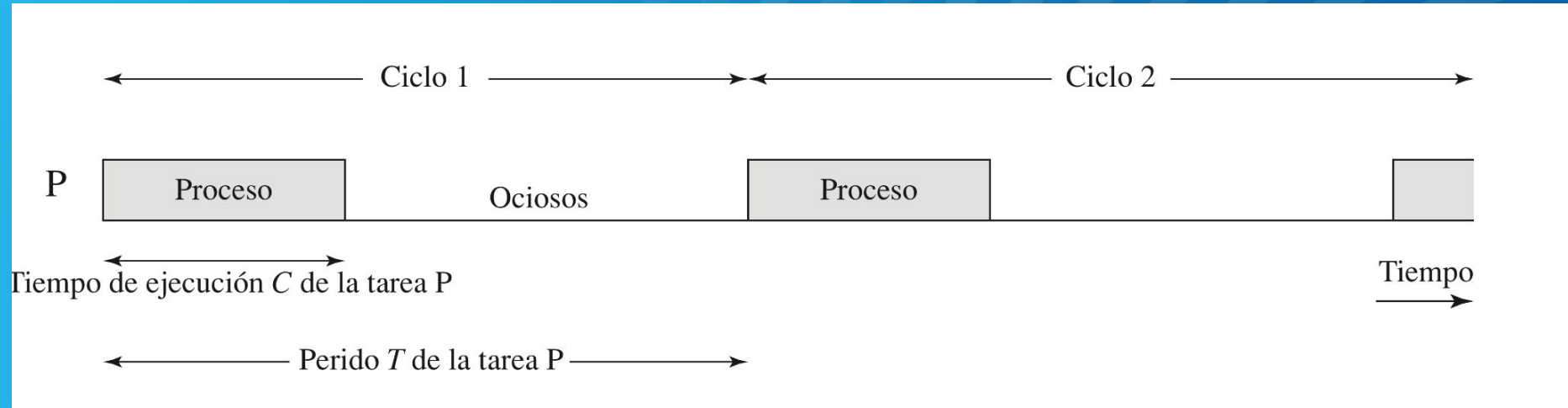
- Resolver conflictos de planificación multitarea para tareas periódicas.
- Asigna prioridades a las tareas en base a sus periodos
- La tarea con mayor prioridad es la de periodo más breve

“

La planificación en tiempo real, no busca que se procese de la manera más rápida, sino que se cumplan en todos los plazos duros

Planificación de Tasa Monótona(RMS)

Para que sea posible cumplir todos los plazos debe cumplirse la siguiente relación:



Planificación de Tasa Monótona(RMS)

Para el RMS se cumple la siguiente relación:

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_n}{T_n} \leq n(2^{1/n} - 1)$$

n	$n(2^{1/n} - 1)$
1	1,0
2	0,828
3	0,779
4	0,756
5	0,743
6	0,734
•	•
•	•
•	•
∞	$\ln 2 \approx 0,693$

$$U_i = C_i / T_i$$

Tarea P₁: C₁ = 20; T₁ = 100; U₁ = 0,2

Tarea P₂: C₂ = 40; T₂ = 150; U₂ = 0,267

Tarea P₃: C₃ = 100; T₃ = 350; U₃ = 0,286

La utilización total de estas tres tareas es 0,2 + 0,267 + 0,286 = 0,753

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} \leq 3(2^{1/3} - 1) = 0,779$$

$$0,753 < 0,779$$

Inversión de prioridad

Fenómeno que puede suceder en cualquier esquema de planificación expulsivo basado en prioridades

Sucede cuando las circunstancias dentro del sistema fuerzan a una tarea de mayor prioridad a esperar por una de menor, sobre todo cuando ha bloqueado un recurso

Una situación más seria, es la inversión de prioridad ilimitada: la duración de la inversión depende también de las acciones impredecibles de otras tareas no relacionadas.

RMS: Ejemplo Pathfinder

T_1 . Comprueba periódicamente la salud del software y los sistemas de la nave espacial.

T_2 . Procesa datos de imágenes.

T_3 . Realiza ocasionalmente una comprobación del estado de los equipos.

t_1 . T_3 comienza a ejecutar.

t_2 . T_3 bloquea el semáforo s y entra en su sección crítica.

t_3 . T_1 , que tiene mayor prioridad que T_3 , expulsa a T_3 y comienza a ejecutar.

t_4 . T_1 intenta entrar en su sección crítica pero se bloquea porque el semáforo está en propiedad de T_3 , T_3 retoma su ejecución en su sección crítica.

t_5 . T_2 , que tiene mayor prioridad que T_3 , expulsa a T_3 y comienza a ejecutar.

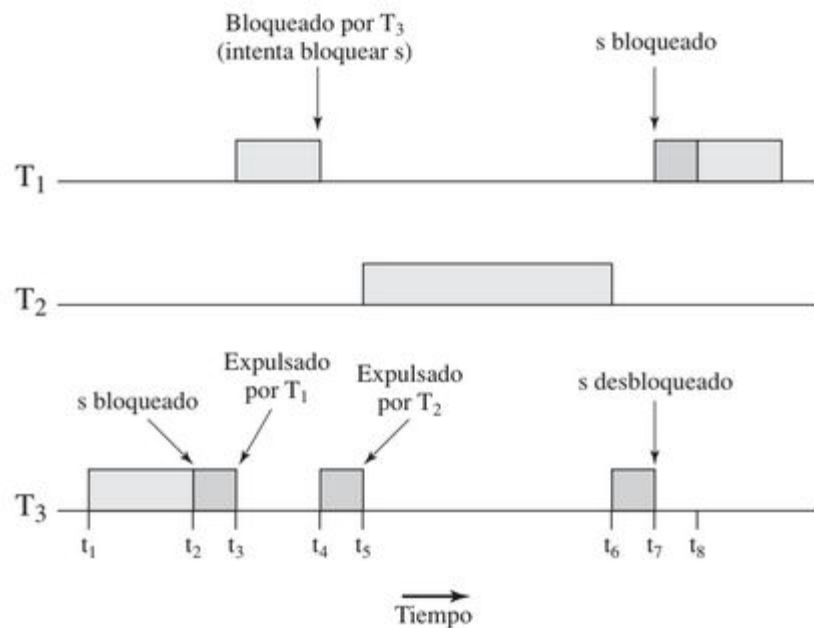
t_6 . T_2 se suspende por alguna causa no relacionada con T_1 ni T_2 , y se retoma T_3 .

t_7 . T_3 abandona su sección crítica y desbloquea el semáforo. T_1 expulsa a T_3 , bloquea el semáforo y entra en su sección crítica.

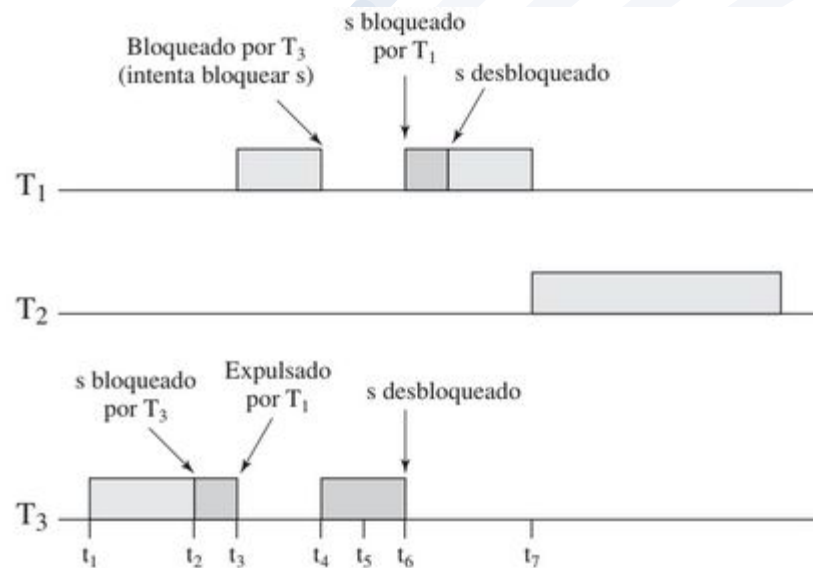
En esta secuencia de circunstancias, T_1 debe esperar a que T_3 y T_2 concluyan, y no alcanza a re-inicializar el temporizador antes de que expire.

Soluciones

- **Herencia de Prioridad:**
Una tarea de menor prioridad heredará la prioridad de cualquier tarea mayor pendiente de un recurso que compartan
- **Protocolo de techo de prioridad:**
Se asocia una prioridad con cada recurso. La prioridad asociada con un recurso es un nivel más alta que la prioridad de su usuario más prioritario.
Entonces, el planificador asigna esta prioridad a cualquier tarea que acceda al recurso.



(a) Inversión de prioridad ilimitada



(b) Uso de herencia de prioridad

Ejecución normal
 Ejecución en sección crítica



Gracias!