

PRÁCTICAS TIEMPO REAL EN SYS- BIOS

Sergio García Esteban 755844

Práctica 1

Medida de tiempos de ejecución. Análisis y diseño basado en asignación de prioridades primero al más urgente (DM). Estructuras básicas de programación tiempo real concurrente.

Apartado 1

El módulo *computos* ofrece la rutina CS() cuyo tiempo de cómputo es proporcional a su parámetro de llamada.

Para medir el tiempo de computo de una rutina, en el laboratorio lo podemos medir con ayuda del osciloscopio, en CSS lo hacemos con la ayuda de la herramienta *Duration*.

*Duration: Summary							
	Source	Count	Min	Max	Average	Total	Percent
1	CORTEX_M4_0, WCET	23	20004937	20005063	20,004,994.5	460,114,874	100.0

Apartado 2

El programa concurrente de este apartado consta de tres tareas periódicas T1, T2 y T3 activadas por retrasos que evolucionan de forma independiente (sin comunicaciones). Los períodos respectivos son: 100 ms, 150 ms y 350 ms. El plazo de respuesta es igual al periodo.

Hay que medir el tiempo de cómputo de cada una de estas tareas y asignar prioridades, dando prioridad al más urgente (DM), para completar la siguiente tabla.

Tarea	C	P	D	Prio
T1	20	100	100	3
T2	40	150	150	2
T3	100	350	350	1

Además, debemos comprobar que con los datos de la tabla se va a garantizar el cumplimiento de plazos.

$$C1 \leq D1$$

$$20 \leq 100$$

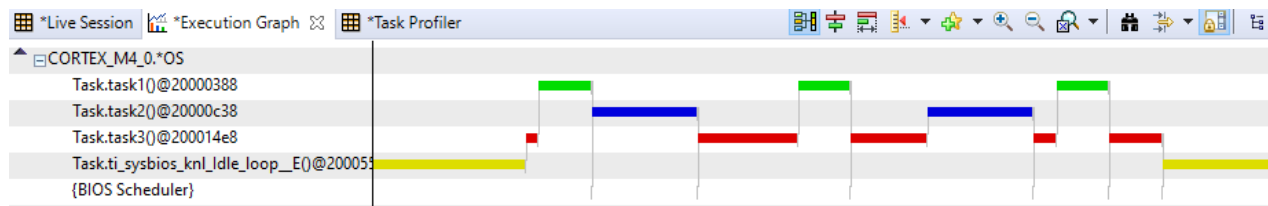
$$C2 + C1 \cdot [D2/P1] \leq D2$$

$$40 + 20 \cdot [150/100] = 80 \leq 150$$

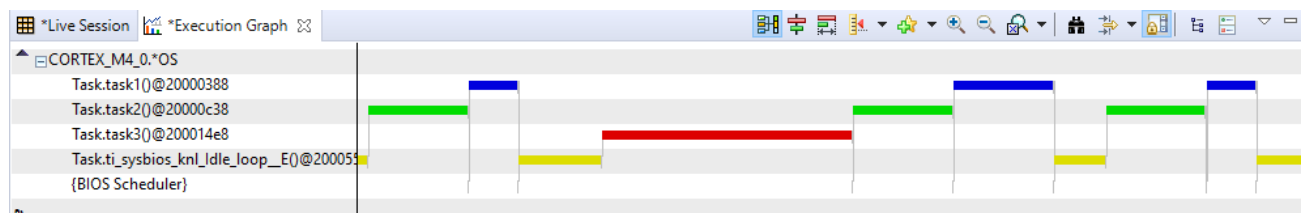
$$C3 + C2 \cdot [D3/P2] + C1 \cdot [D3/P1] \leq D3$$

$$100 + 40 \cdot [350/150] + 20 \cdot [350/100] = 300 \leq 350$$

Monitorizamos la ejecución del programa con la herramienta *Execution Graph* para observar que **se cumplen los plazos**

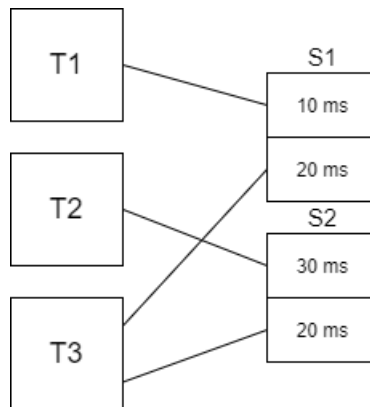


Si invertimos las prioridades asignando a la tarea T1 prioridad 1 y a la tarea T3 prioridad 3, podemos observar que T3 se ejecuta con mayor prioridad incumpliendo el plazo de T1.



Apartado 3

El programa concurrente de este apartado consta de tres tareas periódicas T1, T2 y T3 activadas por interrupción que comunican entre sí mediante dos servidores. Los períodos respectivos son: 100 ms, 200 ms y 400 ms. El plazo de respuesta es igual al periodo.



Hay que medir el tiempo de cómputo de cada una de estas tareas y asignar prioridades, dando prioridad al más urgente (DM), para completar la siguiente tabla.

Además, calculamos el bloqueo máximo de cada tarea calculado con el protocolo de herencia de prioridad.

Tarea	C	P	D	Prio	Bhp
T1	10	100	100	5	20
T2	60	200	200	3	20
T3	30	400	400	1	0

Ahora, debemos comprobar que con los datos de la tabla se va a garantizar el cumplimiento de plazos.

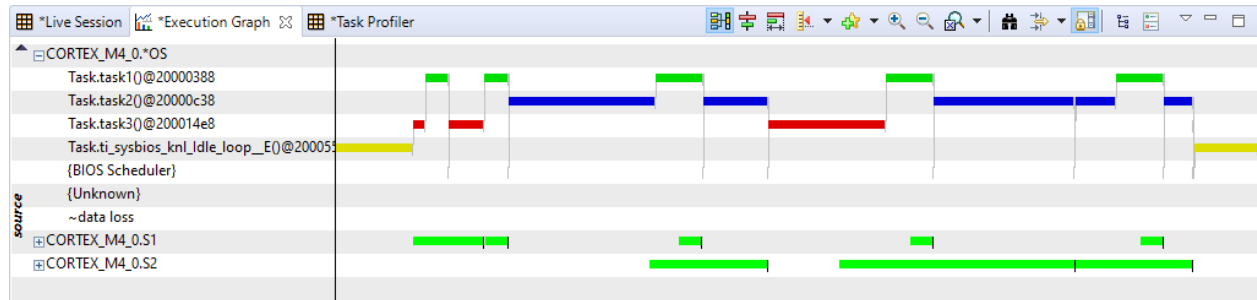
$$W1(D1) = C1 + B1 = 20 + 20 = \mathbf{40} \leq \mathbf{100} = D1$$

$$W2(D2) = [D2/P1]*C1 + C2 + B2 = 2*20 + 90 + 20 = \mathbf{150} \leq \mathbf{200} = D2$$

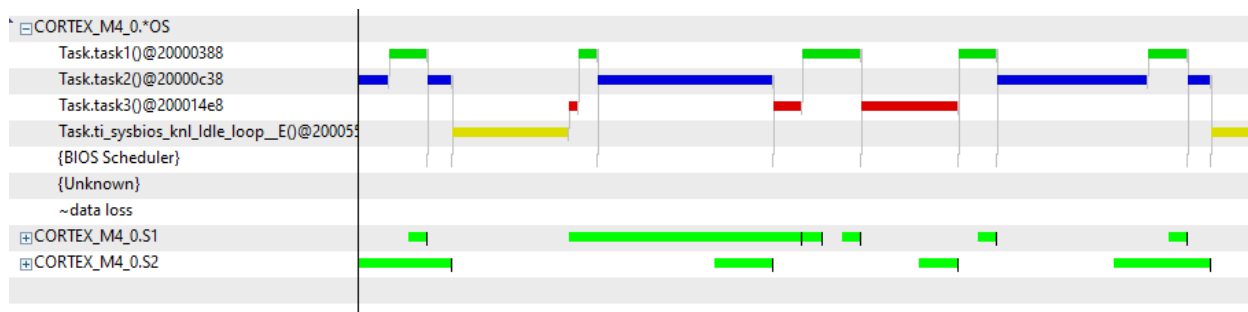
$$W3(D3) = [D3/P1]*C1 + [D3/P2]*C2 + C3 + B3 = 4*20 + 2*90 + 70 + 0 = \mathbf{330} \leq \mathbf{400} = D3$$

Al ejecutar el programa en el simulador observamos que se cumplen los plazos de respuesta.

También observamos que se realizan inversiones de prioridad. Al comenzar la traza, T3 está ejecutando el servicio S1, salta la tarea T1 con mayor prioridad y, cuando necesita ejecutar S1, se realiza la **inversion de prioridad** para que T3 termine de ejecutar el servicio S1 (15 ms) y posteriormente pueda utilizar el servicio la tarea T1.



Si cambiamos a unos mutex que no tienen protocolo de herencia de prioridad ocurre lo que muestra la siguiente traza. Observamos como T1 no puede ejecutar el servicio S1 y no puede haber una inversión de prioridad para que T3 lo libere cuanto antes. T1 **tiene que esperar a que T3 tenga prioridad.**



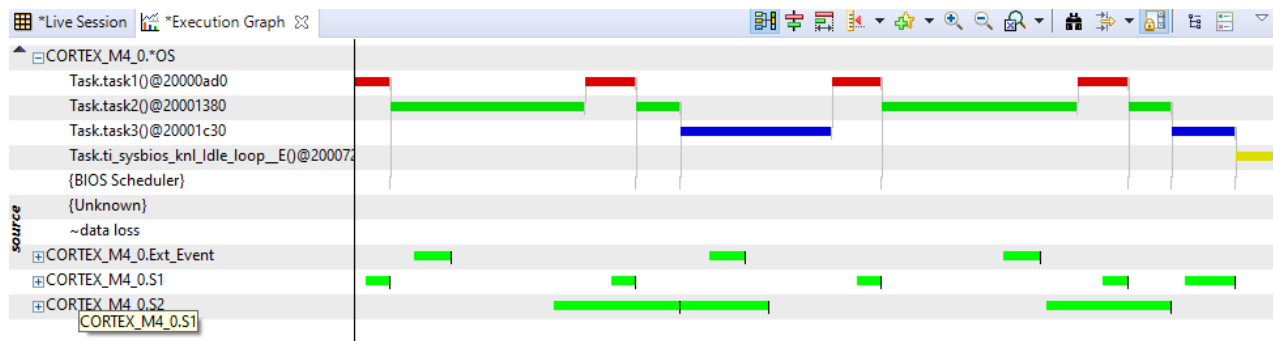
Práctica 2

En esta práctica incorporamos al escenario de tareas y servidores del apartado 3 de la práctica 1, una tarea periódica de 15 ms cada 120 ms.

Apartado 1

En este apartado el evento esporádico genera flanco de subida en el pin PD2. Uniremos los pines PD2 y PD3 para crear una rutina de interrupción que se active al detectar flanco de subida y que se encargue de ejecutar la tarea esporádica.

De esta manera estamos tratando esta tarea de forma directa con máxima prioridad, obteniendo un tiempo de respuesta muy bueno, pero expulsando el resto de las tareas.



$$W1(D1) = [D1/PE]*CE + C1 + B1 = 1*15 + 20 + 20 = \mathbf{55} \leq \mathbf{100} = D1$$

$$W2(D2) = [D2/PE]*CE + [D2/P1]*C1 + C2 + B2 = 2*15 + 2*20 + 90 + 20 = \mathbf{180} \leq \mathbf{200} = D2$$

$$W3(D3) = [D3/PE]*CE + [D3/P1]*C1 + [D3/P2]*C2 + C3 + B3 = 4*15 + 4*20 + 2*90 + 70 + 0 = \mathbf{390} \leq \mathbf{400} = D3$$

Comprobamos que se garantiza el cumplimiento de plazos ya que se cumplen todas las condiciones.

Apartado 2

En este apartado, cada vez que llegue la interrupción del evento esporádico, en vez de ejecutar con máxima prioridad la tarea, vamos a guardar el evento en un mailbox para ejecutar la tarea desde un servidor esporádico.

Tarea	C	P	D	Prio	Bhp
T1	10	100	100	5	20
TE	15	120	120	4	0
T2	60	200	200	3	20
T3	30	400	400	1	0

$$W1(D1) = C1 + B1 = 1*15 + 20 + 20 = \mathbf{40} \leq \mathbf{100} = D1$$

$$WE(DE) = [DE/P1]*C1 + CE + BE = 2*20 + 15 + 0 = \mathbf{55} \leq \mathbf{120} = DE$$

$$W2(D2) = [D2/PE]*CE + [D2/P1]*C1 + C2 + B2 = 2*15 + 2*20 + 90 + 20 = \mathbf{180} \leq \mathbf{200} = D2$$

$$W3(D3) = [D3/PE]*CE + [D3/P1]*C1 + [D3/P2]*C2 + C3 + B3 = 4*15 + 4*20 + 2*90 + 70 + 0 = \mathbf{390} \leq \mathbf{400} = D3$$

Tras asignar prioridades DM, comprobamos que se cumplen plazos. En la traza generada en el simulador también podemos verificar que se cumplen y que se realizan las pertinentes inversiones de prioridad.

