



Universidad Nacional  
de Entre Ríos

Tecnicatura universitaria en desarrollo web

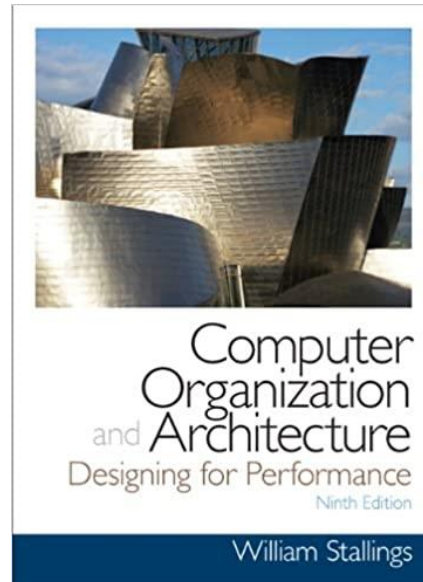
# Interrupciones

---

Semana 7 – Arquitectura de computadoras

Esta presentación esta basada en el libro de:

- ❑ William Stallings, Computer Organization and Architecture, 9th Edition, 2017.



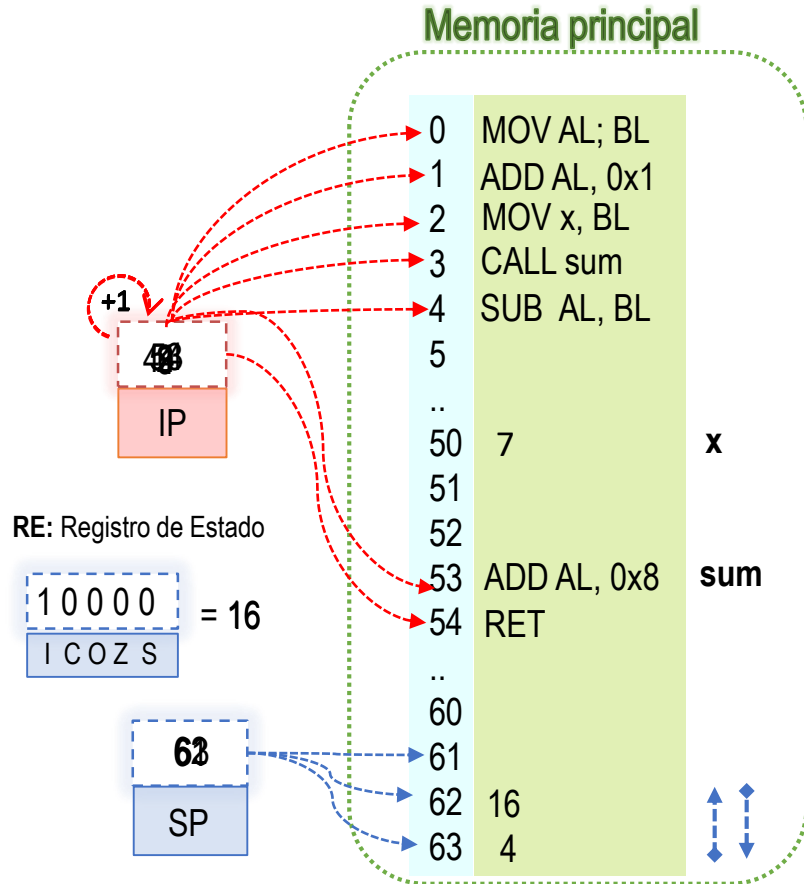
Archivos presentación y ejemplos se alojan en:



<https://github.com/ruiz-jose/tudw-arq.git>

# Interrupciones

- Instrucción CALL/RET
- Arranque del sistema
- Vector de interrupciones
- Tipos de interrupciones:
  - Interrupción Software: Instrucción INT/IRET
  - Interrupción Hardware
  - Excepción



**CALL** dirección → llamada a una rutina

**CALL sum**

- a) PUSH *IP*: {[SP] ← IP ; SP ← SP – 1}
- b) PUSH *RE*: {[SP] ← RE ; SP ← SP – 1}
- c) IP ← dirección

**RET** → retorno al programa llamador.

**RET**

- a) POP *RE*: {SP ← SP + 1 ; RE ← [SP]}
- b) POP *IP*: {SP ← SP + 1 ; IP ← [SP]}

**RE:** Registro de Estado

1 0 0 0 0 = 16

I C O Z S

0

Presionamos el botón de encendido, llega energía y arranca el sistema

1

El CPU comienza a ejecutar el **BIOS (Basic Input Output System)**, que se encuentra en la memoria **ROM** de la placa madre

2

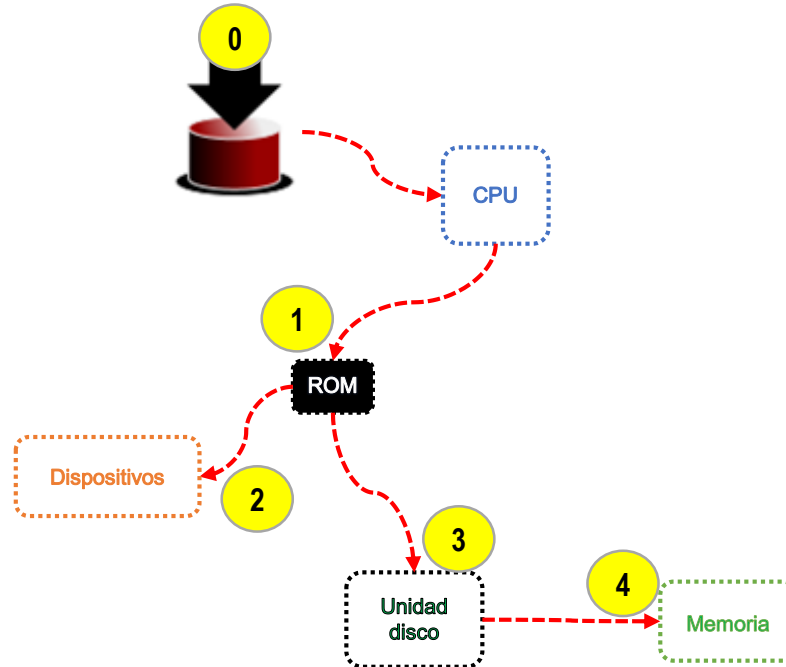
La BIOS se encargar de correr una serie de diagnósticos llamados **POST (Power On Self Test)**

3

Busca la Unidad «bootable», y copia el «bootloader» a la memoria y comienza ejecutarlo

4

El «bootloader» carga el sistema operativo en memoria y sede el control al mismo.



Cuando la computadora se enciende, el BIOS y el SO establecen una tabla de rutinas de tratamiento de interrupciones en las primeras localidades de memoria.

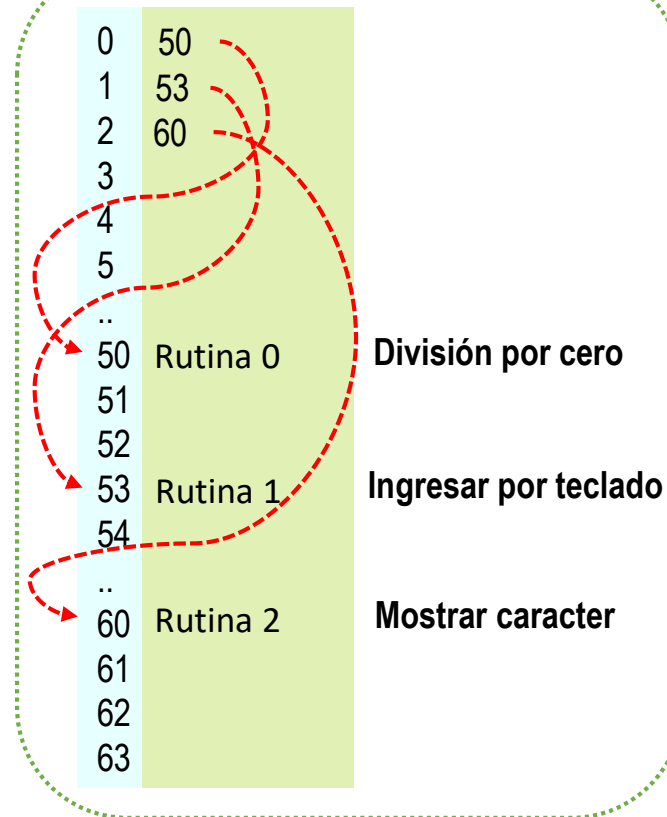
### ❑ Tabla de vectores de Interrupción

Refleja las direcciones de las distintas rutinas de tratamiento de interrupciones.

## ¿Quién invoca la rutina de tratamiento de interrupción?

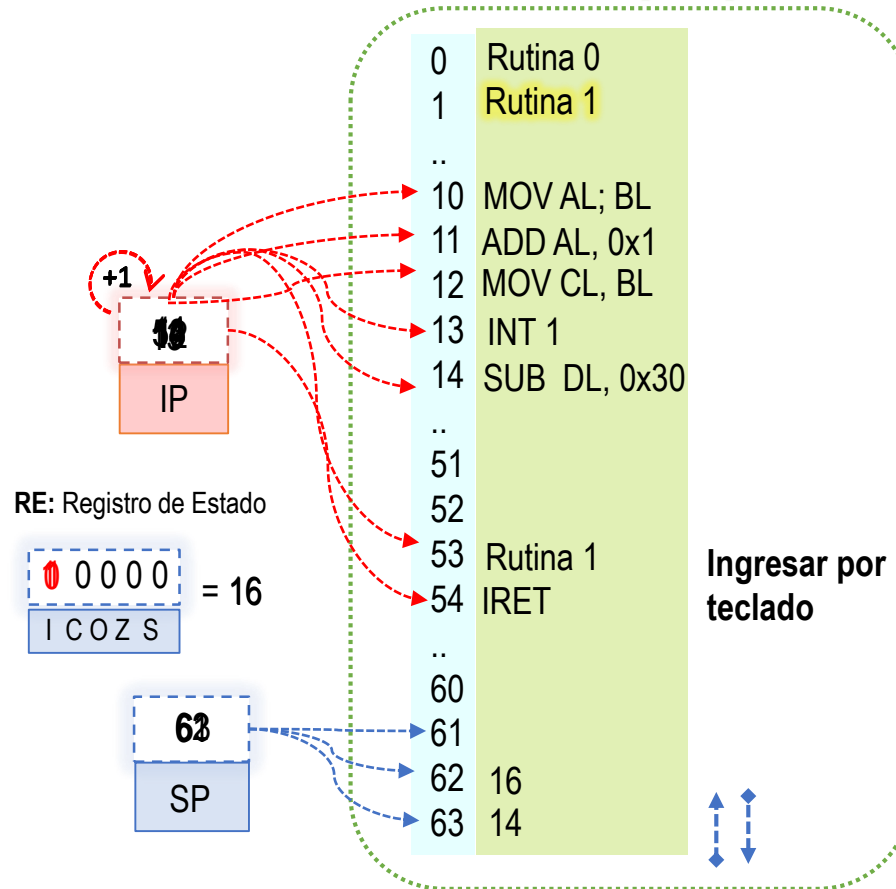
- Interrupción Software: Instrucción INT/IRET
- Interrupción Hardware
- Excepción

### Memoria principal



## Interrupción por Software

## Memoria principal



## ¿Cómo se invoca a estas rutinas?

**INT** Nro\_INT → llamada a una rutina de interrupción.

**INT 1** → Llama a la rutina 1 del vector de interrupciones devuelve en DL el código ASCII del dato ingresado

**Nuevo flag I:**

que indica si el procesador puede ser interrumpido o no.

**RE: Registro de Estado**

1 0 0 0 0 = 16  
I C O Z S

**INT 1**

- a) PUSH IP: {[SP] ← IP ; SP ← SP - 1}
- b) PUSH RE: {[SP] ← RE ; SP ← SP - 1}
- c) I ← 0
- d) IP ← [1]

**IRET** → retorno al programa llamador.

**IRET**

- a) POP RE: {SP ← SP + 1 ; RE ← [SP]}
- b) POP IP: {SP ← SP + 1 ; IP ← [SP]}
- c) I ← 1



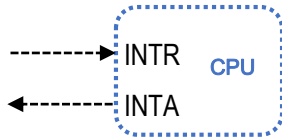


## ❑ Interrupciones por hardware

### El procesador 8086

Incluye dos nuevas señales:

- ❖ Entrada: INTR (Solicitud interrupción)
- ❖ Salida: INTA (Interrupción reconocida)



### ¿Qué pasa cuando un dispositivo interrumpe al CPU?

Cuando se produce una interrupción si el dispositivo de E/S activa la señal de interrupción (INTR) y el flag I vale 1, termina de ejecutar la instrucción en curso y realiza atómicamente los siguientes pasos:

- PUSH IP:  $\{[SP] \leftarrow IP ; SP \leftarrow SP - 1\}$
- PUSH RE:  $\{[SP] \leftarrow RE ; SP \leftarrow SP - 1\}$
- $I \leftarrow 0$  : para evitar que el procesador vuelva a interrumpirse
- INTA: Activa la señal INTA para indicarle al dispositivo que atenderá su pedido.
- $IP \leftarrow [\text{dirección rutina servicio de interrupción}]$

Nota: el servicio de interrupción es responsable de preservar el valor actual de los registros y restaurar el registro de estado y PC, mediante la instrucción de retorno de interrupción IRET

0 Presionamos la tecla

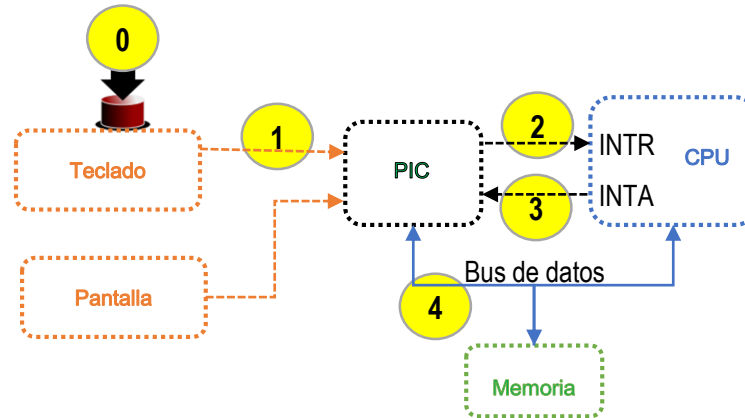
1 El PIC (Programmable Interrupt Controller) solicita al CPU ser atendido

2 CPU recibe la solicitud

3 CPU avisa que va atenderla

4 PIC envía por el bus de datos el Nro de rutina que tiene que buscar el CPU en el vector de interrupciones

¿Cómo identificar el dispositivo que genero la Interrupción?



## ❑ Excepciones (división por cero)

**DIV** (división por cero)

```
MOV AX, 203 ; AX = 00CBh
```

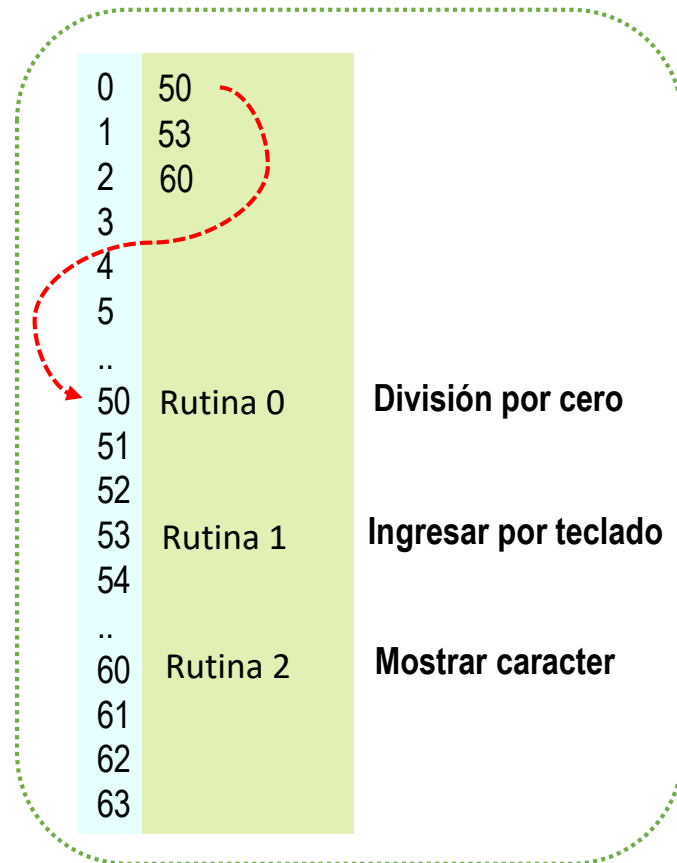
```
MOV BL, 0
```

```
DIV BL
```

```
RET
```

**Al producirse una división por cero el CPU invoca a la rutina 0 del vector de interrupciones**

### Memoria principal



# Preguntas?