



Sistemas Operativos Avanzados

Tema 3. Planificación / Scheduling

Profesor:
Dr. José Octavio Gutiérrez García

octavio.gutierrez@itam.mx

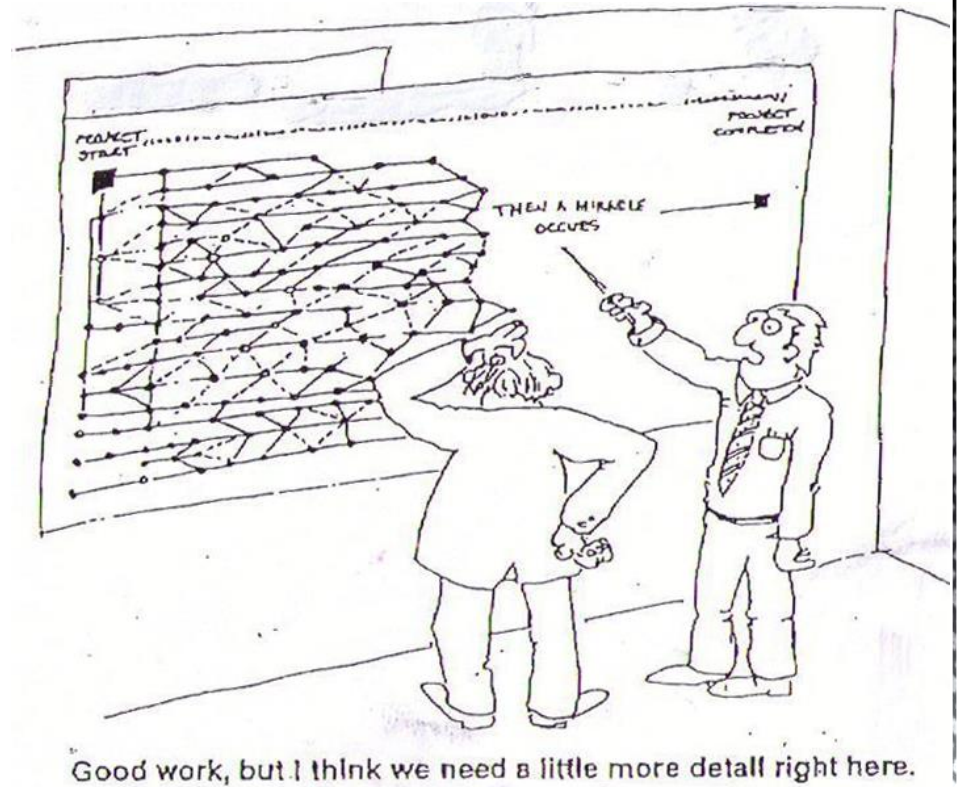


Planificación mono-procesador

- El objetivo de la planificación de un procesador es asignar a los procesos tiempo de ejecución de manera que cumpla con los **objetivos del sistema**.
 - Tiempo de respuesta
 - Rendimiento
 - Eficiencia del procesador

Objetivos de la planificación

- Compartir tiempo equitativamente entre procesos
- Prevenir el hambre de un proceso
- Utilizar el procesador de manera eficiente
- Tener bajo “overhead”
- Priorizar los procesos



Planificación a largo plazo

- Determina qué **programas** son **admitidos** en el sistema para procesamiento, para lo cual carga el programa, sus datos y crea los procesos. Basado en:
 - FIFO
 - Prioridades
 - Tiempo esperado de ejecución
- Entre más procesos, menor es el porcentaje de tiempo que cada proceso es ejecutado



Planificación a mediano plazo

- Decide qué programas **sacar de memoria** y llevar al disco (**swap-out**)
- Ejemplo: procesos inactivos o activos cuyo estado sea bloqueado momentáneamente y luego, cuando desaparezcan las causas de su bloqueo, traerlos nuevamente a memoria (**swap-in**) para continuar su ejecución.

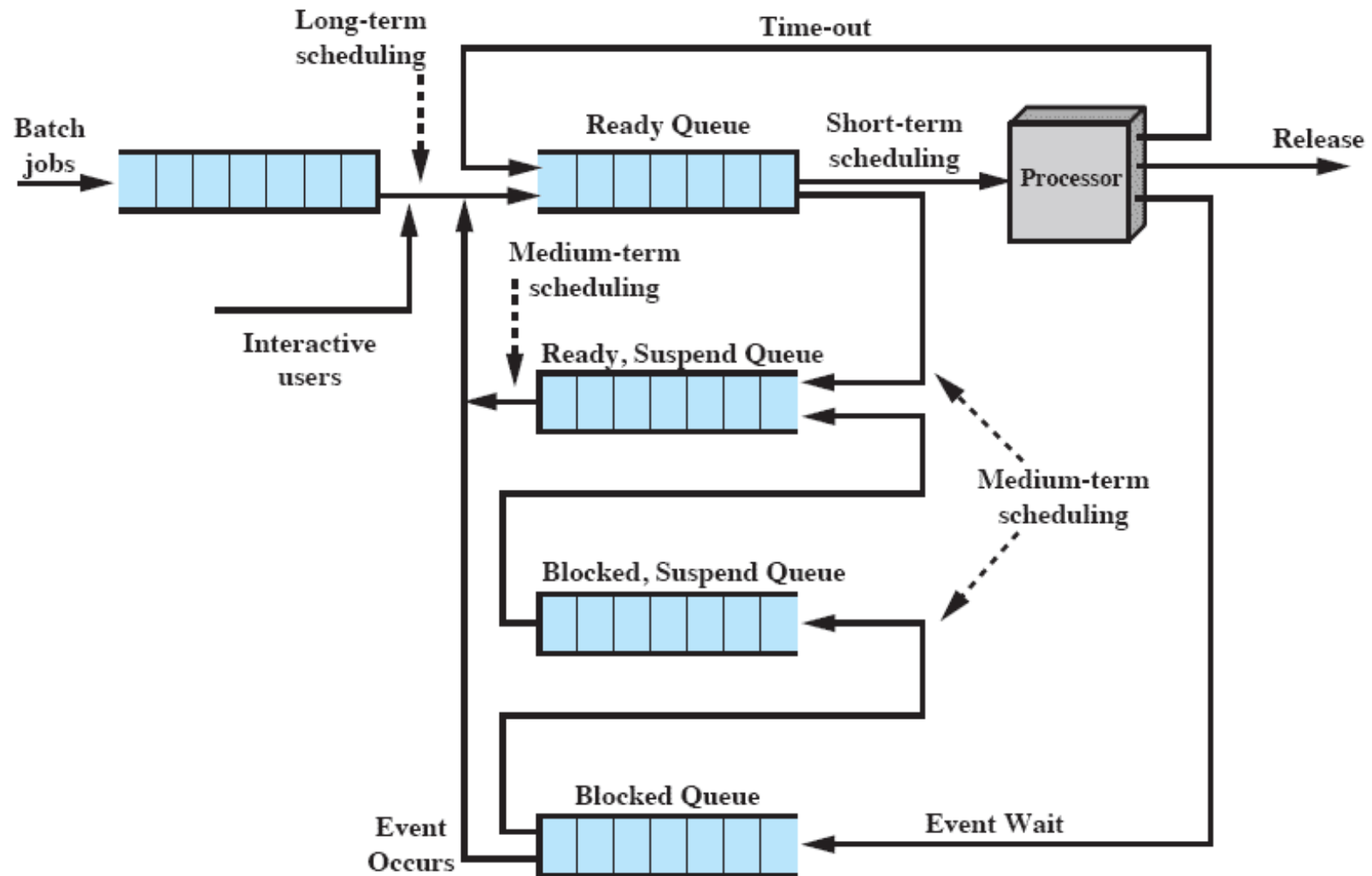


Planificación a corto plazo



- **Responsable** de decidir quién, cuándo y por cuánto **tiempo** recibe el **procesador** a un proceso que está preparado en la *Ready Queue*.
- También maneja las **interrupciones**

Interacción de planificadores



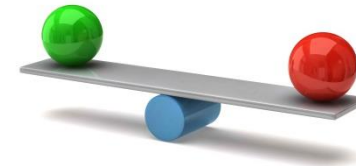
Criterios para evaluar el rendimiento

- **Turnaround time**: intervalo de tiempo entre la entrada y salida de un proceso.
- **Tiempo de respuesta**: intervalo de tiempo en el que la respuesta de una solicitud empieza a ser atendida.
- Cumplimiento de **fechas de entrega**.

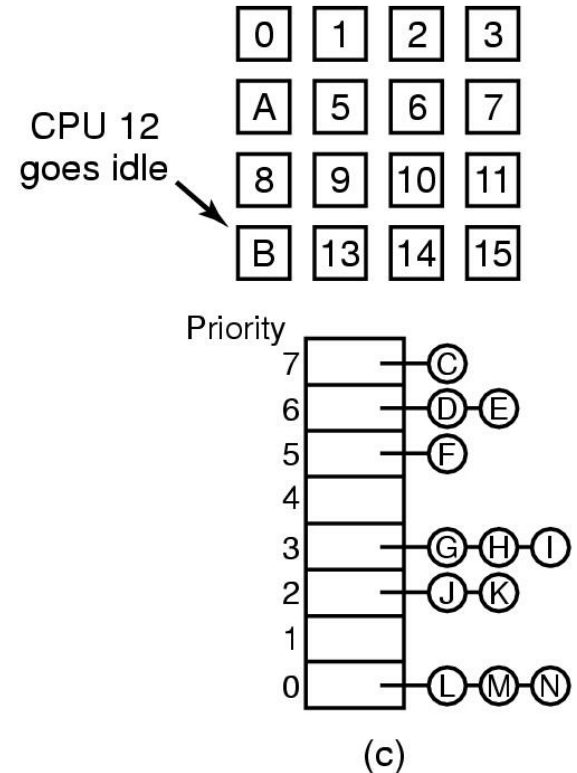
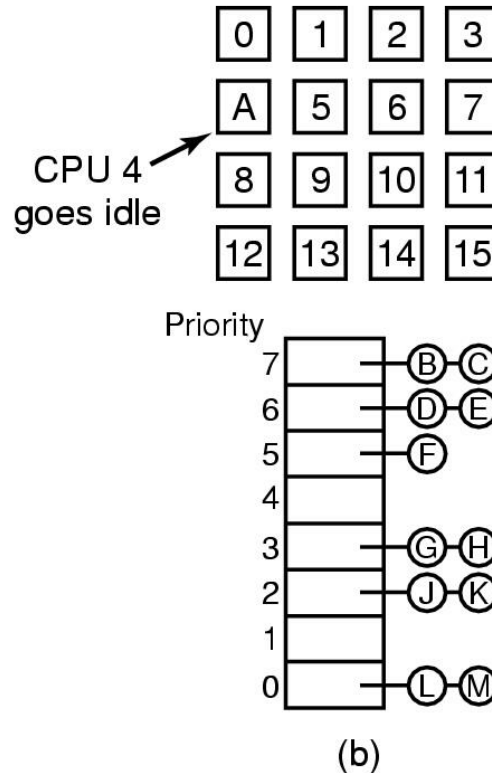
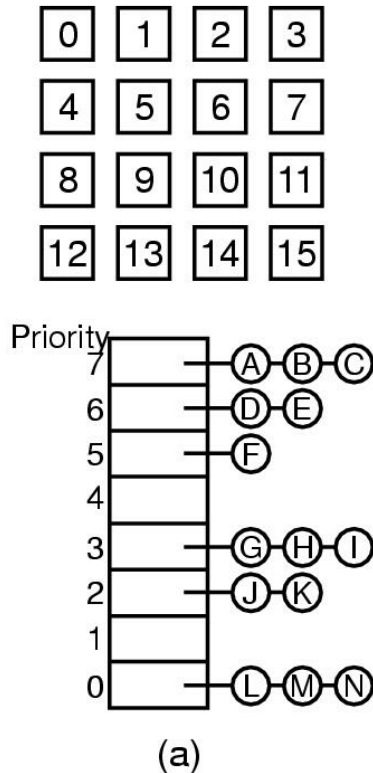


Criterios para evaluar el rendimiento

- *Utilización del procesador*
- *Justicia*
- *Imponer prioridades*



Planificación multi-procesador



Clasificación de sistemas multi-procesador

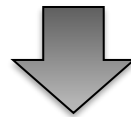
- Procesadores *ligeramente acoplados*.
Cada uno tiene su memoria y canales de E / S
- Procesadores *fuertemente acoplados*.
Los procesadores comparten la memoria principal



Planificación multi-procesador: Granularidad

- Frecuencia de sincronización, entre los procesos en un sistema.

Paralelismo Independiente (sin sincronización)



Paralelismo de grano fino (con alta sincronización)



Planificación multi-procesador

- Programación de un multiprocesador implica tres cuestiones relacionadas entre sí:
 - La asignación de **procesos a procesadores**
 - El uso de **multiprogramación** en los procesadores individuales
 - Despacho (**Dispatching**) **real** de un proceso
 - Cambio de contexto: corre hilo, guarda su estado, carga otro hilo.

Otras cuestiones a considerar:

- El nivel de granularidad y
- El número de procesadores disponibles



Asignación de procesos a procesadores

- Si todos los procesadores son iguales, es más simple tratar a los procesadores como un *recurso común* y asignar procesos a los procesadores on-demand.



¿Asignación estática o dinámica?

■ **Asignación Dinámica:**

- Los hilos se mueven de una cola para un procesador a una cola para otro procesador.

■ **Asignación Estática:**

- Permanentemente se asigna un proceso a un procesador.
- Cola de corto plazo para cada procesador
- Menos overhead
- Permite el uso de "grupo" o la programación "gang"
- **PROBLEMA:** puede dejar un procesador inactivo, mientras que otros tienen un retraso



Modelos arquitectónicos de asignación de procesos a procesadores

■ Maestro / esclavo

- Funciones clave del Kernel siempre se ejecutan en un procesador
- **Nodo maestro** responsable de la planificación
- **Nodo esclavo** envía solicitud de servicio al maestro

■ *Desventajas*

- Un solo punto de fallo.
- Maestro se convierte en cuello de botella.



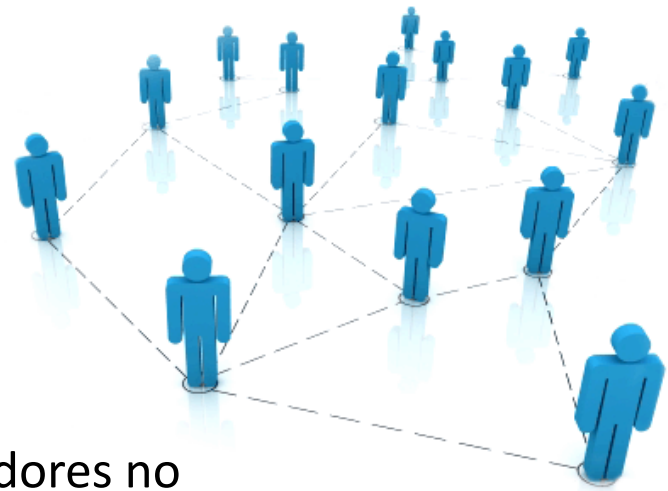
Modelos arquitectónicos de asignación de procesos a procesadores

■ Peer to Peer

- Kernel se puede ejecutar en cualquier procesador
- Cada procesador tiene su propio planificador

■ *Desventajas*

- Complica el sistema operativo
- Hay que asegurarse de que los procesadores no elijan el mismo proceso



Planificación en tiempo real

- El correcto funcionamiento del sistema depende *no sólo* del *resultado* lógico del cálculo, *sino* también del *momento en que se producen* los resultados



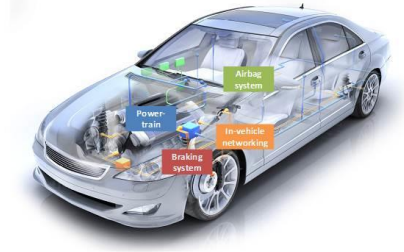
Sistemas en tiempo real

- Robótica
- Control de tráfico aéreo
- Telecomunicaciones
- Sistemas de mando y control militar

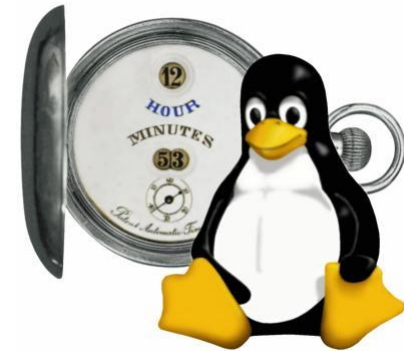


Características de Sistemas Operativos de Tiempo Real

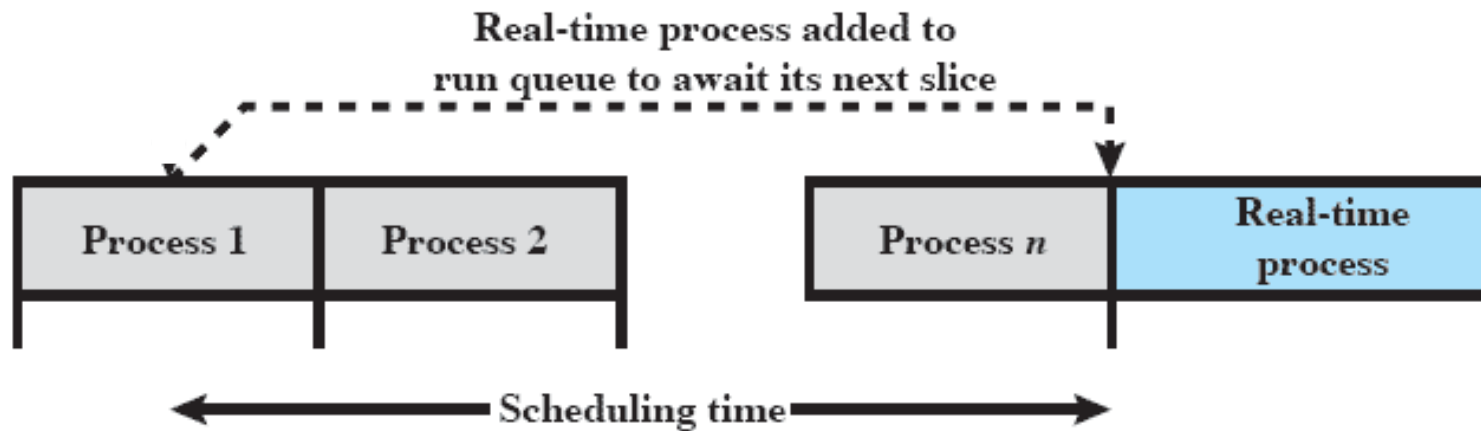
- Cambio rápido de procesos en ejecución
- Normalmente pequeños (sistemas empuotrados)
- Capacidad para responder a las interrupciones externas rápidamente
- Multitarea con sincronización
- “Preemptive” planificación basada en prioridades
- Retraso de tareas por un tiempo determinado



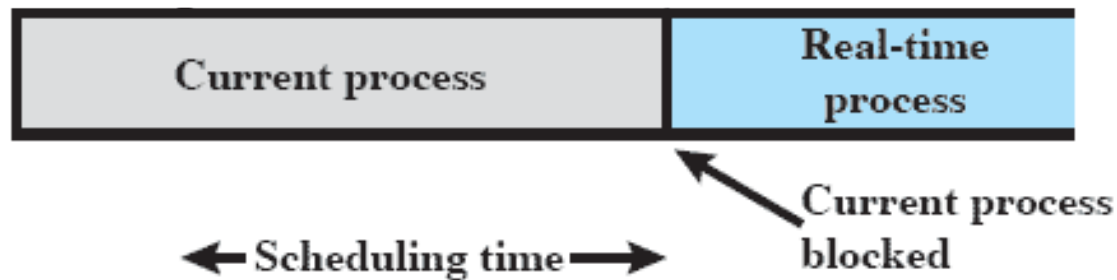
RTLinux



¡Planificación Round-Robin Inaceptable!

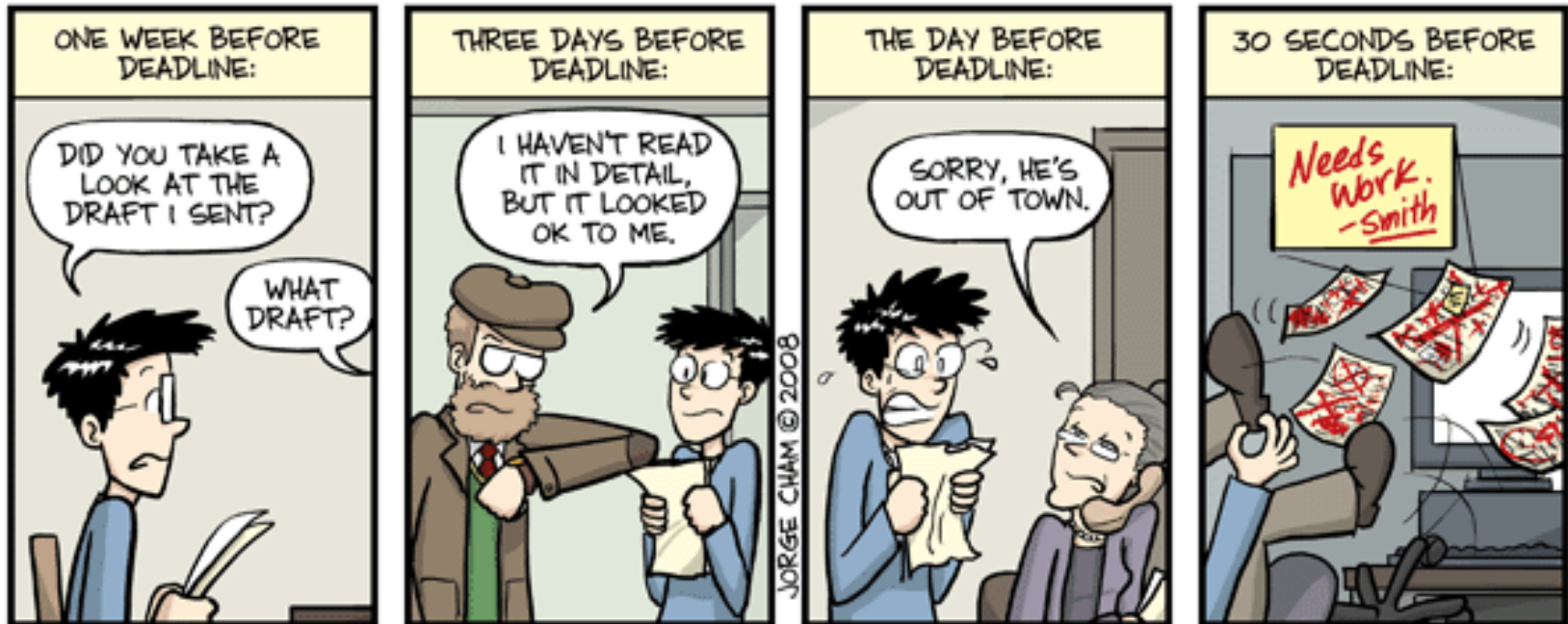


¡Planificación no *preemptive* basada en prioridades **Inaceptable!**



Deadline scheduling

- Las aplicaciones en tiempo real **no** tienen que ver (del todo) con la **velocidad**, sino con la realización de las tareas **antes** de cierto **tiempo**.



Información usada por Deadline Scheduling

- Ready time
- Starting deadline
- Completion deadline
- Processing time
- Resource requirements
- Priority