



# ARQUITECTURA DE COMPUTADORAS

---

## Unidad N° 3

- Interrupciones

Profesor: Fabio Bruschetti

Ver 2013-01



# Interrupciones

---

- Una interrupción es un procedimiento desencadenado por un evento externo que produce una transferencia impredecible del control de ejecución, desde el programa o rutina que es interrumpida hacia otra rutina (denominada de atención de la interrupción)
- Una interrupción se compone de 4 etapas
  - Solicitud: a través de una señal asincrónica de entrada a la CPU
  - Reconocimiento: dadas ciertas condiciones temporales y lógicas, la CPU reconoce el pedido y comienza con la atención de la interrupción. La transferencia del control de ejecución no es inmediata ni incondicional.
  - Atención: se ejecuta la rutina (programa) de atención al dispositivo llamada ISR (Interrupt Service Routine)
  - Retorno: Finalizada la ejecución de la rutina de atención, se produce el retorno del control de ejecución a la rutina que fue interrumpida, permitiendo al micro continuar con la ejecución de la misma.

# Interrupciones - Clasificación

## ■ Interrupciones Enmascarables

- Una máscara de interrupción es un bit que condiciona el reconocimiento de la interrupción
- **Es el flag IF (interrupt Flag)**
  - Si IF=0, la solicitud es ignorada. Si IF=1, el pedido es aceptado y se inicia el ciclo de reconocimiento

```
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1575 ES=1575 SS=1575 CS=1575 IP=0100  NU UF EI PL NZ NA PO NC
1575:0100 0000      ADD     [BX+SI],AL      DS:0000=CD
-
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1575 ES=1575 SS=1575 CS=1575 IP=0100  NU UF DI PL NZ NA PO NC
1575:0100 0000      ADD     [BX+SI],AL      DS:0000=CD
-
```

Flag IF=1

Flag IF=0

- Las solicitudes se realizan a través de la señal **INTR** (Interrupt Request)
  - Se sensa en el último ciclo de reloj de la instrucción en curso
- Si es INTR=1, la CPU emite la señal **INTA** (Interrupt Acknowledge) como reconocimiento



# Interrupciones - Clasificación

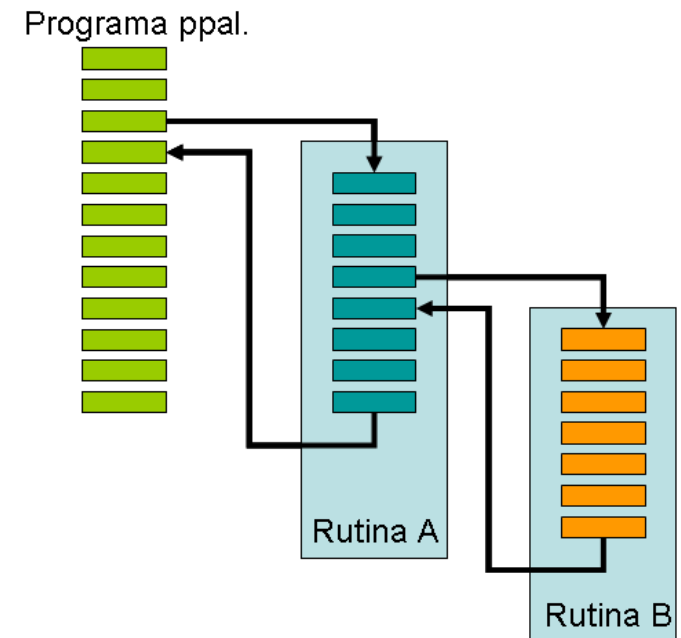
---

## ■ Interrupciones No Enmascarables

- La solicitud se realiza a través de la entrada **NMI** (Non Maskable Interrupt). Es una entrada asincrónica activa por flanco ascendente
- No es necesaria la señal de INTA
- Su atención es inevitable. Es siempre la interrupción más prioritaria ya que puede interrumpir la ejecución de cualquier servicio generado por INTR
- NMI, se usa para situaciones graves que requieran atención incondicional de la CPU

# Interrupciones – Anidamiento

- Una interrupción se anida en otra, cuando esta interrumpe la ejecución de la ISR en curso
- Antes de ejecutar la primera instrucción de la ISR, la CPU coloca el  $IF=0$
- Esto puede evitarse mediante el flag IF
  - Deseo anidamiento
    - La primera instrucción de atención a la interrupción es la habilitación de la CPU a atender nuevas interrupciones (STI)
    - La última instrucción de atención a la interrupción devuelve la posibilidad a la CPU para atender interrupciones (CLI)
  - No deseo anidamiento
    - No modifico el flag IF





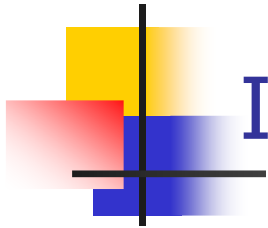
# Interrupciones – Atención de dispositivos

- Reconocimiento: Secuencia de eventos
  - Luego de haber finalizado la instrucción en curso, la CPU se dispone a reconocer al dispositivo solicitante
  - El dispositivo solicitante, coloca en el bus de datos 1 byte correspondiente a su identificación. A ese byte se lo llama “tipo” (Interrupt Type)
  - La CPU efectúa un ciclo de lectura y toma el “tipo” y con él tiene que determinar en que dirección de MP está la ISR para comenzar a ejecutarla
  - Como una dirección de memoria se forma con 2 registros de 16 bits (SEGMENT:OFFSET), necesita un puntero de 4 bytes para ubicar la ISR en la MP. En nuestro caso, los registros son **CS:IP**
  - Con 1 byte puedo tener 256 “tipos” o dispositivos distintos
  - Si necesito 4 bytes de puntero para cada uno de ellos → Tendré un espacio en MP de 1024 bytes (1K) con todos los punteros. Este espacio de 1K se llama **Vector de Interrupciones (IVT)**

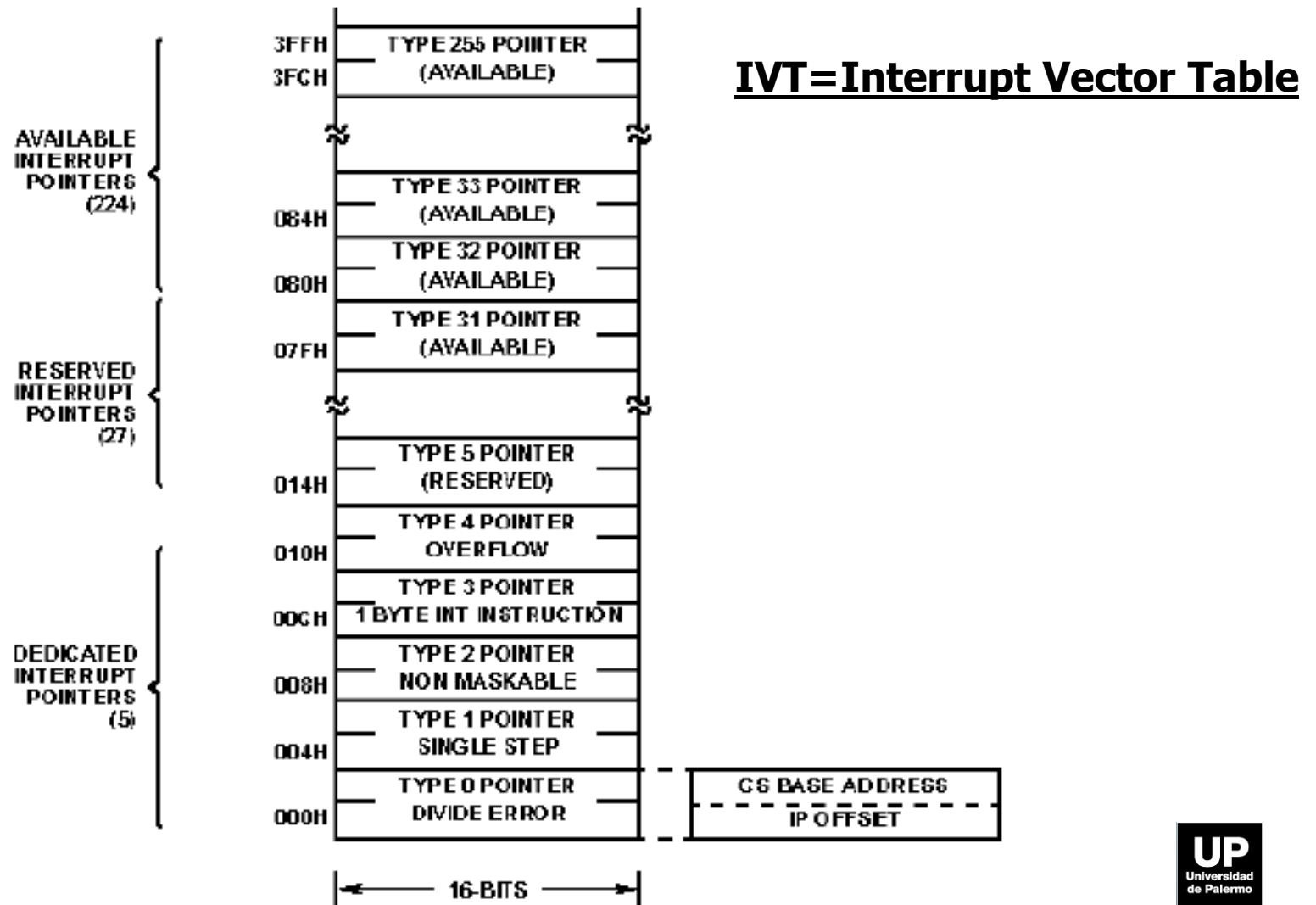


# Interrupciones – Atención de dispositivos

- Vector de interrupciones (IVT)
  - Se encuentra en los primeros 1024 bytes del mapa de memoria. Va desde la dirección:  $00000_h$  a la  $003FF_h$
  - Como se ubica en la MP puede ser modificado durante la ejecución de cualquier programa
  - Entradas del vector de interrupciones (4 bytes cada una)
    - Está compuesta por dos bytes para alojar el IP y dos para el CS
    - El sentido es:
      - LSB – IP
      - MSB – IP
      - LSB – CS
      - MSB – CS.



# Interrupciones – Atención de dispositivos







# Pseudo Interrupciones

---

## ■ Interrupciones por SW

- Son instrucciones que provocan que la CPU ejecute alguna de las rutinas de atención de interrupciones del “tipo” que se indique. Finalizada la ejecución de la rutina de atención, la CPU continúa ejecutando la instrucción posterior a la instrucción de interrupción
- No son interrupciones propiamente dichas ya que al ser instrucciones, son predecibles
- Como son instrucciones, no respetan ningún esquema de prioridades y se anidan siempre que se incluyan dentro de una rutina de atención de interrupción. Obviamente no son enmascarables
- Estas instrucciones pueden usarse para transferir el control a rutinas que son reubicables en memoria (en las que el programa que las llama no conoce la dirección de inicio de las mismas)
- Salvan en el stack el registro de banderas
- No ejecutan el ciclo de INTA, pero si inhiben Single Step y IF durante su ejecución



# Pseudo Interrupciones

- Instrucciones de interrupciones por SW

- INT3

- 1 byte de longitud
- Interrupción de "tipo"=3
- Se utiliza para insertar puntos de ruptura en rutinas que están siendo depuradas

- INT  $X_h$

- Ejecuta el servicio de la interrupción del "tipo"=n
- Ocupa dos bytes, uno para el código de operación y otro para el dato del "tipo"
- Se puede usar para simular cualquier tipo de interrupción y para depuración
- Como puede manejar varios servicios una rutina ISR?
  - El tipo de servicio es pasado como parámetro
  - Los parámetros son pasados por registros, no por el stack.
  - El valor de retorno, depende del parámetro de código de servicio.



# Pseudo Interrupciones

---

- Instrucciones de interrupciones por SW
  - INTO
    - "Tipo"=4
    - Interrumpe el programa si existe un overflow indicado por la activación del flag OF. Si  $OF = 0$  la instrucción no ejecuta nada (se comporta como un NOP)
    - Esta instrucción permite corregir errores en el caso de overflow



# Pseudo Interrupciones

- Ejemplos de interrupciones por SW
  - Funciones de DOS: (Int 21<sub>h</sub>)
    - Imprimir un mensaje: DS:DX = dir. Mensaje; AH = 9; int 21<sub>h</sub>
    - Salir AH = 4C; int 21<sub>h</sub>
    - Leer un carácter AH = 1; int 21<sub>h</sub> --> AL =caracter ASCII
    - Imprimir un carácter DL = caracter ASCII ; AH = 2; int 21h
    - Salida a Impresora DL = caracter ASCII ; AH = 5; int 21h
  - Servicios del BIOS
    - Display (Int 10<sub>h</sub>)
    - Funciones de E/S a discos (Int 13<sub>h</sub>)
    - Teclado (Int 16<sub>h</sub>) [Lectura AH=0]
    - Impresoras (Int 17<sub>h</sub>) para funciones de impresora



# Pseudo Interrupciones

---

- Interrupciones de error

- Se producen por lo general por un error de software
- No respetan esquema de prioridades
- En el 8086 hay solo una: "Tipo"=0 – Error al dividir
  - Ocurre cuando se intenta dividir por 0
  - La dirección de retorno de la rutina de atención de este tipo apunta a la misma instrucción que se terminó de ejecutar antes de producirse este error
  - Permite corregir el error que provocó la interrupción en la rutina de atención y volver a empezar
  - No es enmascarable.



# Pseudo Interrupciones

---

- Interrupciones de Trampa (Modo paso a paso)
  - Si el bit TF (Trap Flag) está activo, después de la ejecución de cada instrucción se produce una interrupción “tipo”=1
  - Al aceptar esta interrupción, se borra el TF de manera de ejecutar la rutina de atención en forma normal
  - Se usa para implementar una rutina para el monitoreo durante la depuración de programas
  - Comienza a ejecutarse una instrucción después de que se activó el bit TF
  - No es enmascarable



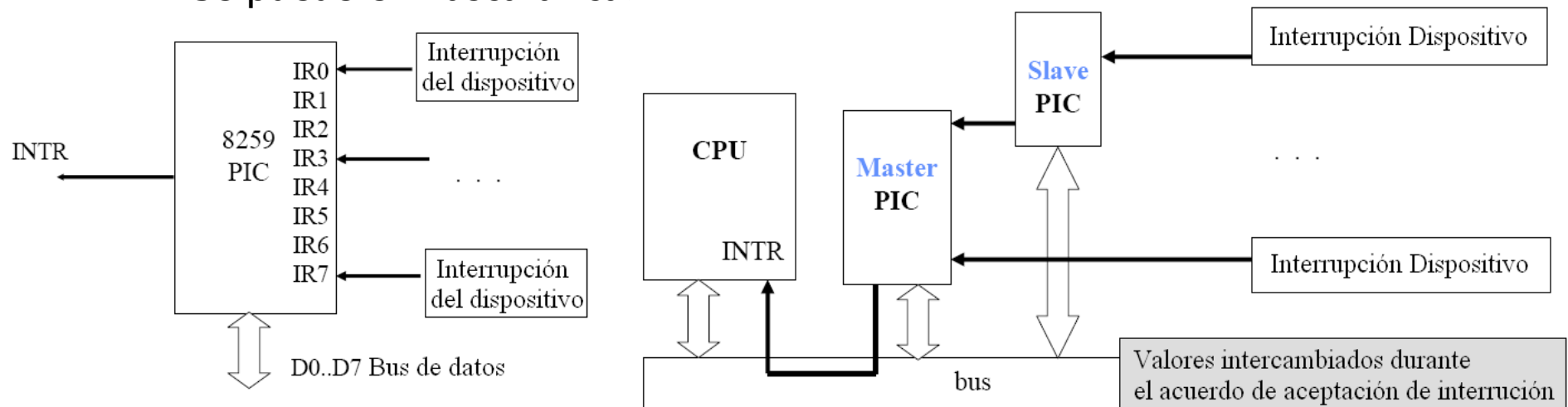
# Interrupciones – Secuencia

Cada vez que la CPU finaliza la ejecución de una instrucción, comprueba:

- Si está activa la trampa (TF=1)
- Si hubo pedido NMI
- Si hay overflow
- Si la próxima es una instrucción INT
- Si se da alguna de estas condiciones, entonces se produce la siguiente secuencia:
  - Si hay un INTR, se realiza la secuencia de reconocimiento de interrupción
  - Se salvan los flags en la pila
  - Se desactivan la máscara de interrupción IF y el TF
  - Se salva el CS en la pila
  - Se salva el IP en la pila
  - Se recupera el contenido del vector de interrupción y se carga el CS y el IP con su contenido
  - la CPU continúa ejecutando en la atención de la interrupción

# Controlador de Interrupciones

- Intel 8259A (PIC = Programmable Interrupt Controller) - Chipset
  - Amplía la estructura de interrupciones del 8086
  - Es configurable y programable
  - Se lo considera como un dispositivo más del chipset
  - Conectándolo 8 8259A en cascada (8+1) → maneja 64 interrupciones
  - Durante el booteo, la BIOS programa mapea las IR0 a IR7 a los tipos 08<sub>h</sub> a 0F<sub>h</sub>. La IR0 tiene la mayor prioridad
  - Se puede enmascarar cada una de las IR por separado

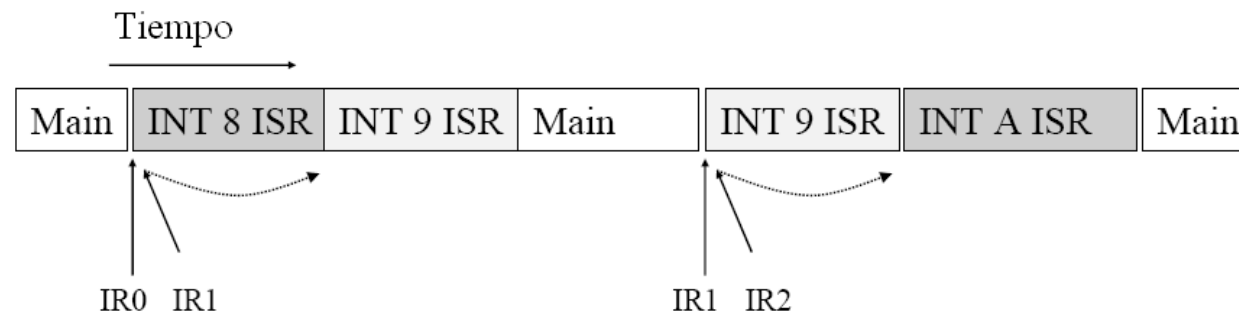




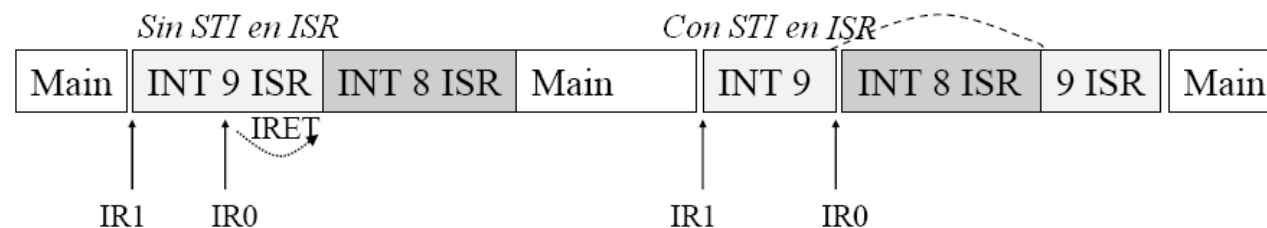
# Controlador de Interrupciones

- Prioridad de Interrupción

- Que rutina ISR se ejecuta si dos interrupciones ocurren al mismo tiempo



- Si la CPU esta ejecutando una rutina ISR y un segundo dispositivo (de mayor prioridad) interrumpe, ¿que ISR sigue ejecutando?

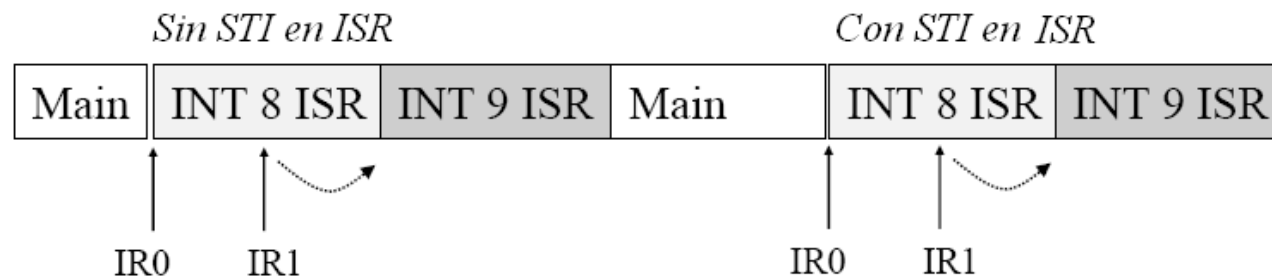


- El controlador intentará permitir a una interrupción de mayor prioridad interrumpir otra de menor. La segunda interrupción no será reconocida por el procesador hasta que IF=1 (habiliten las interrupciones)

# Controlador de Interrupciones

- Prioridad de Interrupción

- Si una interrupción de baja prioridad sucede a otra de alta prioridad, el controlador mantiene la prioridad.



- El controlador recuerda la interrupción de baja prioridad hasta que la de alta prioridad haya terminado
- Cuando terminó, el controlador genera otra interrupción a favor del dispositivo lento
  - ¿Cuántas interrupciones puede recordar el controlador?
  - ¿Cómo sabe el controlador que la interrupción de alta prioridad terminó?



# Interrupciones de la BIOS

---

- Las Interrupciones de la BIOS

- INT 10h: Servicios de Vídeo (texto y gráficos).
- INT 11h: Informe sobre la configuración del equipo.
- INT 12h: Informe sobre el tamaño de la memoria convencional.
- INT 13h: Servicios de disco (muy elementales: pistas, sectores, etc.).
- INT 14h: Comunicaciones en serie.
- INT 15h: Funciones cassette (PC) y servicios especiales del sistema (AT).
- INT 16h: Servicios de teclado.
- INT 17h: Servicios de impresora.
- INT 18h: Llamar a la ROM del BASIC (sólo máquinas IBM).
- INT 19h: Reinicialización del sistema.
- INT 1Ah: Servicios horarios.
- INT 1Fh: Apunta a la tabla de los caracteres ASCII 128-255 (8x8 puntos).



# Lista de Interrupciones

---

- Las cinco Interrupciones dedicadas (0 a 4)
  - Interrupción 0 (división por cero)
    - Invocada por la CPU después de un DIV o IDIV, si el divisor es 0 o el resultado no entra en el destino
  - Interrupción 1 (paso a paso)
    - Usado por los debuggers para soportar pasos simples
  - Interrupción 2 (no enmascarable)
    - Interrupciones x hardware que no puede ser deshabilitada
  - Interrupción 3 (breakpoint)
    - Usado por los debuggers para establecer puntos de detención en el programa
  - Interrupción 4 (overflow)
    - Usado en librerías numéricas para atrapar errores de overflow



# Lista de Interrupciones

---

- Las Interrupciones de Hardware
  - INT 8h: Se produce con una frecuencia periódica determinada por el IR0 del chip temporizador 8253/8254 (en la práctica, unas 18,2 veces por segundo).
  - INT 9<sub>h</sub>: Generada al pulsar o soltar una tecla
  - INT 0A<sub>h</sub>, 0B<sub>h</sub>, 0C<sub>h</sub>, 0D<sub>h</sub>, 0E<sub>h</sub>, 0F<sub>h</sub>: Puertos serie, impresora y controladores de disquete
  - INT 70<sub>h</sub>, 71<sub>h</sub>, 72<sub>h</sub>, 73<sub>h</sub>, 74<sub>h</sub>, 75<sub>h</sub>, 76<sub>h</sub>, 77<sub>h</sub>: Generadas en los ATs y maquinas superiores por el segundo chip controlador de interrupciones.



# Interrupciones del DOS

---

- Las Funciones del DOS

- INT 20<sub>h</sub>: Terminar programa (tal vez en desuso).
- INT 21<sub>h</sub>: Servicios del DOS. (devuelve acarreo si hay error)
- INT 22<sub>h</sub>: Control de finalización de programas.
- INT 23<sub>h</sub>: Tratamiento de Ctrl-C.
- INT 24<sub>h</sub>: Tratamiento de errores críticos.
- INT 25<sub>h</sub>: Lectura absoluta de disco (sectores lógicos).
- INT 26<sub>h</sub>: Escritura absoluta en disco (sectores lógicos).
- INT 27<sub>h</sub>: Terminar dejando residente el programa (en desuso).
- INT 28<sub>h</sub>: Idle (ejecutada cuando el ordenador está inactivo).
- INT 29<sub>h</sub>: Impresión rápida en pantalla (no tanto).
- INT 2A<sub>h</sub>: Red local MS NET.
- INT 2B<sub>h</sub>-2D<sub>h</sub>: Uso interno del DOS.
- INT 2E<sub>h</sub>: Procesos Batch.
- INT 2F<sub>h</sub>: Interrupción Multiplex.
- INT 30<sub>h</sub>-31<sub>h</sub>: Compatibilidad CP/M-80.
- INT 32<sub>h</sub>: Reservada.