

IRQ y secciones críticas

Mg. Ing. Gonzalo E. Sanchez
Esp. Ing. Hanes N. Sciarrone
MSE - 2023

IRQ y secciones críticas

- Manejo de IRQ
- Secciones críticas
- Recomendaciones

Manejo de IRQ

Manejo de IRQ

- Como es sabido, un OS tiene la tarea primordial de gestionar el hardware donde corre.
- El primer recurso que gestiona es el CPU, mediante la asignación del mismo a distintas tareas.
- Esto da la posibilidad de multitasking.
- Hasta el momento no se ha gestionado otro hardware.

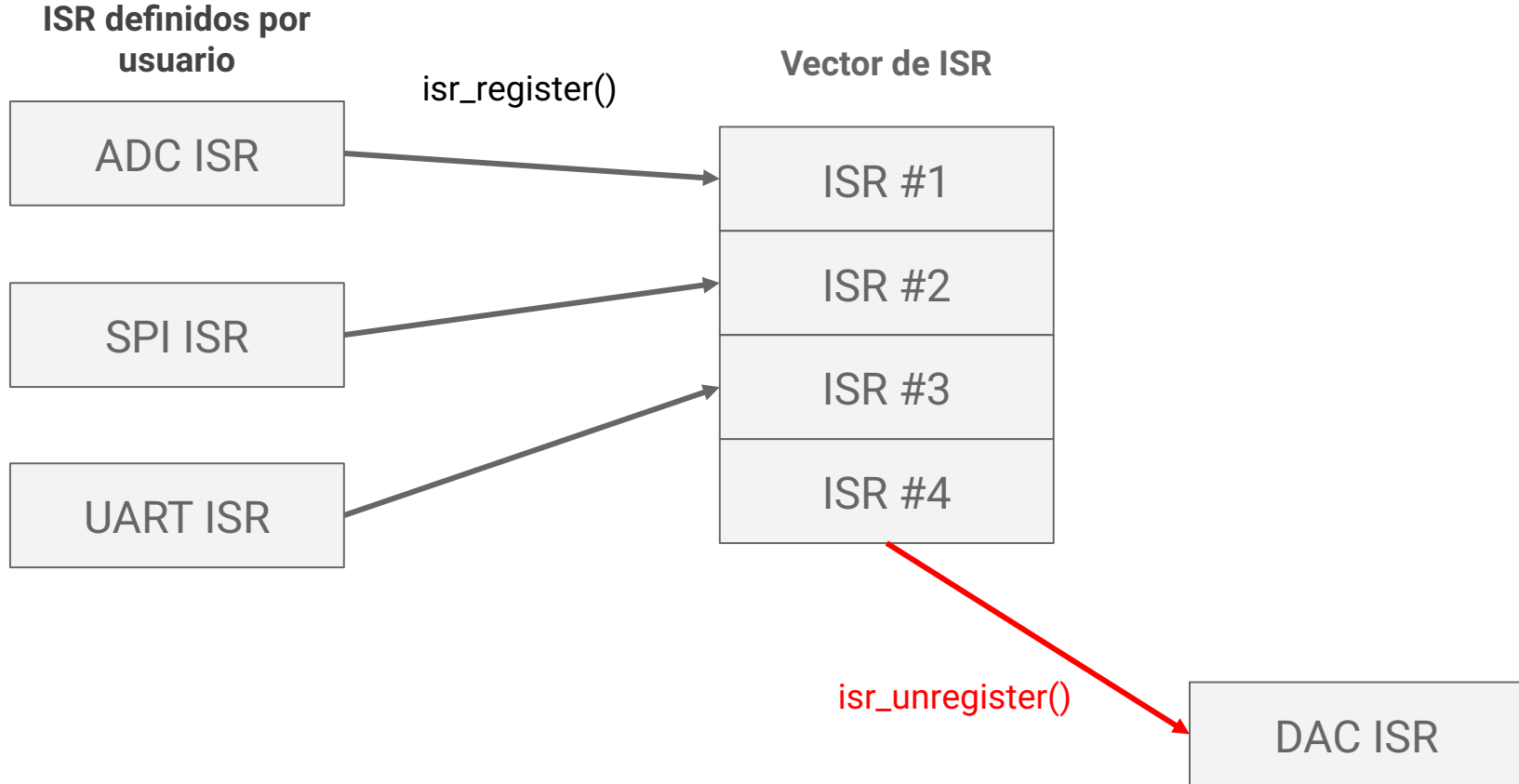
Manejo de IRQ

- Una de las principales características de un microcontrolador es la integración de hardware alrededor del procesador.
- Según el fabricante, la mayoría del hardware tiene asociado un vector de interrupción.
- En este momento, una interrupción no está contemplada por el OS, salvo las excepciones de SysTick y PendSV.
- No es buena práctica dejar librado al usuario el manejo de hardware.

Manejo de IRQ

- Es el OS que decide quién usa qué hardware, y en qué momento.
- Se debe proveer una función de API para registrar ISR que sean ejecutadas al momento de efectuarse una IRQ.
- De esta forma el vector de interrupciones es “armado” por el OS.

Manejo de IRQ



Manejo de IRQ

- Las interrupciones normalmente son eventos esperados por alguna tarea (datos recibidos, conversión ADC, etc).
- Se debe proveer un mecanismo para que el OS genere un scheduling luego de sucedido un evento.
- No todas las interrupciones contienen código que hace necesario un scheduling (eventos).

Secciones críticas

Secciones críticas

- El procesador Cortex M permite definir prioridades y grupos de prioridades para las excepciones.
- Cualquier IRQ será atendida antes o interrumpirá a PendSV (recordar que se setea como prioridad más baja).
- En este momento, solo SysTick está presente.
- Se resguarda de efectuar un cambio de contexto si el scheduler está en ejecución.

Secciones críticas

- Hay acciones que deben ser necesariamente atómicas.
- Dependerá de las necesidades del OS.
- Un ejemplo de esto es el cambio de contexto.
- Hay otras operaciones que implican la determinación de punteros accedidos por el scheduler que pueden cambiar.
- A veces el hardware impone limitaciones de accesos a buffers.

Secciones críticas

- Para todos estos casos es necesario definir una sección de código que no puede ser interrumpida.
- Recibe el nombre de sección crítica.
- Las secciones críticas deben ser lo más cortas posible.
- Dentro de una sección crítica el cambio de contexto está deshabilitado.

Secciones críticas

```
void tarea1(void) {  
    //Variables de la tarea  
    //Inicializacion de tarea  
  
    while(1) {  
        os_enterCritical();  
        /*  
        ***** seccion critica *****  
        */  
        os_exitCritical();  
    }  
}
```

Secciones críticas

- La opción más sencilla es deshabilitar todas las interrupciones.
- Hace imperativo que la sección crítica se mantenga lo más pequeña y eficiente posible.
- Otorga facilidad a la hora de diseñar el comportamiento del scheduling luego de una ISR.
- No es la opción más elegante o eficiente, pero se utiliza por simplicidad.

Secciones críticas

- Dentro del ISR de PendSV puede lograrse una sección crítica.
- Como la función `getContextoSiguiente()` es llamada por el ISR de PendSV, queda dentro de una sección crítica.
- Se utilizan las instrucciones `cpsid` y `cpsie` con el argumento `i`.
- Eso hace que se haga un clear o un set del registro PRIMASK.
- Recordar que esto afecta a todas las excepciones, excepto HARDFAULT, NMI y RESET.

Secciones críticas

PendSV_Handler:

```
cpsid i
```

```
/******
```

Todo el resto del handler

```
*****/
```

```
cpsie i
```

```
bx lr
```

```
//----- Fin de handler -----
```


Recomendaciones

Recomendaciones

- Para la implementación de IRQ, proveer al usuario dos funciones API para instalar y remover ISR.
- El vector de interrupciones contendrá una única dirección, que será un handler de IRQ del OS.
- Este handler invocará a la función cargada en un vector de ISR.
- El vector de ISR se poblará con direcciones provistas por la función API instalar ISR.

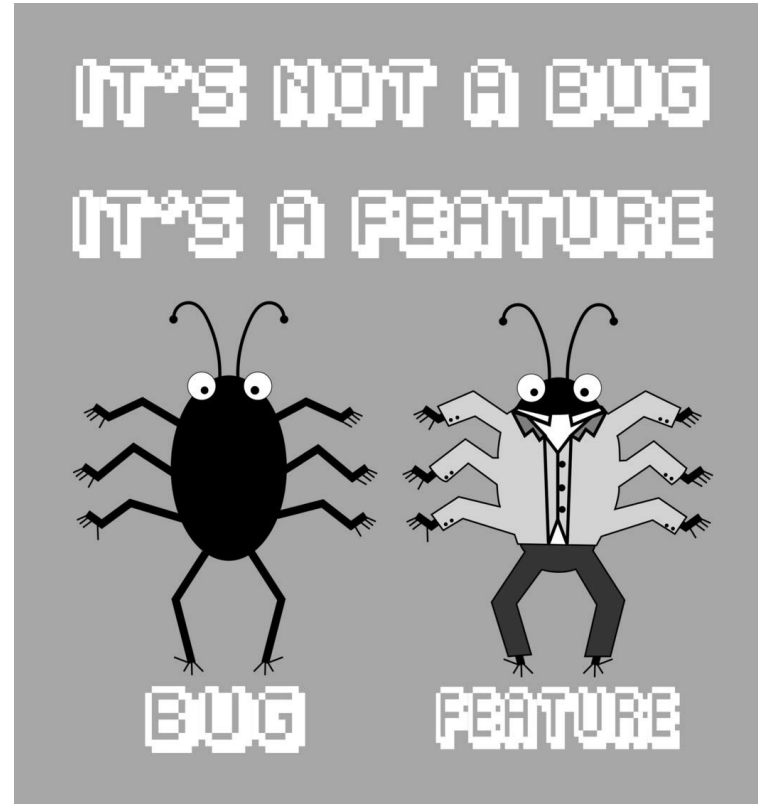
Recomendaciones

- Para la implementación de APIs del OS que esperan eventos (semáforos, colas) existen dos opciones:
 - Implementar funciones API específicas para el uso en IRQ (método freeRTOS).
 - Implementar un nuevo estado de OS y checkearlo dentro de las funciones API.
- Cualquiera de las dos aproximaciones es correcta. A criterio del estudiante cuál utilizar.

Recomendaciones

RECUERDE

**SI ESTÁ
DOCUMENTADO
NO ES UN BUG**



IRQ y secciones críticas

HANDS ON

1. Implementar Secciones críticas.
2. Implementar manejo de IRQ



Gracias.

