

Proyecto de investigación - Interrupciones

Valentina Jaramillo Raquejo

valentina.jaramillor@udea.edu.co

29 de junio de 2020

A modo de introducción es importante definir qué es una interrupción en el contexto de los microprocesadores. Una interrupción es un procedimiento que hace que el orden de ejecución de las instrucciones de un programa sea cambiado o siga otra ruta cuando se presenta un acontecimiento importante para el programa, o que podría ser necesario prestarle atención, esto sucede ya que el evento fue manifestado por el hardware de entrada o de salida, y es algo que puede suceder en cualquier momento, por lo que las interrupciones nos dan la posibilidad de tomar el control de estas situaciones sin necesidad de agregar instrucciones que verifiquen el estado del evento en cada ciclo (F.Ingeniería Universidad de la República. Uruguay, s.f.).

Aunque la historia de las interrupciones es extensa, pues hubo muchos protagonistas en la creación de este tipo de implementación, a continuación se pretende dar unos puntos claves de manera cronológica, que ilustran su desarrollo:

- El UNIVAC 1103 / 1103A (1953 / 1956) es citado frecuentemente como el primero en implementar interrupciones. Por otro lado, algunos otros le atribuyen el título al 1103A (1956).

- Otras referencias indican que se desarrolló un sistema de interrupción por los empleados de la NACA Dick Turner y Joe Rawlings a finales de 1955 y que empezó a funcionar en febrero de 1956. Univac proporcionó una versión de este sistema de interrupción a los clientes como una característica del modelo "1103A".

- El trabajo en el sistema de interrupciones para el 1103 se llevó a cabo en el Laboratorio de Propulsión de Vuelo Lewis de la NACA en Cleveland, Ohio. (El laboratorio ahora tiene el nombre del Centro de Investigación John Glenn de la NASA en Lewis Field.) El 1103 fue modificado para que pudiera ser interrumpido para recoger datos en tiempo real de túneles de viento.

- El IBM 650 (1954) realizó el primer uso de enmascaramiento de interrupción. El 650 tenía una opción de consola para pasar automáticamente a una secuencia de reinicio en caso de un error de la máquina.

- El NBS DYSEAC (1954) tuvo las primeras interrupciones de E/S. DYSEAC era un ordenador móvil (transportado en dos remolques de 12 y 8 toneladas, respectivamente) construido para el Cuerpo de Señales del Ejército de los Estados Unidos. DYSEAC tenía dos contadores de programa, y una señal de E/S hacía que la ejecución cambiara de un PC a otro. Un bit en cada instrucción indica qué PC usar para la siguiente instrucción.

- Un artículo de diciembre de 1957 indica que el IBM Stretch tenía un enfoque combinado de interrupciones y ramificaciones condicionales.

- Electrologica X-1 (1957-1958). El X-1 incluía un vector de interrupción de siete niveles, una máscara de interrupción de seis bits y una habilitación de interrupción general de un bit. Las interrupciones funcionaban insertando una de las siete llamadas de subrutinas predefinidas en el flujo de instrucciones (Smotherman, 2017).

Luego de conocer acerca de la historia de las interrupciones, es importante saber ahora sobre los tipos de interrupciones que existen. Podemos clasificarlas dependiendo de su ámbito; si son de hardware o de software, dependiendo también de la periodicidad con la que aparecen y dependiendo de su sincronización con el reloj del sistema. A continuación se hablará más detalladamente de cada categoría.

- Interrupciones de hardware: Si la señal para el procesador es de un dispositivo externo o el hardware se llama interrupciones de hardware. Ejemplo: desde el teclado se presiona la tecla para realizar alguna acción, esta presión en el teclado generará una señal que es dada al procesador para realizar la acción. Tales interrupciones se llaman interrupciones de hardware. Las interrupciones de hardware pueden clasificarse en dos tipos que son:

- o Interrupción enmascarada: Pueden ser retrasadas cuando se ha producido una interrupción de mayor prioridad para el procesador.

- o Interrupción no enmascarable: El hardware que no puede ser retrasado y debe ser procesado por el procesador inmediatamente.

- Interrupciones de software: La interrupción de software también puede dividirse en dos tipos. Estos son:

- o Interrupciones normales: las interrupciones que son causadas por las instrucciones del software se llaman interrupciones del software.

- o Excepción: las interrupciones no planeadas mientras se ejecuta un programa se llaman Excepción. Por ejemplo: mientras se ejecuta un programa, si se obtiene un valor que debe ser dividido por cero se llama excepción.

Con respecto a las interrupciones según la periodicidad de la ocurrencia, se tienen las siguientes:

- Interrupción periódica: Si las interrupciones ocurrieron en un intervalo fijo en la línea de tiempo, entonces esas interrupciones se llaman interrupciones periódicas.

- Interrupción no periódica: Si no se puede predecir la ocurrencia de una interrupción, entonces esa interrupción se llama interrupción no periódica.

Finalmente, las interrupciones según la relación temporal con el reloj del sistema consisten en:

- Interrupción sincrónica: Si la fuente de la interrupción está en fase con el reloj del sistema, se llama interrupción sincrónica. En otras palabras, estas son las interrupciones que dependen del reloj del sistema. Ejemplo: el servicio de temporizador que utiliza el reloj del sistema.

- Interrupción asincrónica: Si las interrupciones son independientes o no están en fase con el reloj del sistema, se llaman interrupciones asincrónicas. (*Types of Interrupts and How to Handle Interrupts*, 2015).

Profundizando más acerca de los dos tipos principales de interrupciones; las de hardware y las de software, sus implementaciones serán explicadas con mayor detalle.

Por parte de las interrupciones de hardware, estas son generadas por los dispositivos de hardware cuando ocurre algo inusual; esto podría ser una pulsación de tecla o un movimiento del mouse o cualquier otra acción. Esto se hace para minimizar el tiempo de ejecución en la CPU, de lo contrario la CPU tendría que comprobar todo el hardware instalado en busca de datos en un gran ciclo (esto se

llama "polling") y esto llevaría mucho tiempo. Por ejemplo, un PC IBM estándar tiene dos controladores de interrupción, que son responsables de estas interrupciones de hardware; ambos permiten hasta 8 fuentes de interrupción diferentes (IRQs, solicitudes de interrupción). El segundo controlador se conecta al primero a través de la IRQ 2 por razones de compatibilidad, por ejemplo, si el controlador 1 obtiene una IRQ 2, le entrega la IRQ al controlador 2. Por esto, se pueden manejar hasta 15 fuentes de interrupción de hardware diferentes. Todos los códigos y datos tocados por estos manejadores deben ser bloqueados (a través de las diferentes funciones de bloqueo) para evitar fallos de página en el momento de la interrupción. Debido a que las interrupciones de hardware son activadas con las interrupciones deshabilitadas, el handler o manejador tiene que habilitarlas antes de volver a la ejecución normal del programa. Adicionalmente, una interrupción de hardware debe enviar un comando EOI (fin de la interrupción) al controlador responsable (*Hardware interrupts*, 2020).

En otro contexto, una interrupción de software es un tipo de interrupción que es causada por una instrucción especial en el conjunto de instrucciones o por una condición excepcional en el propio procesador (*Software Interrupt*, 2016). Se puede introducir una llamada de servicio que permita que un proceso cause una interrupción de software en otro, con un formato de este tipo: `Interrupt(process id, interrupt number)`. Y otra que le permite a un proceso asociar un handler o manejador con una interrupción, de esta forma: `Handle(interrupt number, handler)` (Dewan, 2006).

Las interrupciones de software permiten comunicar sólo un bit de información, el cual significa que ha ocurrido un evento asociado con el número de la interrupción. Normalmente son utilizadas por el sistema operativo para informar un proceso sobre eventos como, por ejemplo, que el usuario tecleó la 'clave de atención', o que una alarma programada por un proceso ha expirado o que se ha superado algún límite, como el tamaño de un archivo o del tiempo virtual.

Es importante distinguir entre interrupciones, trampas, interrupciones de software y excepciones. En todos los casos, un evento se procesa de forma asincrónica por algún procedimiento del handler. Los números de interrupción y de trampa son definidos por el hardware que también es responsable de llamar al procedimiento en el espacio del kernel. Se llama a un handler de interrupciones en respuesta a una señal de otro dispositivo, mientras que se llama a un handler de trampas en respuesta a una instrucción ejecutada dentro de la CPU. Los handlers de interrupciones y excepciones del software son activados en el espacio del usuario y se llama a un handler de interrupción de software en respuesta a la invocación de una llamada al sistema. Los números de interrupción de software son definidos por el sistema operativo y las excepciones son definidas y procesadas por el lenguaje de programación (Dewan, 2006).

La noción de interrupciones de software es algo confusa en algunos entornos como el PC, donde las trampas a las rutinas de E/S proporcionadas por el kernel se denominan interrupciones de software. Hay una instrucción especial en el PC llamada `INT` que se utiliza para invocar estas trampas. Por ejemplo, la instrucción `'int 16H'` ejecuta la rutina de interrupción de la BIOS para procesar el carácter actual recibido del teclado. (Es ejecutada por el handler de interrupciones del kernel Xinu para pedirle al handler de la BIOS del PC que obtenga el carácter del teclado). El término interrupción se utiliza porque estas rutinas son llamadas generalmente por rutinas de interrupción de hardware (Dewan, 2006).

Referencias

- Dewan, P. (2006, 2). *Software Interrupt*. Descargado de <https://www.cs.unc.edu/~dewan/242/s07/notes/ipc/node3.html>
- F.Ingeniería Universidad de la República. Uruguay. (s.f.). *Interrupciones*. Descargado de <https://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teo/arq-teo08.pdf>
- Hardware interrupts*. (2020, 6). Descargado de <https://www.freepascal.org/docs-html/current/rtl/go32/hardwareinterrupts.html>
- Smotherman, M. (2017, 7). *Interrupts*. Descargado de <https://people.cs.clemson.edu/~mark/interrupts.html>
- Software Interrupt*. (2016, 11). Descargado de <https://www.techopedia.com/definition/22195/software-interrupt>
- Types of Interrupts and How to Handle Interrupts*. (2015, 8). Descargado de https://www.electronicshub.org/types-of-interrupts-and-how-to-handle-interrupts/#Types_of_Interrupts