

Software en Tiempo Real

Conceptos Básicos

Mg. G. Friedrich

1

Consideraciones generales

- Deben responder a eventos externos, asegurando un tiempo máximo de respuesta determinado.
- La secuencia de ejecución no sólo está determinada por decisiones del sistema, sino también por eventos que ocurren en el mundo real.
- Habitualmente requieren interfaces con una gran cantidad de sensores y actuadores.
- Generalmente son complejos.

Mg. G. Friedrich

2

Consideraciones generales

- Deben ser muy confiables. Su respuesta debe ser controlada, incluso en condiciones de sobrecarga.
- No pueden “volver atrás” y reiniciar desde un contexto preexistente.
- Los requerimientos del medio ambiente por lo general son en paralelo, provocando problemas de planificación y prioridades.

Mg. G. Friedrich

3

Consideraciones generales

- Las restricciones de tiempo hacen que la demostración de correctitud funcional no alcance para garantizar su desempeño en condiciones reales.
- Son de “tiempo infinito”, por lo que deben poder recuperarse automáticamente de condiciones de excepción.

Mg. G. Friedrich

4

Software de Tiempo Real

- Un Sistema de Tiempo Real debe:

- ejecutar las tareas que tiene asignadas,
- respetando las metas temporales de c/u de ellas.

Es decir: en un STR no solamente es importante que las tareas se realicen, sino que también deben cumplirse en los plazos de tiempo previstos.

Mg. G. Friedrich

5

Software de Tiempo Real

- Típicamente un STR interactúa con su entorno

- recibe información (eventos) del medio ambiente
- realiza acciones para producir cambios en dicho entorno.
- En estos casos se los denomina sistemas **REACTIVOS**, ya que reaccionan ante estímulos del medio ambiente.

Mg. G. Friedrich

6

Software de Tiempo Real

- Típicamente un STR no maneja una única tarea, sino varias simultáneamente.

→ estrategias para analizar y poder predecir el cumplimiento de las metas temporales.

⇒ planificación (scheduling)

- Un Sistema de Tiempo Real debe ser **PREDECIBLE**

- Un error común es asociar tiempo real con velocidad.

7

Software de Tiempo Real

- Aplicaciones:

- Control de Procesos Industriales
- Control de Aeronaves en vuelo
- Sistemas de Armas
- Control de Redes de Comunicación
- Procesamiento de Señales
- Electrónica del automóvil
- Robótica
- Control de Plantas Nucleares

→ En muchos casos son aplicaciones CRITICAS

- Vidas humanas
- \$\$\$

8

Software de Tiempo Real

- Clasificación: → Según las restricciones temporales

- **Sistemas de tiempo real duro** (hard real-time)
 - Los límites de tiempo son estrictos
 - El no cumplimiento puede tener consecuencias más o menos graves
 - En algunos casos puede ser preferible un trabajo imperfecto pero terminado a tiempo
 - Ej.: control de un reactor nuclear.
- **Sistemas de tiempo real blando** (soft real-time)
 - Los límites de tiempo son flexibles.
 - Ej.: sistema de reserva de pasajes.

9

Software de Tiempo Real

- Clasificación: → Según las restricciones temporales

- **Sistemas de tiempo real firme** (firm real-time)

Sistemas de tiempo real duro que pueden tolerar pérdidas, si la probabilidad de ocurrencia de las mismas es baja.

Mg. G. Friedrich

10

Software de Tiempo Real

- Clasificación: → Según las escalas de tiempo

- **Basados en reloj**
 - El pasaje del tiempo
 - P. ej.: sistemas periódicos.
- **Basados en eventos**
 - P. ej.: las acciones se inician a partir del cierre de una llave, o la lectura de un sensor.
- **Interactivos**
 - P. ej.: un operador ingresando datos (vago)

Mg. G. Friedrich

11

Software de Tiempo Real

- Clasificación: → Según la integración con el sistema físico (o con el medio)

- **Embebidos (o empotrados) (Embedded)**
 - P. ej.: sistema de control de inyección de combustible de un automóvil.
- **No embebidos**
 - **Orgánicos**
 - Independientes del hardware en que corren.
 - **Debilmente acoplados**
 - Pueden correr en otro hardware reescribiendo ciertos módulos.

12

Software de Tiempo Real

- Clasificación:

→ Según la forma de procesamiento

- **Sistemas centralizados**
 - Un único nodo (mono o multiprocesador) encargado de atender a todas las tareas.
 - Las tareas se comunican a través de memoria compartida (el gasto de tiempo en comunicación es insignificante)

Mg. G. Friedrich

13

Software de Tiempo Real

- Clasificación:

→ Según la forma de procesamiento

- **Sistemas distribuidos**
 - Varios nodos, unidos a través de una red se reparten la atención de los distintos procesos.
 - Las tareas se comunican a través de la red, no hay memoria compartida (el gasto de tiempo en comunicación es importante).

Mg. G. Friedrich

14

Software de Tiempo Real

- Clasificación:

→ Según la estrategia de planificación de las tareas

- **Sistemas estáticos**
 - Todas las tareas, su naturaleza y características son conocidas de antemano, y en tiempo de diseño se planifica la ejecución de las mismas.
 - El sistema no admite la aparición de una nueva tarea sobre la marcha.
 - Bajo costo de ejecución.

Mg. G. Friedrich

15

Software de Tiempo Real

- Clasificación:

→ Según la estrategia de planificación de las tareas

- **Sistemas dinámicos (o adaptivos)**
 - Puede haber un conjunto de tareas conocido de antemano, pero ante la aparición de una nueva tarea, el sistema analiza si la puede garantizar sin afectar a las tareas que ya maneja, y en ese caso la agrega a la lista de tareas.
 - Mayor costo de ejecución, porque el planificador tiene un trabajo de análisis adicional.

Mg. G. Friedrich

16

Software de Tiempo Real

Los conceptos de sistemas de tiempo real no solo se aplican a sistemas de computación, sino que la misma teoría se aplica a otras áreas, como por ejemplo en REDES.

Conceptos y modelos matemáticos similares a los usados para analizar si un procesador puede ser capaz de ejecutar a tiempo un conjunto de tareas, se usan para analizar si una determinada red podrá o no despachar a tiempo los requerimientos de transmisión de los distintos nodos.

Mg. G. Friedrich

17

Planificación de Tareas en Sistemas de Tiempo Real

Mg. G. Friedrich

18

Planificación de tareas en Sistemas de T. Real

Conceptos y Definiciones:

- En un sistema de tiempo real típico coexisten varias tareas (multitasking)
- El despachador debe hacer ejecutar la tarea lista (ready) de mayor prioridad.
- Las prioridades pueden ser fijas (planif. estática) o variables (planif. dinámica).

Mg. G. Friedrich

19

Planificación de tareas en Sistemas de T. Real

- Definiciones:

- **TIEMPO LIMITE / META (DEADLINE)**: instante tope para finalizar la ejecución del último requerimiento de una tarea.
Si la tarea es periódica, es el instante en el cual llegará el próximo requerimiento.
- **TIEMPO DE RESPUESTA**: tiempo que media entre el instante en que una tarea es requerida, y el instante en que dicha tarea ha sido completada.

Mg. G. Friedrich

20

Planificación de tareas en Sistemas de T. Real

- Definiciones:

- **INSTANTE CRITICO**: es aquel en el cual si una tarea es requerida tendrá su mayor tiempo de respuesta.

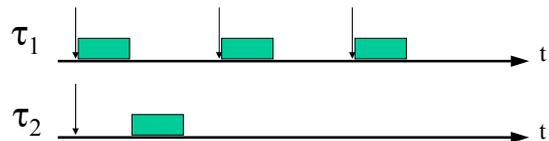
Según un teorema de Liu y Layland, se da un instante crítico para una tarea cuando la misma es requerida simultáneamente con todas las de mayor prioridad.

Mg. G. Friedrich

21

Planificación de tareas en Sistemas de T. Real

- **ZONA CRITICA**: intervalo de tiempo entre un instante crítico y la finalización de la tarea.



Mg. G. Friedrich

22

Planificación de tareas en Sistemas de T. Real

Si tenemos un sistema de tiempo real con N tareas, siempre se analiza el comportamiento del mismo a partir de un requerimiento simultáneo de las N tareas.

Esto corresponde a un instante crítico para las N-1 tareas de menor prioridad, lo cual representa el peor caso.

Si el sistema tiene una planificación factible en ese caso, lo tendrá en cualquier otra condición.

Mg. G. Friedrich

23

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973)

- Sistema de tiempo real con m tareas.
- Todas las tareas son periódicas.
- Todas las tareas son independientes:
 - una tarea no depende de otra para ejecutarse
 - no hay recursos compartidos entre ellas
 - ⇒ una tarea no puede bloquear a otra de mayor prioridad.
- La planificación es con desalojo

24

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973)

- Se define el **FACTOR DE UTILIZACION** como:

$$U = \sum (C_i / T_i)$$

donde: C_i : tiempo de ejecución de la tarea i

T_i : período entre dos requerimientos consecutivos de la tarea i

- Condición necesaria: $U \leq 1$

Mg. G. Friedrich

25

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973) - **PMC**

Primer propuesta: PLANIFICACION ESTATICA

- Mediante teoremas se demuestra que la forma más adecuada para asignar prioridades a las tareas es en función inversa al período entre requerimientos.

Es decir: Mayor prioridad a las tareas de menor período entre requerimientos.

⇒ **PRIORIDADES MONOTONICAS CRECIENTES**

Mg. G. Friedrich

26

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973) - **PMC**

- Mediante teoremas L&L demuestran que esta asignación de prioridades es **óptima**, es decir:

Si usando algún otro esquema de asignación **estática** de prioridades se obtiene una planificación factible, entonces se puede asegurar que también será factible usar PMC.

Mg. G. Friedrich

27

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973) - **PMC**

- Mediante teoremas L&L demuestran que la condición suficiente para la factibilidad del sistema es :

$$U = \sum (C_i / T_i) \leq m (2^{(1/m)} - 1)$$

donde: m : cantidad de tareas

Mg. G. Friedrich

28

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973) - **PMC**

$$U = \sum (C_i / T_i) \leq m (2^{(1/m)} - 1)$$

Si $m = 2 \Rightarrow U \leq 0.828$

$m = 3 \Rightarrow U \leq 0.779$

$m = 4 \Rightarrow U \leq 0.757$

$m \rightarrow \infty \Rightarrow U \leq \ln(2) = 0.69$

Mg. G. Friedrich

29

Planificación de tareas en Sistemas de T. Real

Liu y Layland (1973) - **PMC**

- Ejemplo: dado $S(C_i, T_i) = \{ (1, 3), (2, 5), (1, 9) \}$

$$U = \sum (C_i / T_i) = 1/3 + 2/5 + 1/9 = 0.70996$$

$$m (2^{(1/m)} - 1) = 3 (2^{(1/3)} - 1) = 0.77976$$

$U < m (2^{(1/m)} - 1) \Rightarrow$ Planificación Factible

Mg. G. Friedrich

30

Planificación de tareas en Sistemas de T. Real

Rueda Cíclica Justa (RCJ) (Round Robin)

- Es un esquema de planificación muy simple
- Las tareas van siendo despachadas en el orden en que arriban los requerimientos (FIFO).
- Condición de factibilidad: $\sum C_i < \text{mínimo}(T_i)$

- Ejemplo: dado $S(C_i, T_i) = \{ (1, 3), (2, 5), (1, 9) \}$

$$\sum C_i = 1 + 2 + 1 = 4 > \text{mínimo}(T_i) = T_1 = 3$$

⇒ Planificación NO Factible

31

Factibilidad de la planificación de tareas por PMC

Cuando el factor de utilización $U = \sum (C_i / T_i)$ es tal que

$$m(2^{(1/m)} - 1) < U \leq 1$$

Se puede analizar si el sistema igualmente tiene una planificación factible. Hay varios trabajos al respecto, de los cuales vamos a analizar dos:

- Cálculo de la Función Trabajo en Puntos Críticos (Lehoczky, Sha y otros)
- Método de las ranuras libres (Santos y otros -UNS-) 32