

Sistemas Empotrados - UGR - 2019/2020

Elena Cantero Molina

Tema 1

Tema 2

1. El compilador emitirá a la sección .data ...

- Todas las variables globales inicializadas a un valor distinto de cero
- Todas las variables inicializadas a un valor distinto de cero
- Todas las variables
- Todas las variables globales
- No contestar la pregunta

2. La dirección de carga de un símbolo ...

- Apunta a una memoria en la que se almacena dicho símbolo de forma permanente en el sistema
- Es la dirección que tiene dicho símbolo en tiempo de ejecución
- Es una dirección que siempre apunta a la RAM del sistema
- Siempre es diferente de su dirección de ejecución
- No contestar la pregunta

3. Según la arquitectura ARM, para llamar a la función $\text{pow}(x,y)$, en la que tanto sus parámetros de entrada (x e y) como su resultado son enteros de 32 bits, ¿Cómo se pasan los parámetros de entrada y cómo se recogen el resultado de la función?

- Los parámetros de entrada se pasan en los registros $r0$ y $r1$ y el resultado se recoge en $r0$
- Tanto los parámetros de entrada como el resultado se pasan por la pila
- Como el resultado se recibe en $r0$, los parámetros de entrada se pasan en $r1$ y $r2$
- Dado que los parámetros de entrada de una función pueden ser varios, se pasan por la pila. El resultado, al ser sólo un valor, se retorna en el registro $r0$
- No contestar la pregunta

4. En la arquitectura ARM se pueden ejecutar condicionalmente ...

- Todas las instrucciones del repertorio
- Sólo los saltos locales
- Sólo los saltos globales
- Sólo los saltos (tanto globales como locales)
- No contestar la pregunta

5. El repertorio de instrucciones thumb ...

- Permite que el código de un programa ocupe menos memoria RAM

- Permite ejecutar bytecodes de Java en un procesador ARM
- Mejora notablemente las prestaciones del código
- No tiene nada que ver con los procesadores ARM
- No contestar la pregunta

6. En un script de enlazado, las secciones de entrada ...

- Son las que componen los ficheros objeto que se enlazarán para formar el fichero ejecutable final
- Son las secciones que forman el ejecutable final
- Son las que definen las direcciones LMA de la imagen de la aplicación
- Son las que definen las direcciones
- VMA de la imagen de la aplicación
- No contestar la pregunta

7. Indica en cuál de las siguientes opciones podrían coincidir las direcciones VMA con las LMA

- Para las constantes
- Para las variables globales inicializadas
- Para las variables locales
- Para las variables globales sin inicializar
- No contestar la pregunta

8. ¿Cómo se puede modificar el valor de los flags de estado del procesador?

- Añadiendo el sufijo 's' a una instrucción
- Ejecutando cualquier instrucción. Los flags se fijarán en función del resultado
- Solamente mediante el uso de instrucciones de comparación
- Añadiendo a cualquier instrucción los sufijos de condición (lt, gt, eq, ...)
- No contestar la pregunta

9. Las instrucciones ldrsb y ldrsh cargan un entero de 8 y 16 bits respectivamente en la parte menos significativa de un registro de 32 bits. ¿Qué hacen dichas instrucciones con la parte más significativa del registro, no ocupada por el dato que se ha cargado?

- Se extiende el signo del dato
- Se rellenan dichos bits con el valor 0
- Se rellenan dichos bits con el valor 1
- Se dejan inalterados
- No contestar la pregunta

10. El ABI ...

- Define la interfaz a bajo nivel entre un programa y el sistema operativo u otro programa
- Es la agrupación de subrutinas, funciones y procedimientos que ofrece una biblioteca para ser empleado por otro software como capa de abstracción
- Proporciona interoperabilidad de ficheros de cabecera entre diferentes compiladores, plataformas, etc.
- Es una interfaz que proporciona los mismos servicios que el API, distinguiéndose en que el ABI los proporciona a nivel de código fuente

- No contestar la pregunta
-

Tema 3

1. Las variables locales ...

- Se crean en la pila cada vez que se invoca a la función en la que se han declarado
- Son almacenadas por el linker en la sección .data
- Son almacenadas por el linker en la sección .bss
- Se almacenan en el heap
- No contestar la pregunta

2. ¿Cuál de las siguientes opciones indica correctamente la ubicación del heap en memoria?

- Desde el final de la imagen de la aplicación hasta alcanzar el área de memoria dedicada a las pilas
- Desde el final de la pila hasta el final de la memoria RAM
- Al principio de la memoria RAM
- En la zona de memoria RAM que indique la aplicación
- No contestar la pregunta

3. El boot loader

- No puede escribirse completamente en C
- Debe escribirse completamente en ensamblador
- Puede escribirse completamente en C
- Se genera automáticamente por el compilador
- No contestar la pregunta

4. Las direcciones de inicio y fin del heap ...

- Se definen explícitamente en el linker script
- Se definen en el código fuente del cargador del BSP
- Se generan automáticamente en el proceso de enlazado
- Se definen por la aplicación, en función de sus necesidades de memoria dinámica
- No contestar la pregunta

5. A la hora de implementar un sistema empujado, la función main ...

- No debe retornar jamás
- Retornará un cero si se ha ejecutado correctamente
- Retornará un valor distinto de cero si se ha ejecutado correctamente
- Sólo retornará en caso de error
- No contestar la pregunta

6. En el caso de que se copie el código de la aplicación en la RAM para su ejecución, la aplicación usará ...

- Sólo direcciones VMA
- Tanto las direcciones VMA como las LMA, de forma simultánea
- Sólo direcciones LMA

- Tanto direcciones VMA como LMA, pero nunca simultáneamente
- No contestar la pregunta

7. ¿Cuántas pilas necesitan los sistemas basados en procesadores ARM?

- Una para cada modo de ejecución privilegiado
- Solamente una pila
- Una para el modo de ejecución USER y otra para los modos privilegiados
- Una para el modo USER, otra para los modos de interrupción y otra para el resto de modos
- No contestar la pregunta

8. ¿Qué debe haber siempre en la dirección 0 de memoria en un sistema basado en un procesador ARM?

- Un salto al cargador del sistema
- La primera instrucción del código de la aplicación
- La primera instrucción del cargador del sistema
- La dirección del manejador de la excepción reset
- No contestar la pregunta

9. ¿Cual de los siguientes pasos ejecuta antes el boot loader?

- Inicializar la tabla de vectores de excepción
- Habilitar interrupciones
- Remapear la memoria
- Detectar el origen de la aplicación
- No contestar la pregunta

10. La inicialización de toda la pila del sistema a un patrón determinado (ej. 0xdeadbeef) es ...

- Útil sólo durante el desarrollo para determinar un tamaño adecuado para la pila
- Necesaria para que la pila funcione correctamente durante la ejecución de la aplicación
- Necesaria para acelerar el proceso de inicialización del sistema
- Totalmente innecesaria y no sirve para nada
- No contestar la pregunta

Tema 4

1. Conforme más dispositivos hay en el sistema, la latencia de las interrupciones ...

- Aumenta sólo si no hay controlador de interrupciones
- Siempre aumenta
- Siempre disminuye
- Aumenta sólo si hay controlador de interrupciones
- No contestar la pregunta

2. Indica la opción que, evitando que la UART pueda interrumpir a la CPU cuando haya recibido un nuevo dato, implique la mínima degradación de las prestaciones del sistema

- Configurando el dispositivo para deshabilitar la petición de interrupciones debidas a la recepción de nuevos datos

- Deshabilitando la fuente de interrupción asociada al dispositivo en el controlador de interrupciones
- Deshabilitando la aceptación de peticiones de interrupción en la CPU
- Apagando el dispositivo
- No contestar la pregunta

3. La tabla de manejadores de excepción ...

- No es necesaria para poder implementar un sistema que soporte la gestión de excepciones
- Es necesaria para poder implementar un sistema que soporte la gestión de excepciones
- Contiene saltos a los manejadores de excepción
- Debe estar mapeada a la dirección 0x00000000
- No contestar la pregunta

4. En el diseño jerárquico del mecanismo de atención de interrupciones de un BSP, la implementación del manejador de nivel 0 depende de...

- La arquitectura del procesador del sistema
- El controlador de interrupciones del sistema
- El dispositivo que ha solicitado la interrupción
- La tabla de vectores de excepción
- No contestar la pregunta

5. Indica qué relación existe entre las prioridades de las excepciones Software Interrupt y Undefined Instruction

- Las dos tienen la misma prioridad
- Software Interrupt tiene más prioridad que Undefined Instruction
- Undefined Instruction tiene más prioridad que Software Interrupt
- En la arquitectura ARM no hay definidas prioridades para las excepciones
- No contestar la pregunta

6. Cuando se ejecute una aplicación, la existencia de una condición de carrera entre el código de la aplicación y el código del driver de un dispositivo ...

- No siempre provocará resultados incorrectos
- Siempre provocará resultados incorrectos
- Siempre provocará resultados correctos
- Es algo habitual en los sistemas gestionados mediante interrupciones
- No contestar la pregunta

7. Cuando se provoca una excepción, el procesador almacena en lr la dirección de retorno antes de pasar a ejecutar su manejador. Por lo tanto, a la hora de escribir un manejador de excepción, para poder retornar a la aplicación ...

- Es necesario ajustar el registro lr en función del tipo de excepción antes de retornar
- Sólo es necesario copiar el contenido del registro lr en el registro pc
- Siempre hay que restar 4 bytes al registro lr antes de retornar
- Se debe usar la instrucción ret, que se encargará de hacer el ajuste de lr y retornar
- No contestar la pregunta

8. Indica cuál de las siguientes opciones define correctamente un tipo de datos adecuado para poder gestionar en C la instalación de manejadores de excepción en el BSP

- `typedef void (* excep_handler_t) (void);`
- `typedef int (* excep_handler_t) (void);`
- `typedef int (* excep_handler_t) (int);`
- `typedef void (* excep_handler_t) (int);`
- No contestar la pregunta

9. Si la tabla de vectores de excepción está mapeada a la dirección 0x00000000 y la de manejadores de excepción a la dirección 0x00000020, indica cuál es el contenido correcto de la entrada 0 de vectores (excepción reset)

- `ldr pc, [pc, #0x18]`
- `ldr pc, [pc, #0x20]`
- `ldr pc, [pc, #0x24]`
- `ldr pc, [pc, #0x22]`
- No contestar la pregunta

10. Cuando llega a la CPU una petición de interrupción FIQ, ¿cómo se identifica la fuente?

- No es necesario identificar la fuente
- Mediante el manejador de nivel 0
- Mediante el manejador de nivel 1
- Mediante el controlador de interrupciones
- No contestar la pregunta

Práctica 1

1. ¿Qué orden usarías en OpenOCD para resetear y detener el procesador de la placa de prácticas?

- `soft_reset_halt`
- `reset`
- `halt`
- Ninguna. El procesador se detiene automáticamente al conectar con OpenOCD
- No contestar la pregunta

2. Asumiendo que los pines a los que están conectados los leds rojo y verde están configurados como de salida, ¿cómo apagarías los dos leds? Puedes consultar el esquemático de la placa y el manual de referencia del procesador

- `GPIO_DATA1 <- 0`
- `GPIO_DATA1 <- 0x00003000`
- `GPIO_DATA0 <- 0x00003000`
- `GPIO_DATA0 <- 0xffffcfff`
- No contestar la pregunta

3. ¿Qué orden de OpenOCD usarías para ejecutar un programa previamente cargado en la memoria de la placa?

- resume
- run
- continue
- Ninguno. El programa se ejecuta automáticamente tras cargarlo en la memoria
- No contestar la pregunta

4. Para conectarnos a la interfaz de ordenes de OpenOCD debemos ...

- Hacer un telnet a localhost:4444
- Hacer un telnet a localhost:3333
- Mandar la orden soft_reset_halt
- Hacerlo a través del depurador
- No contestar la pregunta

5. La orden de OpenOCD load_image ...

- Carga una imagen en formato binario en la memoria de la placa
- Carga una imagen en formato binario en la memoria de la placa y la ejecuta
- Carga un fichero ELF en la memoria de la placa
- Carga un fichero ELF en la memoria de la placa y lo ejecuta
- No contestar la pregunta

6. ¿Qué flag se usa en OpenOCD para indicarle los ficheros de configuración?

- -f file
- -c file
- Ninguno, simplemente se indica el fichero
- No hay que indicar ningún fichero. OpenOCD detecta la placa en tiempo de ejecución
- No contestar la pregunta

7. Indica a qué pin del GPIO está conectado el led rojo de la placa. Puedes consultar el esquemático de la placa

- GPIO44
- GPIO45
- GPIO52
- GPIO59
- No contestar la pregunta

8. OpenOCD...

- Proporciona una interfaz estándar entre el depurador y casi cualquier placa de desarrollo
- Es un depurador de código a bajo nivel
- Permite hacer un telnet a la placa de desarrollo
- Es un demonio de depuración que se ejecuta en la placa de desarrollo para permitir la depuración de programas
- No contestar la pregunta

9. ¿Cómo configurarías el GPIO para que se puedan encender y apagar los leds rojo y verde mediante software? Puedes consultar el esquemático de la placa y el manual de referencia del procesador

- GPIO_PAD_DIR_SET1 <- 0x00003000
- GPIO_PAD_DIR_RESET1 <- 0x00003000
- GPIO_PAD_DIR_SET0 <- 0x00003000
- GPIO_PAD_DIR_RESET0 <- 0x00003000
- No contestar la pregunta

10. La orden de OpenOCD mww 0x80000054 0x00002000

- Apaga el led verde
 - Enciende el led rojo
 - Fija el pin conectado al led rojo como de salida
 - Fija el pin conectado al led verde como de entrada
 - No contestar la pregunta
-

Práctica 2

1. Indica cual de las siguientes opciones se debe usar para definir la etiqueta loop

- loop: str r5, [r6]
- .global loop
- .label loop
- @loop:
- No contestar la pregunta

2. Indica cual es la opción correcta para retornar desde una función al programa principal

- mov pc, lr
- ret
- ldr pc, lr
- ldr pc, =lr
- No contestar la pregunta

3. Si en el registro r6 está la dirección del registro GPIO_DATA_SET1 y en r7 la dirección del registro GPIO_DATA_RESET1, ¿qué opción apaga los leds indicados por la máscara almacenada en r5?

- str r5, [r7]
- str r5, [r6]
- ldr r5, [r7]
- str r7, [r5]
- No contestar la pregunta

4. Para detectar una pulsación de un botón es necesario consultar el estado del registro ...

- GPIO_DATA0
- GPIO_DATA_SET0
- GPIO_DATA_RESET0
- GPIO_DATA1
- No contestar la pregunta

5. Selecciona la opción correcta para configurar pines a los que están conectados los leds de la placa

- Escribir unos en los bits 12 y 13 de GPIO_PAD_DIR1
- Escribir unos en los bits 12 y 13 de GPIO_PAD_DIR0
- Escribir unos en los bits 44 y 45 de GPIO_PAD_DIR1
- Escribir ceros en los bits 12 y 13 de GPIO_PAD_DIR1
- No contestar la pregunta

6. Selecciona la opción correcta para configurar pines a los que están conectados los botones de la placa

- Ninguna de las otras respuestas es correcta. La detección de las pulsaciones de los botones no depende solamente de la configuración del registro GPIO_PAD_DIR0
- Sólo es necesario modificar el registro GPIO_PAD_DIR0, escribiendo ceros en los bits que gestionan los botones
- Sólo es necesario modificar el registro GPIO_PAD_DIR0, escribiendo unos en los bits que gestionan los botones
- Como cada botón está conectado a dos pines, para cada botón escribiremos en GPIO_PAD_DIR0 un cero en uno de sus pines y un uno en el otro. La detección de las pulsaciones no depende de ningún otro registro
- No contestar la pregunta

7. En ensamblador, las directivas ...

- Son órdenes que se le dan a las herramientas de compilación
- Se ejecutan por el procesador en tiempo de ejecución
- Son un tipo especial de instrucción máquina
- No existen
- No contestar la pregunta

8. Suponiendo que la etiqueta pause apunta a la primera instrucción de una función, indica cual es la forma correcta de invocar a dicha función

- blgt pause
- mov pc, #pause
- bx label
- ble label
- No contestar la pregunta

9. Indica la opción correcta para definir una máscara para el led verde

- .set MASC_VERDE, (1 << 13)
- .set MASC_VERDE, 1 << 13
- set MASC_VERDE, (1 << 13)
- .set MASC_VERDE (1 << 13)
- No contestar la pregunta

10. La detección de pulsaciones de los botones, tal y como se ha desarrollado en la práctica ...

- No siempre detectará la pulsación de un botón
- Siempre detectará la pulsación de un botón
- Genera una interrupción cuando se pulsa un botón
- No permite detectar la pulsación de dos botones simultáneamente

- No contestar la pregunta
-

Práctica 3

1. El compilador emitirá a la sección .data ...

- Todas las variables globales inicializadas a un valor distinto de cero
- Todas las variables inicializadas a un valor distinto de cero
- Todas las variables
- Todas las variables globales
- No contestar la pregunta

2. La dirección de carga de un símbolo ...

- Apunta a una memoria en la que se almacena dicho símbolo de forma permanente en el sistema
- Es la dirección que tiene dicho símbolo en tiempo de ejecución
- Es una dirección que siempre apunta a la RAM del sistema
- Siempre es diferente de su dirección de ejecución
- No contestar la pregunta

3. Según la arquitectura ARM, para llamar a la función $\text{pow}(x,y)$, en la que tanto sus parámetros de entrada (x e y) como su resultado son enteros de 32 bits, ¿Cómo se pasan los parámetros de entrada y cómo se recogen el resultado de la función?

- Los parámetros de entrada se pasan en los registros $r0$ y $r1$ y el resultado se recoge en $r0$
- Tanto los parámetros de entrada como el resultado se pasan por la pila
- Como el resultado se recibe en $r0$, los parámetros de entrada se pasan en $r1$ y $r2$
- Dado que los parámetros de entrada de una función pueden ser varios, se pasan por la pila. El resultado, al ser sólo un valor, se retorna en el registro $r0$
- No contestar la pregunta

4. En la arquitectura ARM se pueden ejecutar condicionalmente ...

- Todas las instrucciones repertorio
- Sólo los saltos locales
- Sólo los saltos globales
- Sólo los saltos (tanto globales como locales)
- No contestar la pregunta

5. El repertorio de instrucciones thumb ...

- Permite que el código de un programa ocupe menos memoria RAM
- Permite ejecutar bytecodes de Java en un procesador ARM
- Mejora notablemente las prestaciones del código
- No tiene nada que ver con los procesadores ARM
- No contestar la pregunta

6. En un script de enlazado, las secciones de entrada ...

- Son las que componen los ficheros objeto que se enlazarán para formar el fichero ejecutable final
- Son las secciones que forman el ejecutable final
- Son las que definen las direcciones LMA de la imagen de la aplicación
- Son las que definen las direcciones VMA de la imagen de la aplicación
- No contestar la pregunta

7. Indica en cuál de las siguientes opciones podrían coincidir las direcciones VMA con las LMA

- Para las constantes
- Para las variables globales inicializadas
- Para las variables locales
- Para las variables globales sin inicializar
- No contestar la pregunta

8. ¿Cómo se puede modificar el valor de los flags de estado del procesador?

- Añadiendo el sufijo 's' a una instrucción
- Ejecutando cualquier instrucción. Los flags se fijarán en función del resultado
- Solamente mediante el uso de instrucciones de comparación
- Añadiendo a cualquier instrucción los sufijos de condición (lt, gt, eq, ...)
- No contestar la pregunta

9. Las instrucciones ldrsb y ldrsh cargan un entero de 8 y 16 bits respectivamente en la parte menos significativa de un registro de 32 bits. ¿Qué hacen dichas instrucciones con la parte más significativa del registro, no ocupada por el dato que se ha cargado?

- Se extiende el signo del dato
- Se rellenan dichos bits con el valor 0
- Se rellenan dichos bits con el valor 1
- Se dejan inalterados
- No contestar la pregunta

10. El ABI ...

- Define la interfaz a bajo nivel entre un programa y el sistema operativo u otro programa
- Es la agrupación de subrutinas, funciones y procedimientos que ofrece una biblioteca para ser empleado por otro software como capa de abstracción
- Proporciona interoperabilidad de ficheros de cabecera entre diferentes compiladores, plataformas, etc.
- Es una interfaz que proporciona los mismos servicios que el API, distinguiéndose en que el ABI los proporciona a nivel de código fuente
- No contestar la pregunta

Práctica 4

1. ¿Para que utilizamos la herramienta mc1322x-load?

- Cargar una imagen en la RAM de la placa usando el cargador de su ROM

- Borrar la memoria flash de la placa
 - Mandar órdenes a OpenOCD
 - Reseteear la placa
 - No contestar la pregunta
2. Indica cuál es el único símbolo que es necesario definir en el script de enlazado del BSP de la Econotag
- `_bss_end`
 - `_data_start`
 - `_bss_start`
 - `_text_start`
 - No contestar la pregunta
3. Según el manual del MC1322x, una vez que el cargador de la ROM ha copiado la imagen ejecutable de nuestra aplicación a la RAM, las 8 primeras palabras de la RAM deben contener una tabla de vectores de excepción. ¿Cómo se genera dicha tabla?
- Definiéndola al principio del fichero `hal/crt0.s` del BSP
 - Reservándole espacio al principio de la RAM en el script de enlazado
 - La genera el cargador de la ROM, justo antes de copiar la imagen a la RAM
 - La genera el cargador de la ROM, justo después de copiar la imagen a la RAM
 - No contestar la pregunta
4. Indica cuál de los siguientes componentes del BSP no es necesario para su construcción
- El script de enlazado
 - El cargador
 - Los drivers de los dispositivos
 - El fichero `Makefile`
 - No contestar la pregunta
5. Indica la dirección de memoria a la que estará mapeada la tabla de vectores de excepción que se define en nuestro BSP
- `0x00400000`
 - `0x00000000`
 - A cualquier dirección de memoria, siempre que se indique en el linker script
 - A la dirección de memoria que asigne el linker script a la sección `.text`
 - No contestar la pregunta
6. El resultado final de la construcción del BSP es ...
- Una biblioteca
 - Un conjunto de ficheros objeto
 - El firmware que se ejecutará en el sistema empotrado
 - Un fichero objeto
 - No contestar la pregunta
7. El fichero que contiene el cargador del BSP es
- `hal/crt0.s`

- hal/crt0.c
- hal/loader.s
- hal/boot.s
- No contestar la pregunta

8. El cargador de la ROM de la Econotag ...

- No inicializa el puntero de pila de ninguno de los modos de ejecución del procesador
- Inicializa todos los punteros de pila del procesador
- Sólo inicializa el puntero de pila del modo SVC, que es el modo en el que está la placa tras el reseteo
- Inicializa sólo los punteros de pila de los modos de ejecución privilegiados
- No contestar la pregunta

9. Indica cuál de las siguientes partes del cargador del BSP desarrollado en las prácticas se podría realizar en C

- La inicialización de los dispositivos de E/S
- La inicialización de las pilas
- La habilitación de interrupciones
- La carga de la imagen ejecutable en la RAM
- No contestar la pregunta

10. A la hora de diseñar el script de enlazado del BSP para la Econotag, para aquellas secciones que se copiarán de la Flash a la RAM ...

- Solamente es necesario indicar las direcciones VMA
- Solamente es necesario indicar las direcciones LMA
- Es necesario indicar tanto las direcciones VMA como las LMA
- No es necesario indicar nada, ya que serán copiadas por el cargador de la placa
- No contestar la pregunta

Práctica 5

1. ¿En qué dirección de memoria RAM estará la tabla de manejadores una vez que se haya cargado la imagen ejecutable en la placa?

- 0x00400020
- 0x00400000
- 0x00000020
- 0x00000000
- No contestar la pregunta

2. ¿Qué contiene cada una de las entradas de la tabla de manejadores de excepción?

- Un puntero a una función
- Una instrucción ejecutable
- El argumento que se pasará al manejador
- Un puntero a una lista de argumentos para el manejador

- No contestar la pregunta
3. Si insertamos una instrucción no definida en nuestro programa, pero se nos olvida instalar un manejador para que gestione la excepción cuando se trate de ejecutar dicha excepción ...
- Se usará el manejador por defecto definido en el BSP
 - El procesador no atenderá la excepción al no tener definido un manejador que la gestione
 - Provocará un resultado totalmente aleatorio
 - Se reseteará el procesador
 - No contestar la pregunta
4. Indica qué opción es la correcta para insertar una instrucción no definida en un programa escrito en C
- `asm(".word 0x26889912\n");`
 - `.word 0x26889912`
 - `asm(".data .word 0x26889912\n");`
 - `asm(".data 0x26889912\n");`
 - No contestar la pregunta
5. Indica el fichero del BSP en el que está definida la tabla de manejadores
- en `hal/crt0.s`
 - en `econotag.ld`
 - en `hal/excep.c`
 - en `include/excep.h`
 - No contestar la pregunta
6. La función `excep_disable_ints ()` ...
- Retorna el valor que tenían los bits I y F antes de fijarlos a 1
 - Retorna el valor que tenían los bits I y F antes de fijarlos a 0
 - Fija los bits I y F a 1 y no retorna nada
 - Fija los bits I y F a 0 y no retorna nada
 - No contestar la pregunta
7. Los manejadores de excepción ...
- Forman parte del código de la aplicación
 - Se deben declarar en el fichero `excep.c` del BSP
 - Forman parte del BSP, pero deben declararse en un fichero aparte
 - No se pueden cambiar por la aplicación
 - No contestar la pregunta
8. Indica el fichero del BSP en el que están definidos todos los tipos de excepción de la arquitectura ARM
- en `include/excep.h`
 - en `hal/excep.c`
 - en `hal/crt0.s`
 - en `include/system.h`
 - No contestar la pregunta

9. Indica qué opción es la que debe usar el desarrollador de aplicaciones para instalar la función `my_handler` como un manejador personalizado para las excepciones de tipo `Undefined Exception`

- `excep_set_handler (excep_undef, my_handler);`
- `excep_set_handler (my_handler);`
- `_excep_handlers[excep_undef] = my_handler;`
- `attribute ((interrupt ("UNDEF"))) void my_handler (void)`
- No contestar la pregunta

10. Indica cual de las siguientes opciones declara correctamente en C un manejador para una excepción de tipo `Undefined Instruction`

- `attribute ((interrupt ("UNDEF"))) void my_handler (void)`
- `void my_handler (void)`
- `attribute ((exception ("UNDEF"))) void my_handler (void)`
- `attribute (exception) void my_handler (void)`
- No contestar la pregunta

Práctica 6

1. Selecciona la opción correcta para que solamente las peticiones de interrupción del temporizador se manden a la CPU como interrupciones FIQ ...

- `itc_regs->INTTYPE = (1 << itc_src_tmr);`
- `itc_regs->INTTYPE |= (1 << itc_src_tmr);`
- `itc_regs->FIQNUM = itc_src_tmr;`
- `itc_regs->INTTYPE = itc_src_tmr;`
- No contestar la pregunta

2. Indica a cuál de las siguientes funciones es necesario añadirle el atributo `attribute ((interrupt ("IRQ")))`

- `void excep_nonnested_irq_handler (),` definida en `hal/excep.c`
- `void itc_service_normal_interrupt (),` definida en `drivers/itc.c`
- La ISR del dispositivo que ha provocado la interrupción, definida por el usuario en su aplicación
- El vector de excepción del modo IRQ
- No contestar la pregunta

3. Indica cual de las siguientes opciones deshabilita todas las fuentes de interrupción de la Econotag

- `itc_regs->INTENABLE = 0`
- `itc_regs->INTENNUM = 0`
- `itc_regs->INTDISNUM = 0`
- `itc_regs->INTFRC = 0`
- No contestar la pregunta

4. La función `itc_restore_ints...`

- Restaura un valor anterior del registro `itc_regs->INTENABLE`
- Activa todas las fuentes de interrupción del sistema
- Restaura un valor anterior del registro `itc_regs->INTENNUM`

- Copia el contenido del registro `itc_regs->INTDISNUM` en el registro `itc_regs->INTENNUM`
 - No contestar la pregunta
5. Indica la opción correcta para simular una interrupción del conversor analógico-digital sin afectar al resto de fuentes de interrupción
- `itc_regs->INTFRC |= (1 << itc_src_adc);`
 - `itc_regs->INTFRC = (1 << itc_src_adc);`
 - `itc_regs->INTFRC = itc_src_adc;`
 - `itc_regs->INTENABLE = itc_src_adc;`
 - No contestar la pregunta
6. Cada fuente de interrupción en el sistema tiene asociado un número, que indica a qué entrada del controlador de interrupciones está conectada dicha fuente. Indica para qué se usa este número en el controlador de interrupciones
- Para priorizar e identificar la fuente de la interrupción
 - Sólo para identificar la fuente de la interrupción
 - Sólo para priorizar las peticiones de interrupción
 - No se usa para nada. El controlador sondea los dispositivos para identificar la fuente de interrupción
 - No contestar la pregunta
7. Indica qué opción es la correcta para habilitar las peticiones de interrupción de la UART1 sin afectar al resto de fuentes de interrupción
- `itc_regs->INTENABLE |= (1 << itc_src_uart1)`
 - `itc_regs->INTENABLE = (1 << itc_src_uart)`
 - `itc_regs->INTENABLE = itc_src_uart`
 - `itc_regs->INTENABLE |= itc_src_uart`
 - No contestar la pregunta
8. Para comprobar el correcto funcionamiento del driver del controlador de interrupciones desarrollado en la práctica se ha desarrollado un programa de test que fuerza una interrupción de la fuente ASM. ¿Qué hace la ISR de test que hemos desarrollado en dicho programa?
- Dejar de forzar la interrupción y encender el led verde
 - Encender el led verde
 - Dejar de forzar la interrupción y retornar
 - Parpadear un led
 - No contestar la pregunta
9. Asumiendo que en el driver del controlador de interrupciones existe una tabla llamada `itc_handlers` que almacena punteros a las ISR de cada uno de los dispositivos del sistema, y que cuando se produce una interrupción el controlador de interrupciones almacenará en el registro `NIVECTOR` el identificador de la fuente más prioritaria que ha solicitado la interrupción, indica cual de las siguientes opciones dará servicio a la interrupción
- `itc_handlers[itc_regs->NIVECTOR];`
 - `itc_handlers[itc_regs->NIVECTOR]`

- `itc_handlers(itc_regs->NIVECTOR)`
- `itc_handlers->itc_regs->NIVECTOR`
- No contestar la pregunta

10. En el mapa de memoria de los registros del controlador de memoria del MC1322x se indica que las direcciones `ITC_BASE + 0x18` a `ITC_BASE + 0x24` están reservadas. Indica la forma correcta de generar este espacio en la estructura de datos `itc_regs_t`

- `uint32_t reserved[4];`
- `uint32_t reserved[3];`
- `uint32_t reserved[2];`
- `uint32_t reserved;`
- No contestar la pregunta

Práctica 7

1. ¿Qué bits de qué registro hay que alterar para fijar la función del pin 44 a la función `gpio_func_alternate_1`?

- Los bits 24 y 25 del registro `GPIO_FUNC_SEL2`
- Los bits 12 y 13 del registro `GPIO_FUNC_SEL2`
- El bit 44 del registro `GPIO_FUNC_SEL0`
- El bit 12 de `GPIO_FUNC_SEL1`
- No contestar la pregunta

2. Tras llamar a la función `gpio_get_port (gpio_port_0, &port0_data)`, indique cual de las siguientes opciones permite detectar la pulsación del botón S3

- `if (port0_data & (1 << gpio_pin_26))...`
- `if (port0_data && (1 << gpio_pin_26)) ...`
- `if (port0_data & gpio_pin_26) ...`
- `if (port0_data && gpio_pin_26) ...`
- No contestar la pregunta

3. Indica cual de las siguientes opciones implementa correctamente la función `gpio_set_pin (gpio_pin_t pin)`

- `gpio_regs->DATA[pin >> 5] |= 1 << (pin & 0x1f)`
- `gpio_regs->DATA[pin >> 5] = pin & 0x1f`
- `gpio_regs->DATA[pin >> 5] = 1 << (pin & 0x1f)`
- `gpio_regs->DATA[pin >> 5] &= ~(1 << (pin & 0x1f))`
- No contestar la pregunta

4. Teniendo en cuenta el mapa de memoria de los registros del GPIO del Freescale 1322x, ¿qué estructura de datos es la más adecuada para gestionar dichos registros en C?

- Estructura de arrays
- Estructura de campos sencillos
- Array de punteros a estructuras

- Otra
- No contestar la pregunta

5. ¿Cuántas funciones diferentes se pueden asignar a un pin de E/S de la Econotag?

- 4
- 1
- 2
- 3
- No contestar la pregunta

6. Indica el mínimo número de accesos a los registros del GPIO de la Econotag para configurar la dirección de todos los pines de E/S de procesador

- 2
- 1
- 32
- 64
- No contestar la pregunta

7. Indica cual de los siguientes modos de funcionamiento se usa para configurar un pin de E/S para hacer E/S de propósito general

- Normal mode (0)
- Alternate mode 1 (1)
- Alternate mode 2 (2)
- Alternate mode 3 (3)
- No contestar la pregunta

8. La función `gpio_set_pin_func(gpio_pin_t pin, gpio_func_t func)`, debe modificar los dos bits que gestionan la función del pin para que almacenen el valor `func`. Asumiendo que en dicha función se ha obtenido el índice del registro que se debe modificar en la variable local `reg` y el desplazamiento de los bits asociados a pin en dicho registro en la variable local `offset`, indica la opción correcta para realizar el proceso

- `gpio_regs->FUNC_SEL[reg] |= (3 << offset);`
- `gpio_regs->FUNC_SEL[reg] &= (func << offset);`
- `gpio_regs->FUNC_SEL[reg] &= (0 << offset);`
- `gpio_regs->FUNC_SEL[reg] |= (func << offset);`
- No contestar la pregunta

9. Indica cual de las siguientes opciones tiene exactamente el mismo resultado que la instrucción `gpio_regs->PAD_DIR_RESET[port] = mask`

- `gpio_regs->PAD_DIR[port] &= ~mask`
- `gpio_regs->PAD_DIR[port] &= mask`
- `gpio_regs->PAD_DIR[port] |= mask`
- `gpio_regs->PAD_DIR[port] = ~mask`
- No contestar la pregunta

10. Indica cual de las siguientes opciones genera una máscara correcta para los pines 22 y 43 del GPIO

- No se puede generar la máscara propuesta
- $(1 \ll \text{gpio_pin_22}) \mid (1 \ll \text{gpio_pin_43})$
- $(1 \ll \text{gpio_pin_22}) \mid (1 \ll \text{gpio_pin_43} - \text{gpio_pin_32})$
- $\text{gpio_pin_22} \mid \text{gpio_pin_43}$
- No contestar la pregunta

Práctica 8

Práctica 9

Práctica 10