1. CONCEPTO DE SISTEMAS DE TIEMPO REAL. MEDIDAS DE TIEMPO Y MODELO DE TAREAS

1.1. DEFINICIÓN, TIPOS Y EJEMPLOS

La ejecución de un sistema de tiempo real debe producirse dentro de unos plazos de tiempo predefinidos para tener garantía.
“Un sistema de tiempo real es aquel cuyo funcionamiento correcto depende no solo de resultados lógicos producidos por el mismo, sino también del instante de tiempo en el que se producen esos resultados” Stankovic1997.

Corrección funcional + corrección temporal

→ El no cumplimiento de una restricción temporal lleva a un fallo del sistema.

Tipos de Sistermas de Tiempo Real (STR)

-Sistema en línea: Siempre está disponible, pero no se garantiza una respuesta en un intervalo de tiempo acotado. (Cajero automático).
-Sistema interactivo: Suele ofrecer una respuesta en un tiempo acotado, aunque no importa si ocasionalmente tarda más. (Juego).
-Sistemas de respuesta rápida: El sistema ofrece una respuesta en un tiempo acotado lo más corto posible. (Sistema anti incendios, ABS del coche).
Todos presentan las siguientes características:

- Sistema en línea.
- Sistema interactivo
- Sistema de respuesta rápida

1.2. PROPIEDADES DE LOS STR

Al depender la corrección del sistema de las restricciones temporales, los STR deben cumplir:

Reactividad

Un sistema tiene que interaccionar con el entorno, y responder de la manera esperada a los estímulos externos dentro de un intervalo de tiempo previamente definido.

Predecibilidad

La ejecución del sistema tiene que ser determinista, garantizando su ejecución dentro del plazo de tiempo definido.

- Las respuestas han de producirse dentro de las restricciones temporales impuestas por el entorno.
- Es necesario conocer el comportamiento temporal de los componentes software y hardware utilizados, así como el lenguaje de programación.
- Si no se puede tener un conocimiento temporal exacto, hay que definir marcos de tiempo acotados (tiempo máximo de acceso a un dato E/S).

Confiabilidad

Mide el grado de confianza de un sistema. Depende de:

- Disponibilidad (avalaibility). Capacidad de proporcionar servicios siempre que se solicita.
- Robustez o tolerancia a fallos (fault tolerant). Capacidad de operar en situaciones excepcionales sin poseer un comportamiento catastrófico.
- Fiabilidad (reliability): Capacidad de ofrecer siempre los mismo resultados bajo las mismas condiciones.

- Seguridad: Capacidad de protegerse ante ataques o fallos accidentales o deliberados (safety), y a la no vulnerabilidad de los datos (security).

STR críticos → safety-critical system
A veces la confiabilidad es crítica, es decir, su no cumplimiento lleva a pérdida humana o económica. (sistemas de aviónica o automoción, centrales nucleares…)

1.3. MODELO DE TAREAS

Tareas en un STR

Un STR se estructura en tareas que acceden a los recursos del sistema. Una tarea es un conjunto de acciones que describen el comportamiento del sistema o parte de él en base a la ejecución secuencial de un trozo de código. (Equivalente a proceso o hebra).

- La tarea satisface una necesidad funcional concreta.
- La tarea tiene definida unas restricciones temporales a partir de los Atributos Temporales.
- Una tarea activada es una tarea en ejecución o pendiente de ejecutar.
- Decimos que una tarea se activa cuando es necesario ejecutarla una vez.
- El instante de activación de una tarea es el instante de tiempo a partir del cual debe ejecutarse.

Recursos en un STR. Atributos

Son elementos disponibles para la ejecución de las tareas:

- Recursos activos: Procesador, red…
- Recursos pasivos: Datos, memoria, Dispositivos E/S…

Asumimos que cada CPU disponible se dedica a ejecutar una o varias tareas, de acuerdo al esquema de planificación de uso.

- Si una CPU ejecuta más de una tarea, el tiempo de la CPU debe repartirse entre varias tareas.

Los requerimientos de un STR obligan a asociar un conjunto de atributos temporales a cada tarea de un sistema. Estos atributos son restricciones acerca de cuando se ejecuta activa cada tarea y cuánto puede tardar en completarse desde que se activa.

Planificación de tareas

Es una labor de diseño que determina cómo se le asignan a cada tarea los recursos activos de un sistema, de forma que garantice el cumplimiento de las restricciones dadas por los atributos temporales de la tarea.

Atributos temporales de una tarea
-Tiempo de cómputo o ejecución (C): Tiempo necesario para la ejecución de la tarea.
-Tiempo de respuesta (R): Tiempo que ha necesitado el proceso para completarse totalmente a partir del instante de activación.
-Plazo de respuesta máxima (D): Define el máximo tiempo de respuesta posible.
-Periodo (T): Intervalo de tiempo entre dos activaciones sucesivas.

Tipos de tareas según la recurrencia de sus activaciones

-Aperiódica: Se activa en instantes arbitrarios.
-Periódica: Repetitiva, T es el tiempo exacto entre activaciones.
-Esporádica: Repetitiva, T es el intervalo mínimo entre activaciones.

Diseño de la planificación de tareas

El problema de la planificación supone:
- Determinar los procesadores disponibles para asociar las tareas.
- Determinar las relaciones de dependencia de las tareas.
    - Relaciones de precedencia entre distintas tareas.
    - Recursos comunes a los que acceden distintas tareas.
- Determinar el orden de ejecución de las tareas para garantizar las restricciones especificadas.

Esquema de planificación de tareas

Hay que cubrir los siguientes aspectos:
- Un algoritmo de planificación, que define un criterio que determina el orden de acceso de las tareas a los distintos procesadores.
- Un método de análisis que permite predecir el comportamiento temporal del sistema, y determina si la planificabilidad es factible bajo las condiciones o restricciones especificadas.
    - Se pueden comprobar si los requisitos temporales están garantizados en todos los casos posibles.
    - Se estudia el peor comportamiento posible (WCET).

Cálculo del WCET

Suponemos que siempre se conoce el valor WCET (C_W), que es el máximo valor de C para cualquier ejecución posible de dicha tarea. Hay dos formas de obtener C_W :

- Medida indirecta del tiempo de ejecución (en el peor caso) en la plataforma de ejecución: Se realizan múltiples experimentos y se hace una estadística.
- Análisis del código, se basa en calcular el peor tiempo:
    - Se descompone el código en un grafo de bloques secuenciales.
    - Se calcula el tiempo de ejecución de cada bloque.
    - Se busca el camino más largo.

Asumimos que se tarda lo mismo en ejecutar una tarea determinada → $C = C_W$

Restricciones temporales de una tarea

Para determinar la planificación del sistema necesitamos conocer las restricciones temporales de cada tarea del sistema.

Ejemplo: C = 10ms; T = 100ms; D = 90ms
Tarea periódica que se activa cada 100ms, se ejecuta en cada activación un máximo de 10ms y desde que se activa hasta que concluye no pueden pasar más de 90ms.

Las restricciones temporales para un conjunto de n tareas periódicas se especifican dando una tabla con los valores de T_i , C_i y D_i para cada una de ellas.

- La i-ésima tarea ocupa una fracción C_W / C_W del timpo total de CPU.
- El factor de utilización U es la suma de esas fracciones para todas las tareas.
- En un hardware con p procesadores disponibles, si U > p el sistema no es planificable (alguna tarea no podrá acabar en su período).

2. ESQUEMAS DE PLANIFICACIÓN

Tipos de esquemas de planificación

Para un sistema monoprocesador:
Planificación estática off-line sin prioridades.
Planificación basada en prioridades de tareas:

- Estáticas: prioridades preasignadas, no cambian.
  - RMS (Rate Monotonic Scheduling): prioridad a la tarea con menor T.
  - DMS (Deadline Monotonic Scheduling): prioridad a la tarea con menor D.
- Dinámicas: pioridades cambiantes durante la ejecución.
  - EDF (Earliest Deadline First): prioridad a la tarea con el deadline más próximo.
  - LLF (Least Laxity First): prioridad a tarea de menor holgura.
    (tiempo hasta deadline - tiempo de ejecución restante)

2.1. PLANIFICACIÓN CÍCLICA

La planificación se basa en diseñar un programa (ejecutivo cíclico) que implementa un plan de ejecución (plan principal) que garantice los requerimientos temporales.

- El programa es un bucle infinito llamado ciclo principal.
- En cada iteración del bucle principal se ejecuta otro bucle acotado con k iteraciones. Cada iteración dura lo mismo y se ejecutan una o varias tareas. Este bucle es el ciclo secundario.
- El entrelazado de las tareas en cada iteración del ciclo principal es siempre el mismo. Una iteración i del ciclo secundario puede tener un entrelazado distinto a otra iteración j.

Duraciones del ciclo principal y el secundario

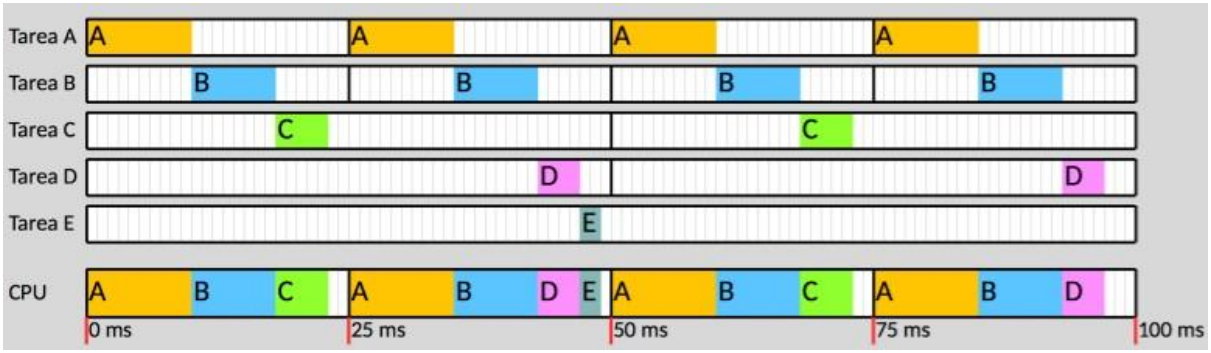
Las duraciones de ambos ciclos son valores enteros:

- La duración del ciclo principal se denomina hiperperiodo y se escribe como T_M
  Es el mínimo común múltiplo de todas las tareas.
- La duración del ciclo secundario se denomina T_S . Se debe cumplir $T_M = k*T_S$

Ejemplo de planificación cíclica

Tarea	T	C
A	25	10
B	25	8
C	50	5
D	50	4
E	100	2

T_M = mcm(25, 25, 50, 50, 100) = 100
Se descompone de k = 4 ciclos secundarios, con T = 25ms
Una posible planificación es la siguiente:




```
process EjecutivoCiclico;
   var inicio: time_point := now();
begin
   while true do begin
      A(); B(); C();        sleep_until(inicio + 25);
      A(); B(); D(); E();   sleep_until(inicio + 50);
      A(); B(); C();        sleep_until(inicio + 75);
      A(); B(); D();        sleep_until(inicio + 100);
      inicio = inicio + 100;
   end
end
```

Selección de la duración del ciclo secundario

Para seleccionar un valor apropiado de T_S se debe tener en cuenta:

Restricciones:

- El valor de T_S tiene que ser un divisor de T_M .
- El valor de T_S tiene que ser mayor o igual que el tiempo de cómputo © de cualquier tarea.

Sugerencia: Es aconsejable que T_S sea menor o igual que el mínimo deadline.

Propiedades de la Planificación Cíclica

No hay concurrencia en la ejecución:

- Cada ciclo secundario es una secuencia de llamadas a procedimientos.
- No se necesita un núcleo de ejecución multitarea

Los procedimientos pueden compartir datos.

- No se necesitan mecanismos de exclusión mútua.

No hace falta analizar el comportamiento temporal.

- El sistema es correcto por construcción.

Problemas de la planificación cíclica

Dificultad para ejecutar tareas con periodos largos.

Las tareas esporádicas son dificiles de tratar.

- Se puede utilizar un servidor de consulta.

El plan ciclo del proyecto es dificil de construir.

- Si los periodos son de diferentes ordenes de magnitud el número de ciclos secundarios se hace muy grande.
- Puede ser necesario partir una tarea en varios procedimientos.
- Es el caso más general de STR críticos.

Es poco flexible y difícil de mantener.

- Cada vez que cambia una tarea hay que rehacer toda la planificación.

2.2. PLANIFICACIÓN CON PRIORIDADES

Permite solventar los problemas descritos. Cada tarea tiene asociada una prioridad (entero positivo):

- Es un atributo de las tareas que depende de sus atributos temporales y/o el entrelazamiento entre ellas.
- Se asignan números mayores a tareas más urgentes.
- Las prioridades determinan que tarea puede ejecutarse en el procesador.
- Pueden ser estáticas (fijadas de antemano) o dinámicas (cambian con el tiempo).

Planificación de las tareas (scheduling)

Debe existir un componente software (planificador) capaz de:

- Asignar el procesador a una tarea activa o ejecutable (despachar).
- Desasignar la CPU a la tarea en ejecución (desalojar la tarea).
- Una tarea puede estar: suspendida, ejecutable o ejecutándose.
- Las tareas ejecutables se despachan en orden de prioridad.

Funcionamiento del planificador

El planificador actúa cuando una o más tareas se activan o cuando la tarea en ejecución termina. A continuación, se selecciona la tarea (A) con mayor prioridad (P_A) entre todas las ejecutables:

- Si la CPU está libre se ejecuta A.
- Si la CPU está ejecutando una tarea B:
    - Si $P_A > P_B \rightarrow$ A pasa a ejecutarse y B a estado ejecutable.
    - Si $P_A >= P_B \rightarrow$ no hay cambios

Al inicio todas las tareas se activan a la vez.

Planificación RMS (Rate Monotonic Scheduling)

Es un método de planificación estático on-line que asigna mayor prioridad a las tareas más frecuentes.

- A cada tarea i se le asigna una prioridad Pi basada en su período (Ti): A menor período mayor prioridad.
- Esta asignación es óptima en el caso de que todas las tareas sean periódicas, y el plazo máximo de respuesta D coincida con el período.

Test de planificabilidad

Los test de planificabilidad permiten determinar si el conjunto de tareas del sistema es planificable.

Test de planificabilidad suficientes:

- En caso de éxito en la aplicación del test el sistema es planificable.
- En caso contrario no tenemos información.

Test de planificabilidad exactos (suficiente y necesario)

- En caso de éxito es planificable.
- En caso contrario no es planificable.

Test de Liu & Layland

Es un test suficiente, determina la planificabilidad para un sistema de n tareas perióicas independientes con prioridades RMS.

- Se usan 2 valores reales, el factor de utilización (U), y el factor de utilización máximo (Uo(n)).

$$U \equiv \sum_{i=1}^{n} \frac{C_i}{T_i} \qquad U_0(n) \equiv n\left(\sqrt[n]{2} - 1\right)$$

- El valor de Uo(1) es 1 y va decreciendo al crecer n hasta el límite (ln 2 = 0.693147…)
- Un sistema pasa el test si el factor de utilización es menor o igual que el máximo posible U <= Uo(n)

Ejemplos

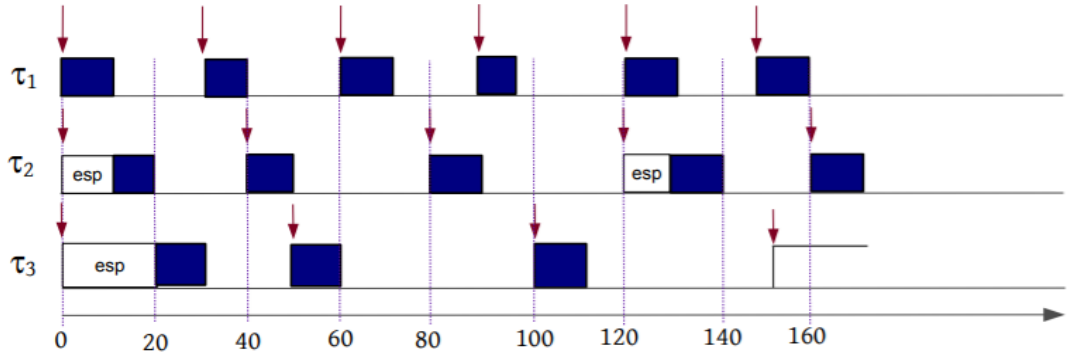
Tareas	C	D	T	C / T
T1	4	16	16	0.25
T2	5	40	40	0.125
T3	32	80	80	0.4

El sistema pasa el test ya que U = 0.775 <= 0.779 = Uo(3)

Tareas	C	D	T	C / T
T1	10	30	30	0.333
T2	10	40	40	0.25
T3	10	50	50	0.2

U = 0.783 > 0.779 = Uo(3)

El sistema no pasa el test. Para saber si es planificable hacemos el cronograma:



Planificación EDF (Earliest Deadline First)

Es un esquema de planificación con prioridades dinámicas.

Se asigna una prioridad más alta a la tarea que se encuentre más próxima a su plazo de respuesta máxima (deadline).

Características:

- En caso de igualdad se hace una elección no determinista de la siguiente tarea a ejecutar.
- Es más óptimo porque no es necesario que las tareas sean periódicas.
- Es menos pesimista que RMS.

Test de planificabilidad para EDF

Liu & Layland se puede aplicar también para EDF.

- En este caso, un sistema pasa el test si el factor de utilización (U) es menor que la unidad.
- Esto implica que la planificación EDF puede aplicarse a más sistemas que la planificación RMS.
- El test ahora es exacto:
    - Si pasa el test, es planificable con EDF
    - Si no pasa el tes, no es planificable con EDF

Ejemplo:

Tarea	C	D	T	C / T
T1	2	5	5	0.4
T2	4	7	7	0.571

U = 0.971 <= 1

El sistema es planificable con EDF (no lo es con RMS).