Módulo III

SISTEMAS OPERATIVOS de TIEMPO REAL

Temas del Módulo III

- **Conceptos de sistemas de tiempo real.**
  - Definición de sistema de tiempo real.
  - Sistemas operativos de tiempo real.
  - Tipos de tareas.
  - Requisitos de un SOTR.
  - Características de un SOTR.
- **Planificación de tiempo real.**
  - Planificación por plazos.
  - Planificación de tasa monótona.

Conceptos de Sistemas de Tiempo Real

Definición de sistema de tiempo real.

- Un sistema de tiempo real es aquel que permite censar y controlar eventos del mundo real. Como los datos provienen de esos eventos, la correctitud de su funcionamiento no solo depende de la computación sino del momento en que se producen los resultados.
- Esto significa que el tiempo es un elemento clave en los sistemas de tiempo real (STR). El sistema debe ser capaz de reaccionar a los eventos externos dentro de un plazo de tiempo determinado, de lo contrario, las consecuencias pueden ser catastróficas.
- Un STR está compuesto por:
  - Hardware apto para responder a los eventos externos, donde las IRQ son el elemento fundamental.
  - Sistema operativo que permita administrar los procesos de tiempo real, y el planificador es el alma de un sistema operativo de tiempo real (SOTR).

Conceptos de Sistemas de Tiempo Real

Definición de sistema de tiempo real. *(cont.)*

- Algunos ejemplos de STR son:
  - Estaciones meteorológicas.
  - Sistemas de navegación de aeronaves y control de tráfico aéreo.
  - Centrales digitales de telecomunicaciones.
  - Controles de procesos industriales.
- Algunos ejemplos comerciales de STR y sus aplicaciones son:
  - PikeOS: automotores, aviación, medicina, ferrocarriles, industria.
  - QNX: comprado por BlackBerry en 2010. SO que lo utilizan: BlackBerry 10. Otras aplicaciones: automotores, medicina.
  - Symbian: diseñado originalmente para PDA, luego utilizado en smartphones de Nokia, Samsung, Motorola y Sony Ericsson.
  - FreeRTOS: sistemas embebidos, típicamente con arquitecturas ARM.

Conceptos de Sistemas de Tiempo Real

Sistemas operativos de tiempo real.

- Un sistema de operativo de tiempo real (SOTR) es aquel diseñado para poder ejecutar tareas de tiempo real.
- Estas tareas intentan controlar o reaccionar a los eventos externos mencionados, por lo que deben ser capaces de seguir el ritmo de esos eventos.
- El planificador es el componente más importante de un SOTR. El algoritmo de planificación es fundamental para el correcto funcionamiento de un STR.
- Junto con las IRQ, determinan la eficiencia general del sistema.

Conceptos de Sistemas de Tiempo Real

Tipos de tareas.

- Para poder seguir el ritmo de los eventos externos, las tareas de tiempo real (TTR) tienen asociados un plazo de tiempo límite, donde se especifica el instante comienzo o finalización.
- Según esto, se pueden clasificar en dos categorías:
  - TTR Dura: es aquella que debe cumplir su plazo límite, de otro modo se producirá un daño inaceptable o error fatal en el sistema.
  - TTR Blanda o Suave: tiene asociado un plazo límite deseable pero no es obligatorio; sigue teniendo sentido planificar y completar la tarea aun cuando su plazo límite ya haya vencido.
- También pueden ser:
  - Periódicas: son tareas repetitivas, con una secuencia de tiempos conocida de antemano.
  - Aperiódicas: no son repetitivas, su ejecución se conoce generalmente a partir de la aparición de ciertos eventos.

Conceptos de Sistemas de Tiempo Real

Requisitos de un SOTR.

- Los sistemas SOTR pueden ser caracterizados por tener requisitos únicos en cinco áreas generales:
  - Determinismo.
  - Reactividad.
  - Control de usuario.
  - Fiabilidad.
  - Operación de fallo suave.
- Analizaremos brevemente cada uno de estos requisitos.

Conceptos de Sistemas de Tiempo Real

Requisitos de un SOTR. *(cont.)*

- **Determinismo.**
- Un sistema operativo es determinista cuando realiza operaciones en instantes de tiempo fijos predeterminados, o dentro de intervalos de tiempo predeterminado.
- Como se tienen múltiples procesos compitiendo por los recursos, ningún SO puede ser totalmente determinista.
- El grado en que se pueden satisfacer de manera determinista las peticiones depende de:
  - La velocidad a la que es capaz de responder a las IRQ.
  - La capacidad de manejar todas las solicitudes dentro del tiempo requerido.
- Se puede medir como el tiempo que se tarda en reconocer que llegó una IRQ.

Conceptos de Sistemas de Tiempo Real

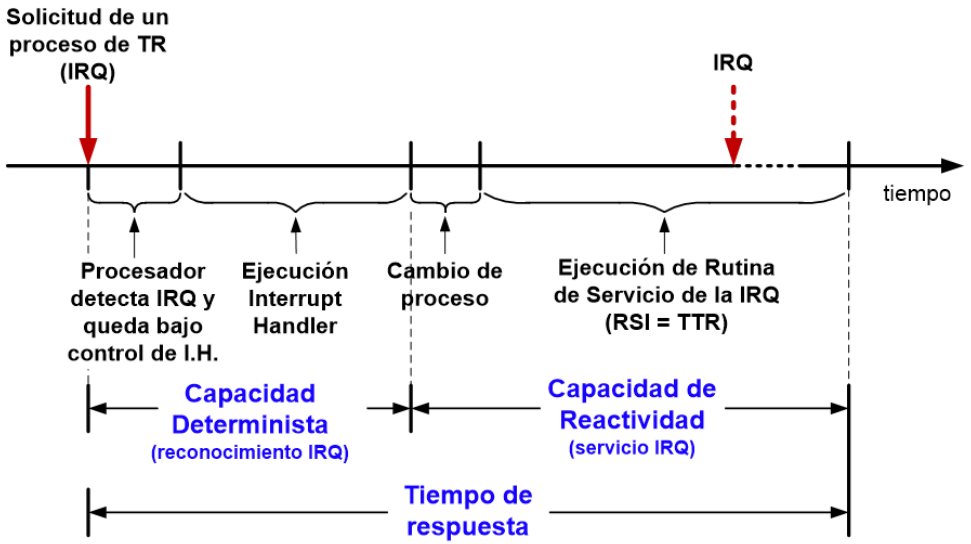
Requisitos de un SOTR. *(cont.)*

- **Reactividad.**
- Está relacionada con el determinismo, y es el tiempo que tarda el SO en atender la IRQ, después de haberla reconocido.
- Este tiempo incluye:
  - La cantidad de tiempo necesaria para manejar la IRQ y comenzar a ejecutar la rutina de servicio de la IRQ (RSI). El retardo puede ser mayor si se requiere un cambio de proceso.
  - El tiempo necesario para ejecutar la RSI, que depende generalmente de la plataforma de hardware.
  - El efecto de anidamiento de las IRQ. Si una IRQ puede ser interrumpida por otra, aumentará el tiempo reactividad.

Conceptos de Sistemas de Tiempo Real

Requisitos de un SOTR. *(cont.)*

- **Reactividad.** *(cont.)*
- El determinismo y la reactividad juntos conforman el tiempo de respuesta a eventos externos. Los requisitos de tiempo de respuesta son críticos para los STR.
- En el siguiente gráfico se muestran los tiempos de determinismo, reactividad y respuesta:



Conceptos de Sistemas de Tiempo Real

Requisitos de un SOTR. *(cont.)*

- **Control de usuario.**
- Es en general mucho mayor en los SOTR que en los SO ordinarios.
- En un SO ordinario, el usuario no tiene control sobre el planificador, a lo sumo puede tener asignada algún tipo de prioridad.
- En un SOTR, el usuario es quien define qué tareas son duras y suaves, debe especificar las prioridades relativas dentro de cada clase.
- Incluso en algunos sistemas, el usuario define ciertas características como el uso de paginación de los procesos, cuáles deben residir siempre en memoria principal, etc.

Conceptos de Sistemas de Tiempo Real

Requisitos de un SOTR. *(cont.)*

- **Fiabilidad.**
- La fiabilidad es normalmente mucho más importante en un STR que para los que no lo son.
- En un sistema ordinario, un fallo transitorio se resuelve simplemente reiniciando el sistema. O un fallo un procesador en un sistema con multiprocesamiento resulta en un servicio degradado hasta que se reemplaza el procesador que falló.
- En cambio, en un STR, la pérdida o degradación de sus funciones puede tener consecuencias catastróficas:
  - Pérdidas económicas.
  - Daños en equipos importantes.
  - Incluso pérdida de vidas.

Conceptos de Sistemas de Tiempo Real

Requisitos de un SOTR. *(cont.)*

- **Operación de fallo suave.**
- Un SOTR debe estar diseñado para responder a varios tipos de fallo. La operación de fallo suave es la habilidad del sistema de fallar de tal manera que se preserve tanta capacidad y datos como sea posible.
- El sistema tratará de corregir el problema o bien minimizar sus efectos, continuando en ejecución. Se notifica al usuario de que se necesita una acción correctiva, y si es necesario apagar el equipo, se tratará de mantener la consistencia de los archivos y datos.
- Un aspecto importante es la estabilidad. Un sistema es estable si en caso de no poder cumplir con todas las tareas, cumple con las más críticas y de mayor prioridad, aunque las menos críticas no se satisfagan.

Conceptos de Sistemas de Tiempo Real

Características de un SOTR.

- Para cumplir con los requisitos mencionados, los SOTR deben tener las siguientes características:
  - Cambio rápido de proceso o hilo.
  - Pequeño tamaño (típicamente un microkernel).
  - Capacidad de responder rápidamente IRQ externas.
  - Multitarea con herramientas de comunicación de procesos (semáforos, señales y eventos).
  - Utilización de archivos secuenciales especiales que puedan acumular datos a alta velocidad.
  - Planificación expulsiva basada en prioridades.
  - Minimización de las inhabilitaciones de las IRQ.
  - Primitivas para retardar tareas y para parar/retomar tareas.
  - Alarmas y temporizaciones especiales.

Planificación de Tiempo Real

- Se mencionó anteriormente que el componente más importante de un SOTR es el planificador de tareas a corto plazo.
- Su diseño es muy distinto al de un SO que no es de TR, en los que se busca equidad en el uso del procesador y minimizar los tiempos de espera y respuesta.
- En un SOTR, el planificador se diseña para que todas las tareas de tiempo real duras se completen (o comiencen) en su plazo, y que la mayoría de las tareas de tiempo real suaves también.

Planificación de Tiempo Real

Planificación por plazos.

- Las aplicaciones de tiempo real no se preocupan tanto de la velocidad de ejecución, sino más bien, como ya se mencionó, de completar o comenzar sus tareas dentro de sus plazos.
- Los enfoques más apropiados para la planificación se basan en tener información adicional de la tarea, como ser:
- Tiempo de activación: momento en el que la tarea está lista para ejecutar. En el caso de las tareas periódicas, es una secuencia de tiempos conocida de antemano. En las tareas aperiódicas, puede ser conocido de antemano, o bien el sistema puede ser consciente de la tarea cuando ésta pasa a estar lista.
- Plazo de comienzo: momento en el cual la tarea deber comenzar.

Planificación de Tiempo Real

Planificación por plazos. *(cont.)*

- Plazo de conclusión: momento para el cual la tarea debe estar completada. Se especifica plazo de comienzo o de conclusión, pero no ambos.
- Tiempo de proceso: es el tiempo necesario para ejecutar la tarea hasta su conclusión. En algunos casos está disponible, y en otros, el sistema lo puede estimar .
- Recursos requeridos: conjunto de recursos, distintos del procesador.
- Prioridad: mide la importancia relativa de la tarea. Puede ser absoluta o relativa, según cuán exigente sea el sistema.
- Estructura de subtareas: una tarea puede ser dividida en tareas duras y suaves, teniendo las duras plazos obligatorios.

Planificación de Tiempo Real

Planificación por plazos. *(cont.)*

- Hay varios aspectos en los que se consideran los plazos:
  - Qué tarea ejecutar a continuación.
  - Qué tipo de expulsión se permite.
- Se puede demostrar que una estrategia de expulsión dada, ya sea utilizando plazos de comienzo o de conclusión, planificando la tarea de plazo más cercano minimiza la cantidad de tareas que fallan.
- La expulsión es el otro factor crítico de diseño.
  - Cuando se especifican plazo de comienzo, tiene sentido una planificación no expulsiva. En ese caso, la tarea deberá bloquearse una vez que complete la parte crítica u obligatoria de su ejecución.
  - Para plazos de conclusión es más apropiada una planificación expulsiva. A veces es necesario sacar una tarea de ejecución para que otra pueda terminar, y así ambas cumplen con sus plazo.

Planificación de Tiempo Real

Planificación por plazos. *(cont.)*

- Como ejemplo de planificación de tareas periódicas con plazos de conclusión, plantearemos un sistema que recolecta y procesa datos de dos sensores, A y B.
- El plazo para el sensor A es cada 20ms, y para el B, cada 50ms.
- El tiempo de ejecución de la tarea A es de 10ms, mientras que de la tarea B es de 25ms.
- El planificador toma decisiones cada 10ms.
- La tabla siguiente muestra el perfil de ejecución de las tareas.

Planificación de Tiempo Real

Planificación por plazos. *(cont.)*

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
•	•	•	•
•	•	•	•
•	•	•	•

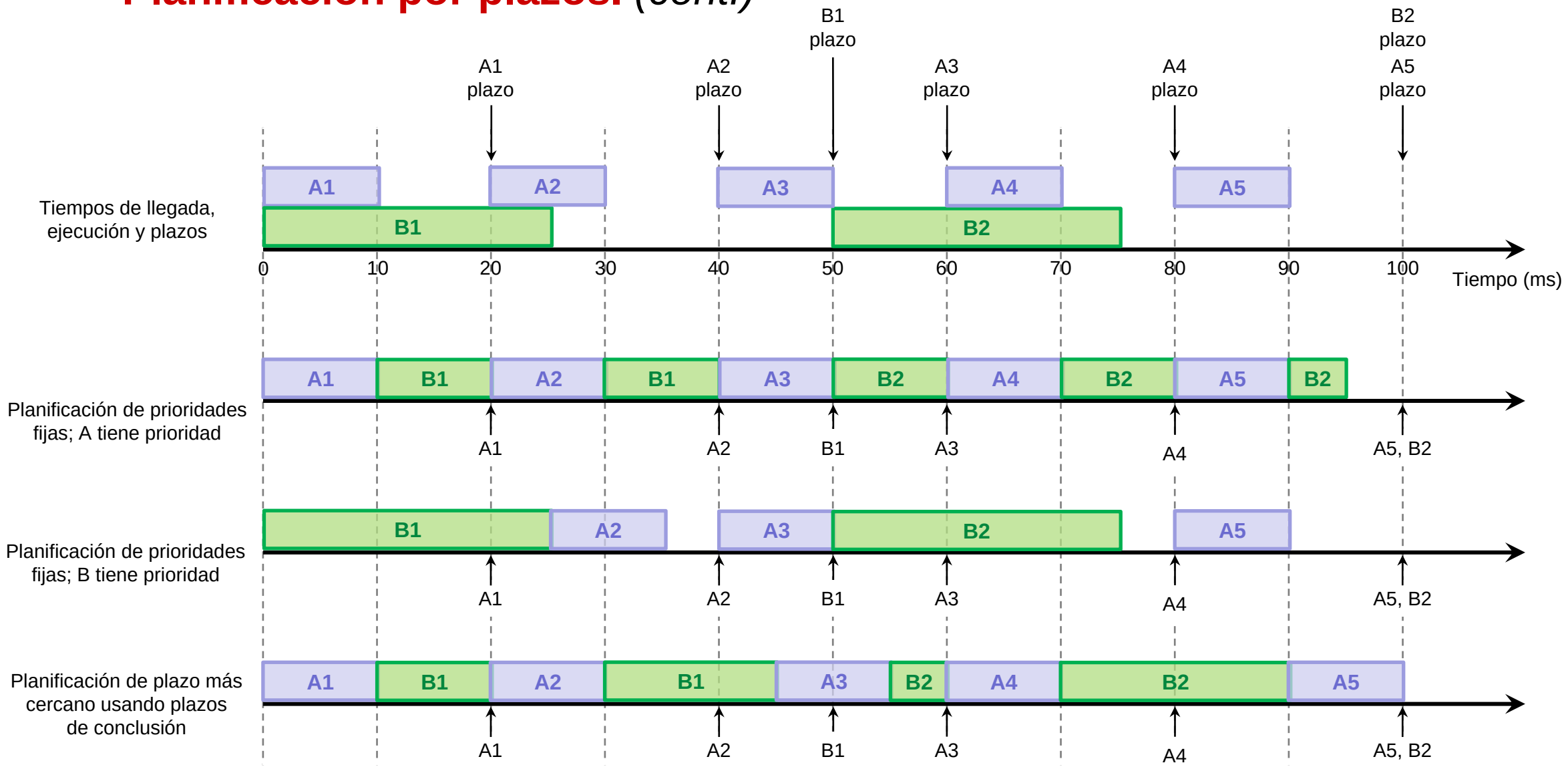
Planificación de Tiempo Real

Planificación por plazos. *(cont.)*

- En el gráfico siguiente se representan, en diagramas temporales, el perfil de ejecución de las tareas y distintas estrategias de planificación.
- El segundo y tercer diagrama muestran el resultado de planificar por prioridades fijas, primero con la tarea A, y luego con la tarea B.
- El último diagrama muestra la planificación por plazo más cercano. Planificando con el criterio de dar prioridad a la tarea cuyo plazo de conclusión es el más cercano, se pueden satisfacer todos los requisitos del sistema.

Planificación de Tiempo Real

Planificación por plazos. *(cont.)*



Planificación de Tiempo Real

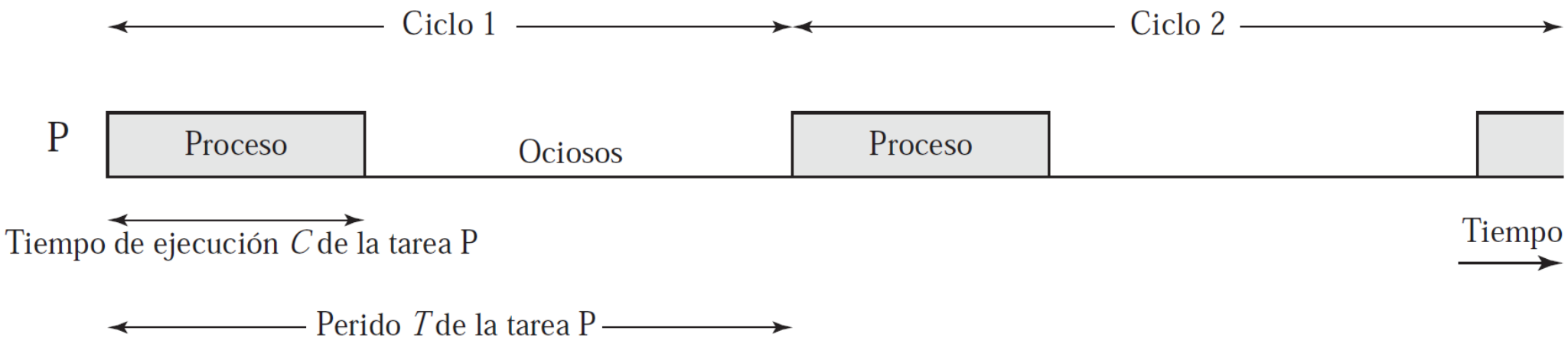
Planificación de tasa monótona.

- Uno de los métodos más prometedores para resolver los conflictos de planificación multitarea para tareas periódicas.
- También es conocida por su sigla en inglés **RMS** (Rate Monotonic Scheduling), fue planteada por primera vez en 1973, pero se empezó a implementar recién en 1994, siendo muy utilizada en ambientes industriales a partir de 1999.
- Para estudiar su funcionamiento veremos primero los parámetros más relevantes de una tarea periódica:
  - Período (*T*): es la cantidad de tiempo entre la llegada de una instancia de la tarea y la siguiente instancia de la misma. Se mide en segundos.
  - Tasa (*R*): es simplemente la inversa del período y se mide en Hertz.
  - Tiempo de ejecución (*C*): cantidad de tiempo de procesamiento que necesita la tarea.

Planificación de Tiempo Real

Planificación de tasa monótona. *(cont.)*

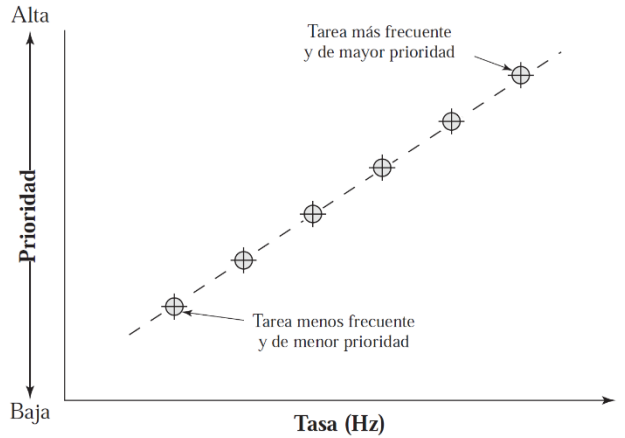
- Si el sistema es monoprocesador, el tiempo de ejecución no puede ser mayor que el período, por lo que se debe cumplir que $C \leq T$.
- Si la tarea siempre ejecuta hasta que concluye, es decir, nunca se le deniegan los recursos que solicita, la utilización de procesador por parte de esta tarea será:
  - Utilización de procesador (U): $U = \frac{C}{T}$



Planificación de Tiempo Real

Planificación de tasa monótona. *(cont.)*

- Para RMS la tarea de mayor prioridad es la que tiene el período más breve, la segunda prioridad será la de la que tenga el segundo período más breve, y así sucesivamente.
- Cuando hay más de una tarea disponible para su ejecución, la que tenga el período más breve se sirve primero.
- Si se grafica la prioridad de las tareas en función de su tasa, se obtiene una función monótona creciente; de ahí viene el nombre del método.



Planificación de Tiempo Real

Planificación de tasa monótona. *(cont.)*

- Una medida de la efectividad de esta planificación es si garantiza o no el cumplimiento de todos los plazos duros.
- Suponiendo *n* tareas con períodos y tiempo de ejecución fijos, es posible cumplir los plazos si se cumple la siguiente relación:

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \cdots + \frac{C_n}{T_n} \leq 1 \qquad (1)$$

- Esto significa que la suma de la utilización del procesador de las tareas no puede ser mayor a 1, que es la utilización total.
- Esta ecuación proporciona un límite al número de tareas que un algoritmo perfecto puede planificar satisfactoriamente.

Planificación de Tiempo Real

Planificación de tasa monótona. *(cont.)*

- Para RMS se puede demostrar que se cumple la siguiente relación:

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

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

Planificación de Tiempo Real

Planificación de tasa monótona. *(cont.)*

- La tabla anterior muestra algunos valores para el límite superior. A medida que la cantidad de tareas crece, la planificabilidad tiende al valor 0,693, lo que significa que la máxima utilización del procesador será del 69,3% cuando la cantidad de tareas sea “infinita”.
- En la práctica, se suelen conseguir utilizaciones de procesador tan altas como el 90%, mayores que los de la ecuación 2.
- La ecuación 1 también se cumple para planificación con plazo más cercano, con lo cual se puede conseguir una mayor utilización del procesador.
- Sin embargo, en la práctica se demostró que RMS es más estable, ya que permite cumplir con los plazos duros a pesar de los errores transitorios y la sobrecarga del sistema.

Fin del Módulo III