

Universidad Politécnica de la Zona Metropolitana de Guadalajara

Ing. Mecatrónica

Programación de Sistemas Embebidos

Maestro: Carlos Enrique Moran Garabito

Alumno: Flores Macias Cesar Fabian

3\_1\_Interrupciones

**Interrupciones**

En el contexto de la [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica), una interrupción (del inglés *interrupt request*, en español «petición de interrupción») es una señal recibida por el [procesador](https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento) de una [computadora](https://es.wikipedia.org/wiki/Computadora), para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

Una interrupción es una suspensión temporal de la ejecución de un [proceso](https://es.wikipedia.org/wiki/Proceso_(inform%C3%A1tica)), para pasar a ejecutar una subrutina de servicio de interrupción, la cual, por lo general, no forma parte del programa, sino que pertenece al [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) o al [BIOS](https://es.wikipedia.org/wiki/BIOS). Una vez finalizada dicha subrutina, se reanuda la ejecución del programa.

Las interrupciones son generadas por los [dispositivos periféricos](https://es.wikipedia.org/wiki/Perif%C3%A9rico_(inform%C3%A1tica)) habilitando una señal del CPU (llamada IRQ del inglés "interrupt request") para solicitar atención del mismo. Por ejemplo. cuando un disco duro completa una lectura solicita atención al igual que cada vez que se presiona una tecla o se mueve el ratón.

La primera técnica que se empleó para esto fue el [*polling*](https://es.wikipedia.org/wiki/Polling), que consistía en que el propio procesador se encargara de sondear los dispositivos periféricos cada cierto tiempo para averiguar si tenía pendiente alguna comunicación para él. Este método presentaba el inconveniente de ser muy ineficiente, ya que el procesador consumía constantemente tiempo y recursos en realizar estas instrucciones de sondeo.

El mecanismo de interrupciones fue la solución que permitió al procesador desentenderse de esta problemática, y delegar en el dispositivo periférico la responsabilidad de comunicarse con él cuando lo necesitara. El procesador, en este caso, no sondea a ningún dispositivo, sino que queda a la espera de que estos le avisen (le "interrumpan") cuando tengan algo que comunicarle (ya sea un evento, una transferencia de información, una condición de error, etc.).

**Funcionamiento del mecanismo de interrupciones**

Todos los dispositivos que deseen comunicarse con el procesador por medio de interrupciones deben tener asignada una línea única capaz de avisar al CPU cuando le requiere para realizar una operación. Esta línea se denomina *IRQ*.

Las IRQ son líneas que llegan al [controlador de interrupciones](https://es.wikipedia.org/wiki/Controlador_de_interrupciones), un componente de [hardware](https://es.wikipedia.org/wiki/Hardware) dedicado a la gestión de las interrupciones, y que puede estar integrado en el procesador principal o ser un circuito separado conectado al mismo. El controlador de interrupciones debe ser capaz de habilitar o inhibir las líneas de interrupción y establecer prioridades entre las mismas. Cuando varias líneas de petición de interrupción se activan a la vez, el controlador de interrupciones utilizará estas prioridades para escoger la interrupción sobre la que informará al procesador principal. También puede darse el caso de que una rutina de tratamiento de interrupción sea interrumpida para realizar otra rutina de tratamiento de una interrupción de mayor prioridad a la que se estaba ejecutando; aunque hay interrupciones que no se pueden deshabilitar (conocidas como [interrupciones no enmascarables](https://es.wikipedia.org/wiki/Interrupci%C3%B3n_no_enmascarable) o NMI).

Un procesador principal que no tenga un controlador de interrupciones integrado, suele tener una única línea de interrupción llamada habitualmente INT. Esta línea es activada por el controlador de interrupciones cuando tiene una interrupción que servir. Al activarse esta línea, el procesador consulta los [registros](https://es.wikipedia.org/wiki/Registro_(hardware)) del controlador de interrupciones para averiguar cuál IRQ hay que atender. A partir del número del IRQ busca en la tabla de [vectores de interrupción](https://es.wikipedia.org/wiki/Vector_de_interrupciones) la dirección de la [rutina](https://es.wikipedia.org/w/index.php?title=Tutina_(inform%C3%A1tica)&action=edit&redlink=1) a la que debe llamar para atender la petición del dispositivo asociado a dicha IRQ.

**Procesamiento de una interrupción**

* Terminar la ejecución de la [instrucción máquina](https://es.wikipedia.org/wiki/Lenguaje_de_m%C3%A1quina) en curso.
* Salvar el estado del procesador (valores de registros y flags) y el valor del [contador de programa](https://es.wikipedia.org/wiki/Contador_de_programa), IP, en la pila, de manera que en la CPU, al terminar el proceso de interrupción, pueda seguir ejecutando el programa a partir de la última instrucción.
* La CPU salta a la dirección donde está almacenada la rutina de servicio de interrupción (Interrupt Service Routine, o abreviado ISR) y ejecuta esa rutina que tiene como objetivo atender al dispositivo que generó la interrupción.
* Una vez que la rutina de la interrupción termina, el procesador restaura el estado que había guardado en la pila en el paso 2 y retorna al programa que se estaba usando anteriormente.

**Tipos de interrupciones**

* Interrupciones de hardware. Estas son asíncronas a la ejecución del procesador, es decir, se pueden producir en cualquier momento independientemente de lo que esté haciendo el CPU en ese momento. Las causas que las producen son externas al procesador y a menudo suelen estar ligadas con los distintos dispositivos de entrada o salida.
* Excepciones. Son aquellas que se producen de forma síncrona a la ejecución del procesador y por tanto podrían predecirse si se analiza con detenimiento la traza del programa que en ese momento estaba siendo ejecutado en la CPU. Normalmente son causadas al realizarse operaciones no permitidas tales como la división entre 0, el desbordamiento, el acceso a una posición de memoria no permitida, etc.
* Interrupciones por software. Las interrupciones por software son aquellas generadas por un programa en ejecución. Para generarlas, existen distintas instrucciones en el [código máquina](https://es.wikipedia.org/wiki/C%C3%B3digo_m%C3%A1quina) que permiten al programador producir una interrupción, las cuales suelen tener nemotécnicos tales como INT (por ejemplo, en [DOS](https://es.wikipedia.org/wiki/DOS) se realiza la instrucción INT 0x21 y en [Unix](https://es.wikipedia.org/wiki/Unix) se utiliza INT 0x80 para hacer [llamadas de sistema](https://es.wikipedia.org/wiki/Llamada_al_sistema)).

**Interrupciones de hardware**

Las interrupciones de hardware son aquellas interrupciones que se producen como resultado de, por lo general, una operación de E/S. No son producidas por ninguna instrucción de un programa sino por las señales que emiten los [dispositivos periféricos](https://es.wikipedia.org/wiki/Perif%C3%A9rico_(inform%C3%A1tica)) para indicarle al procesador que necesitan ser atendidos.

Cuando el microprocesador accede a un periférico (disco duro, puerto de comunicación...), puede transcurrir algún tiempo antes de que los datos sean obtenidos o transmitidos. La solución más simple es esperar hasta recibir los datos o hasta que se haya efectuado la transmisión (polling), pero esta solución bloquea todos los programas en ejecución, y eso no puede admitirse en un sistema [multitarea](https://es.wikipedia.org/wiki/Multitarea). Por ello, en los sistemas modernos se prefiere un funcionamiento mediante interrupciones, ya que éstas permiten mejorar la productividad del procesador, de forma que este último puede ordenar una operación de entrada o salida y, en lugar de tener que realizar una espera activa, se puede dedicar a atender a otro proceso o aplicación hasta que el dispositivo esté de nuevo disponible, siendo dicho dispositivo el encargado de notificar al procesador mediante la línea de interrupción que ya está preparado para continuar o terminar la operación de entrada o salida.

**Excepciones**

Las excepciones son un tipo de interrupción sincrónica típicamente causada por una condición de error en un programa, como por ejemplo una división entre 0 o un acceso inválido a memoria en un proceso de usuario. Normalmente genera un [cambio de contexto](https://es.wikipedia.org/wiki/Cambio_de_contexto) a [modo supervisor](https://es.wikipedia.org/wiki/Modo_supervisor) para que el sistema operativo atienda el error. Así pues, las excepciones son un mecanismo de protección que permite garantizar la integridad de los datos almacenados tanto en el espacio de usuario como en el espacio [kernel](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)). Cuando el Sistema Operativo detecta una excepción intenta solucionarla, pero en caso de no poder simplemente notificará la condición de error a la aplicación/usuario y abortará la misma.

**Interrupciones por software**

Las interrupciones por software, también denominadas [llamadas al sistema](https://es.wikipedia.org/wiki/Llamada_al_sistema), son aquellas generadas por un programa mientras este está ejecutándose. En general, actúan de la siguiente manera: Un programa en ejecución llega a una instrucción que requiere del sistema operativo para alguna tarea, por ejemplo, para leer un archivo en el disco duro (cuando un programa necesita un dato exterior, se detiene y pasa a cumplir con las tareas de recoger ese dato). En ese momento por tanto llama al sistema y se interrumpe virtualmente hasta recibir respuesta, en el ejemplo anterior hasta que no se haya leído el disco y el archivo esté en memoria principal. Durante esa espera las instrucciones que se ejecutarán no serán del programa, sino del sistema operativo. Una vez éste termine su rutina ordenará reanudar la ejecución del programa autointerrumpido en espera. Por último, la ejecución del programa se reanuda.

Bibliografía: <http://logica-reptilia.blogspot.com/2009/03/interrupciones.html>

<https://es.scribd.com/presentation/283575359/Programacion-de-Interrupciones-en-c>

<http://www.fdi.ucm.es/profesor/jjruz/WEB2/Temas/Curso05_06/EC9.pdf>