Interrupciones en FreeRTOS

Electrónica IV

Mg.Ing. Esteban Volentini (evolentini@herrera.unt.edu.ar)

https://facetvirtual.facet.unt.edu.ar/course/view.php?id=165

Interrupciones

- Son un mecanismo que permite suspender la tarea en curso y ejecutar una rutina especifica para atender una situación especial
- Son utilizadas para atender dispositivos sin la sobrecarga de tener que revisar periódicamente el estado del mismo
- El dispositivo procesa una trabajo y, cuando termina, genera un pedido de interrupción al procesador utilizando una línea de hardware especifica

Interrupciones

- El procesador identifica el pedido y decide aceptarlo en función de las prioridades de la tarea actual y del pedido de interrupción
- Si el procesador acepta la interrupción entonces ubica la rutina de servicio correspondiente, guarda el estado de la tarea actual y comienza a ejecutar la rutina de servicio con una prioridad igual a la de la interrupción.

Excepciones

- También pueden llamarse interrupciones por software o System Traps
- Tienen el mismo comportamiento que las interrupciones pero son causadas por eventos internos, normalmente condiciones de error
- Las excepciones son sincrónicas con el programa mientas que las interrupciones son son siempre sincrónicas

Excepciones

- Las excepciones también se utilizan para pedir servicios al sistema operativo en forma similar a la ejecución de una subrutina

- Normalmente se asigna un número y un. nombre a cada interrupción o excepción del sistema

Vector de interrupciones

- Es una tabla en memoria que contiene las direcciones de las rutinas de servicio de las interrupciones y excepciones.

- En el caso del Cortex-M4 las primeras 16 entradas de la tabla corresponden a excepciones y son las mismas para cualquier fabricante.

- El resto de las entradas son para interrupciones y depende del fabricante cuales se utilizan y a que dispositivos corresponden

Vector de interrupciones

- Existe un registro especial VTOR en el procesador que apunta a la dirección de inicio del vector de interrupciones
- Este registro normalmente esta cargado con el valor cero, pero puede cambiarse el valor del mismo y por cambiar totalmente el vector de interrupciones durante la ejecución

Interrupciones en FreeRTOS

- El vector de interrupciones es gestionado por el proyecto C como en BareMetal
- Para el Cortex-M4 se deben dirigir tres excepciones a rutinas del sistema operativo:
  - La interrupción de SysTick que es utilizada para gestionar las esperas y el round robin
  - La excepción SVC Handler que es utilizada para pedir servicios al sistema operativo
  - La excepción PendSV Handler que es utilizada para pedir servicios del sistema operativo en forma diferida

Interrupciones cortas

- Un principio de diseño ampliamente aceptado es que las rutinas de servicios a interrupciones deben mantenerse cortas y simples
- Por muy alta que sea la prioridad asignada a una tarea, la atención de las interrupciones suspende la ejecución de las tareas
- Por lo que las rutinas de servicio de las interrupciones modifican los tiempos de respuesta del sistema

Interrupciones cortas

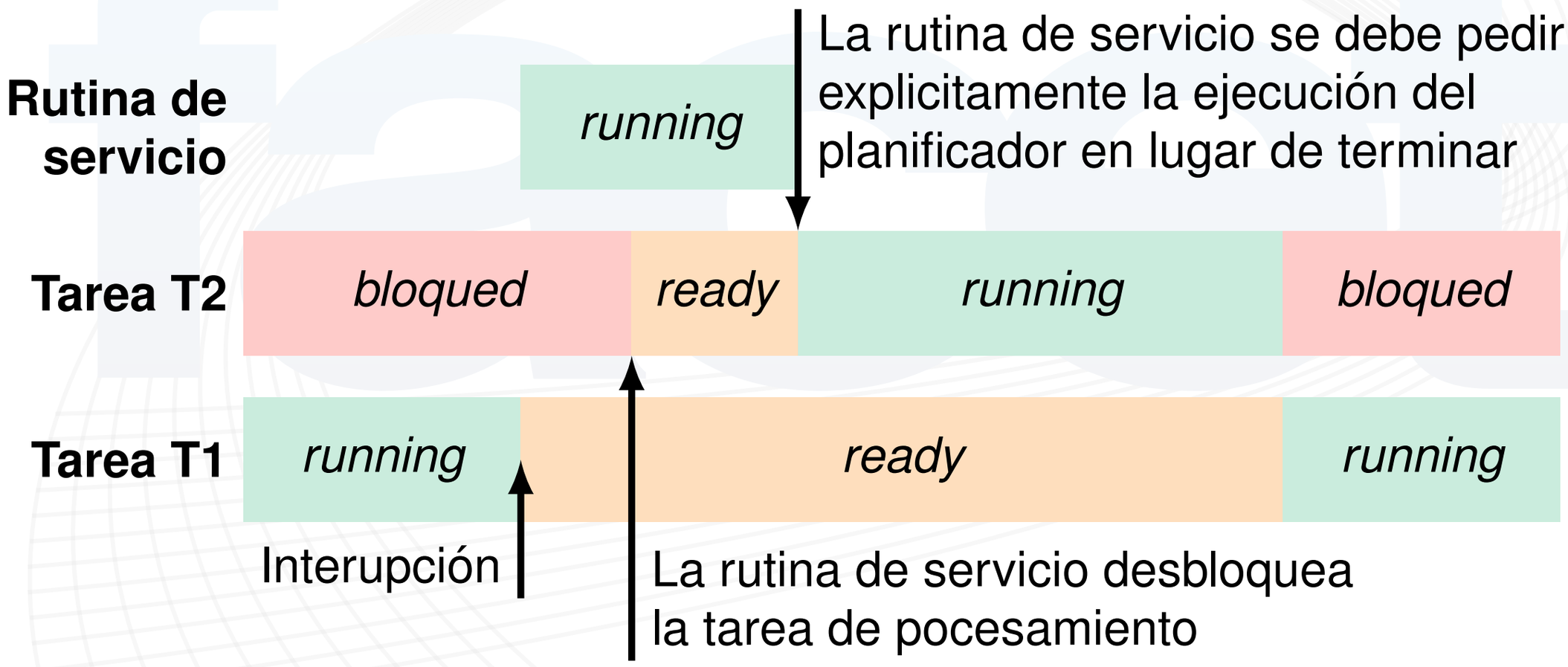
- Si ademas no se permite anidamiento de las interrupciones, entonces hay también una interferencia en los tiempos de repuestas con otras interrupciones
- Si por el contrario se permite el anidamiento de interrupciones, entonces aumenta la complejidad para predecir el comportamiento del sistema

Procesamiento diferido

- Es una técnica muy utilizada para mantener las rutinas de servicio cortas:
- La rutina registra la causa de la interrupción y la borra
- La rutina desbloquea a una tarea auxiliar que será la responsable de procesar la información.
- Esta tarea adicionar se vuelve a bloquear esperando la señal de una nueva interrupción.

Procesamiento diferido

- Si la tarea que procesa la información es la de mayor prioridad entonces se continua inmediatamente con el procesamiento de los datos



Implementación con semáforos

- La tarea que procesa la información pide un semáforo binario que se inicia vacío y se bloquea
- La rutina de servicio recibe los datos y otorga el semáforo lo que desbloquea a la tarea de procesamiento
- La tarea procesa la información recibida por la rutina de servicio y repite el lazo infinito por lo que pide nuevamente el semáforo y vuelve a bloquearse

Implementación con semáforos

- Si en el transcurso del procesamiento hubiese habido una nueva interrupción entonces se repite el procesamiento en forma inmediata
- Si ocurriese un tercer evento antes de terminar el primero entonces se pierde
- Utilizar semáforos contadores revuelve dl problema de perder eventos

Implementación con colas

- Las colas son una forma muy cómoda de comunicación entre las rutinas de servicio y las tareas de procesamiento
- Cuando se define una cola se especifica la cantidad de elementos a ingresar y el tamaño de los mismos, y se realiza la asignación de memoria correspondiente
  - Cuando se ingresan o se retiran elementos de la cola los mismos se copian de las variables a la memoria asignada a la cola y viceversa
  - Por esta razón no son método eficiente si los datos llegan con mucha frecuencia o son muy voluminosos