¿Qué es una interrupción en el contexto de los microprocesadores?

Una interrupción es una suspensión temporal en la ejecución de un programa, para pasar a ejecutar una subrutina en la que se corre un código específico, que generalmente pertenece al sistema operativo o al BIOS, se trata la situación y luego de concluir este subproceso se continúa con la secuencia normal del programa. Las interrupciones permiten a los microprocesadores atender eventos asíncronos.

- ¿Se puede hablar de la historia de las interrupciones?

En los sistemas de comunicación primitivos cuando una aplicación requería que una tecla fuese pulsada, se interrogaba de manera reiterada el teclado esperando que esta tecla fuese presionada, y mientras se esperaba esta tecla no era posible realizar otras tareas, no existía lo que actualmente es conocido como los sistemas multitarea. El proceso anterior es comúnmente conocido como polling o sondeo, que hace alusión a una acción de consulta reiterada, regularmente hacia un dispositivo de hardware para crear una tarea sincrónica, esto también podría implementarse en dispositivos de software. Esta técnica fue reemplazada por las interrupciones. El polling presentaba el problema de ser bastante ineficiente, el procesador gastaba mucho tiempo en las actividades y consumía muchos recursos para ejecutar estas órdenes.

Las interrupciones lograron dar solución a estos inconvenientes y despreocuparse de esta problemática, y el artefacto periférico solo se comunica con el procesador cuando se requiere. El procesador no sondea los aparatos, él solo espera a que le deban comunicar algo, sin la necesidad de consultar de manera constante, solo cuando le interrumpan resuelve el asunto.

- ¿Qué tipo de interrupciones existen?

Según la fuente que crea las interrupciones se pueden catalogar las interrupciones de la siguiente manera:

-Interrupciones por hardware: es una señal eléctrica elaborada por un dispositivo periférico. Con esto se indica al procesador que esta herramienta física requiere ser atendida. El procesador interrumpe la actividad que está realizando y atiende la situación. Al finalizar la interrupción, el procesador continúa ejecutando la tarea que había suspendido, o inicia una nueva en tal caso que lo requiera. Las interrupciones de este tipo las producen fuentes como una tecla al presionarla y soltarla, otros pueden ser periféricos como la impresora, el disco, el puerto serie, etc.

Estas interrupciones se producen por las señales de los periféricos, no son programadas y pueden ocurrir en cualquier instante.

-Interrupciones por software: Se producen por un programa mientras este está corriendo, estas interrupciones comúnmente se conocen como “Llamadas al sistema”. Se puede clasificar este tipo de interrupción en dos categorías, las interrupciones del sistema y del usuario.

Interrupciones del DOS y del BIOS: Estas dos interrupciones se diferencian en que las interrupciones del sistema operativo DOS no son complicadas para usarlas, pero si son más bien lentas, ya que usan el BIOS para lograr lo que se necesita, por el contrario las interrupciones del BIOS son bastante rápidas, pero por ser del hardware son bastante específicas, y dependiendo por ejemplo del fabricante del circuito.

Interrupciones del usuario: el usuario las programa, él determina cuando y donde las ejecuta, comúnmente se usan para Entrada y salida

Excepciones: Son interrupciones sincrónicas para atender causas que hacen que un problema presente dificultades mientras se está ejecutando, y el sistema operativo trata este inconveniente, ejemplos claros pueden ser la división entre cero, un acceso invalido a la memoria.

- ¿Cómo se hace la implementación de interrupciones a nivel de hardware?

En las interrupciones a nivel de hardware lo que se pretende es unir el periférico al procesador del sistema, un caso puntual es tocar un tecla en el teclado del PC, este dispositivo periférico es atendido inmediatamente por el procesador. Los 256 interruptores presentes no pueden llamarse en el mismo instante, para que este no ocurra es importante que al instalar las tarjetas de expansión un interruptor no se utilice para dos dispositivos periféricos diferentes, ya que en tal caso ocurriría un choque y ningún dispositivo periférico cumpliría su tarea, no hay forma que el sistema logre distinguirlos. Así que los dispositivos que necesiten relacionarse con el procesador por interrupciones tienen que poseer una línea única que permita indicar al CPU cuando requiere de él para efectuar una tarea. Esta línea es conocida como IRQ. Estas líneas van hasta el controlador de interrupciones, el cual es un elemento de hardware que maneja las interrupciones y pertenece al procesador. El controlador de interrupciones habilitar o deshabilitar las líneas de interrupciones para no presentar conflictos.

Para llevar a cabo una interrupción a nivel de Hardware se deben cumplir ciertos pasos:

1. Se pausa el proceso que tiene la máquina antes de la interrupción.
2. Es necesario guardar los registros de la última actividad en alguna parte para luego de realizar la interrupción se pueda reanudar el proceso.
3. Se atiende el dispositivo que generó la interrupción.
4. Luego de atender la solicitud se restaura el proceso anteriormente ejecutado y se continúa con la ejecución normal del programa.

- ¿Cómo se implementan las interrupciones por software? Debe quedar claro si el lenguaje de programación importa y si el hardware usado afecta.

Las interrupciones por software se originan por un programa en ejecución. Para su generación el código máquina posee una secuencia de instrucciones que ayudan al programador. Estas interrupciones se generan mientras el programa está en plena ejecución. Para llevar a cabo una interrupción a nivel de software se deben cumplir ciertos pasos:

-Un programa que se encontraba en ejecución, se comunica con el Sistema Operativo, por ejemplo para leer un archivo (Se requiere de un dato exterior, el programa se pausa y pasa a resolver el asunto)

-Dado que no se puede seguir la ejecución de las instrucciones en el programa hasta que no se lea el disco, y mientras no esté el archivo en la memoria principal, todo el proceso se interrumpe, y ahora las instrucciones que se ejecutan no son del programa que venía ejecutándose, ahora son instrucciones del Sistema Operativo, se ordena una interrupción indefinida, se recoge el dato que se solicitó.

-Se halla la subrutina del Sistema Operativo y luego se ejecuta para leer el disco.

-Se realiza la lectura del disco y se verifica una correcta lectura, para luego reanudar la ejecución del programa pausado.

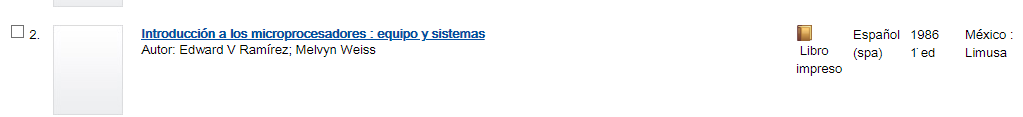
Es importante tener en cuenta que a medida que un programa en un microcontrolador se hace más tesioso y enredado, trabajar con “assembler”, que es el lenguaje de programación de bajo nivel en el que comúnmente se representan las instrucciones básicas para los microprocesadores, computadores, etc. Cuando la complejidad de un programa se aumenta, manejar las interrupciones es una tarea complicada, es importante buscar alternativas para estas implementaciones, por ejemplos usar el lenguaje C, y así los programas pueden simplificarse y volverse mas sencillos y comprensibles.

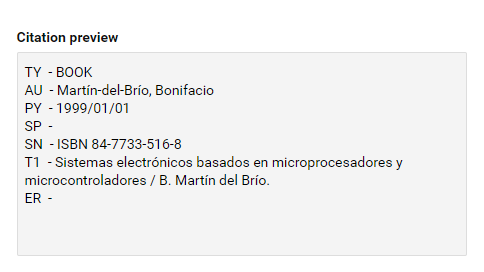
Como parte del hardware, es necesario que el [microprocesador](https://es.wikipedia.org/wiki/Microprocesador), [microcontrolador](https://es.wikipedia.org/wiki/Microcontrolador" \o "Microcontrolador) y otros [circuitos integrados](https://es.wikipedia.org/wiki/Circuito_integrado) programables. Dispongan de una línea especial que debe ser parte del cumulo de líneas de control del bus del sistema y que se denominan línea de petición de interrupción (INT). El módulo de Entrada y salida se comunica con el procesador mediante esta línea y le advierte que esta preparado para hacer la transferencia. El procesador debe contar con un punto de conexión de entrada por donde entraran las interrupciones y el modulo de entrada y salida debe contar con un punto de conexión para la salida por donde se generan las interrupciones.

- Mostrar un ejemplo de interrupción usando la plataforma Arduino. El ejemplo debe ser implementado por usted, sino tiene el Arduino físico puede hacerlo a nivel de simulación. Adjuntar código fuente de la implementación en Arduino. El código debe estar debidamente comentado

Bibliografía:

* **Autores:** Eduardo Santamaría
* **Editores:** [Universidad Pontificia Comillas](https://dialnet.unirioja.es/institucion/360/editor)
* **Año de publicación:** 1993
* **Colecciones:** [Textos Colección Ingeniería](https://dialnet.unirioja.es/servlet/listalibrosporcoleccion?codigo=222), 4
* **País:** España
* **Idioma:** español
* **ISBN:** 84-87840-33-7





<https://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teo/arq-teo08.pdf>

