

# *Introducción a los Sistemas Operativos*

## *Anexo – Arquitectura de Comp.*

### **Profesores:**

Lía Molinari

Juan Pablo Pérez

Nicolás Macia

Nicolás del Rio



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

# I.S.O.

- ✓ Versión: Agosto 2018
- ✓ Palabras Claves: Sistemas Operativos, Hardware, Interrupciones, Registros

Los temas vistos en estas diapositivas han sido mayormente extraídos del libro de William Stallings (Sistemas Operativos: Aspectos internos y principios de diseño)



# *Elementos Básicos de una computadora*

- ☑ Procesador
- ☑ Memoria Principal
  - ✓ Volátil
  - ✓ Se refiere como memoria real o primaria
- ☑ Componentes de E/S
  - ✓ Dispositivos de memoria secundaria
  - ✓ Equipamiento de comunicación
  - ✓ Monitor / teclado / mouse
- ☑ Bus Sistema
  - ✓ comunicación entre procesadores, memoria, dispositivos de E/S



# Componentes de alto nivel

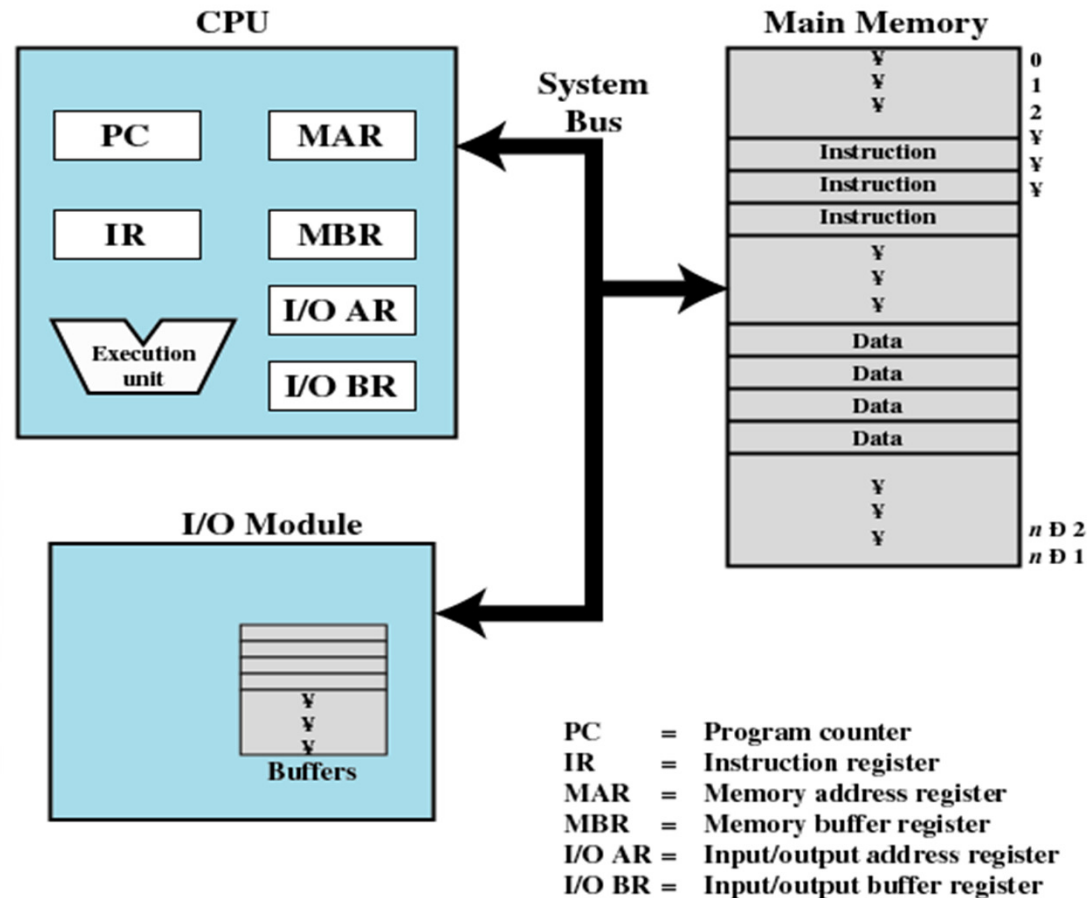


Figure 1.1 Computer Components: Top-Level View



# Registros del Procesador

## ☑ Visibles por el usuario

- ✓ Registros que pueden ser usados por las aplicaciones

## ☑ De Control y estado

- ✓ Para control operativo del procesador
- ✓ Usados por rutinas privilegiadas del SO para controlar la ejecución de procesos



# *Registros Visibles por el usuario*

- ☑ Pueden ser referenciados por lenguaje de máquina
- ☑ Disponible para programas/aplicaciones
- ☑ Tipos de registros
  - ✓ Datos
  - ✓ Direcciones
    - ◆ Index
    - ◆ Segment pointer
    - ◆ Stack pointer



# Registros de Control y Estado

- ☑ Program Counter (PC)
  - ✓ Contiene la dirección de la próxima instrucción a ser ejecutada
- ☑ Instruction Register (IR)
  - ✓ Contiene la instrucción a ser ejecutada
- ☑ Program Status Word (PSW)
  - ✓ Contiene códigos de resultado de operaciones
  - ✓ habilita/deshabilita Interrupciones
  - ✓ Indica el modo de ejecución (Supervisor/usuario)



# *Ciclo Ejecución de Instrucción*

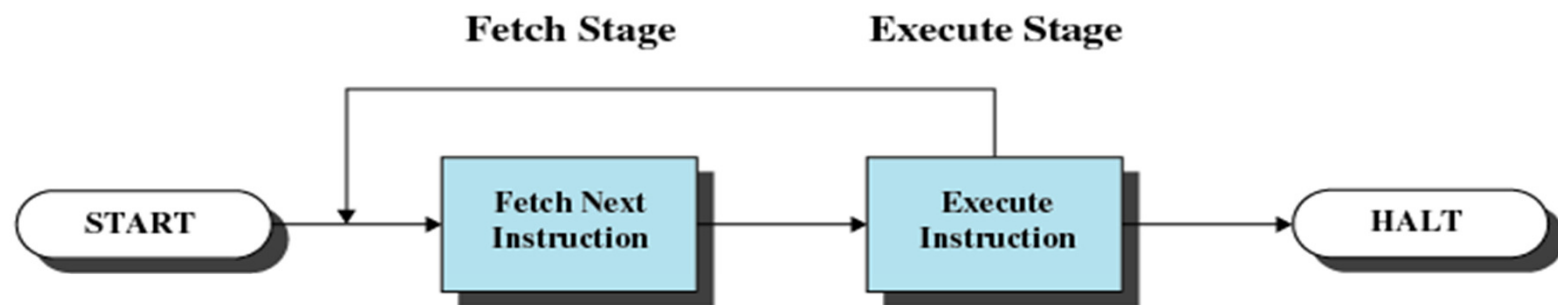
## ☑ Dos pasos

- ✓ Procesador lee la instrucción desde la memoria
- ✓ Procesador ejecuta la instrucción





# Ciclo Instrucción



**Figure 1.2 Basic Instruction Cycle**



# *Instrucción: Fetch y Execute*

- ✓ El procesador busca (fetch) la instrucción en la memoria
  - $(PC) \rightarrow IR$
- ✓ El PC se incrementa después de cada fetch para apuntar a la próxima instrucción
  - $PC = PC + 4$

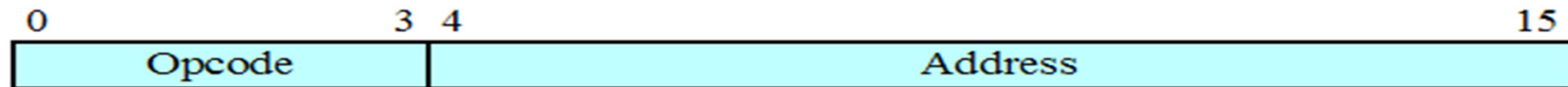


# IR - Instruction Register

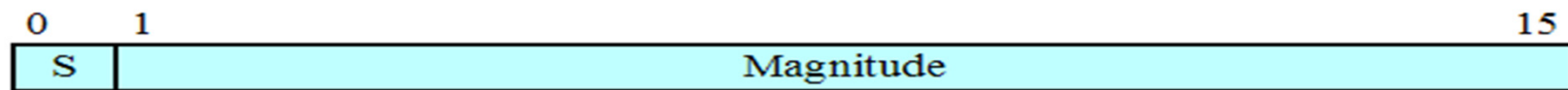
- ☑ La instrucción referenciada por el PC se almacena en el IR y se ejecuta
- ☑ Categorías de instrucciones
  - ✓ Procesador - Memoria
    - ◆ Transfiere datos entre procesador y memoria
  - ✓ Procesador - E/S
    - ◆ Transfiere datos a/o desde periféricos
  - ✓ Procesamiento de Datos
    - ◆ Operaciones aritméticas o lógicas sobre datos
  - ✓ Control
    - ◆ Alterar secuencia de ejecución



# Características de una máquina hipotética



**(a) Instruction format**



**(b) Integer format**

Program Counter (PC) = Address of instruction  
Instruction Register (IR) = Instruction being executed  
Accumulator (AC) = Temporary storage

**(c) Internal CPU registers**

0001 = Load AC from Memory  
0010 = Store AC to Memory  
0101 = Add to AC from Memory

**(d) Partial list of opcodes**

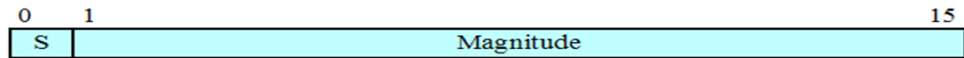
**Figure 1.3 Characteristics of a Hypothetical Machine**



# Ej. de una ejecución de programa



(a) Instruction format



(b) Integer format

Program Counter (PC) = Address of instruction  
 Instruction Register (IR) = Instruction being executed  
 Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from Memory  
 0010 = Store AC to Memory  
 0101 = Add to AC from Memory

(d) Partial list of

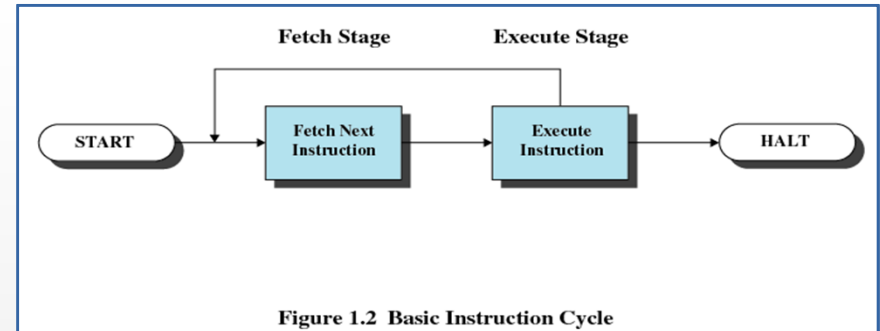
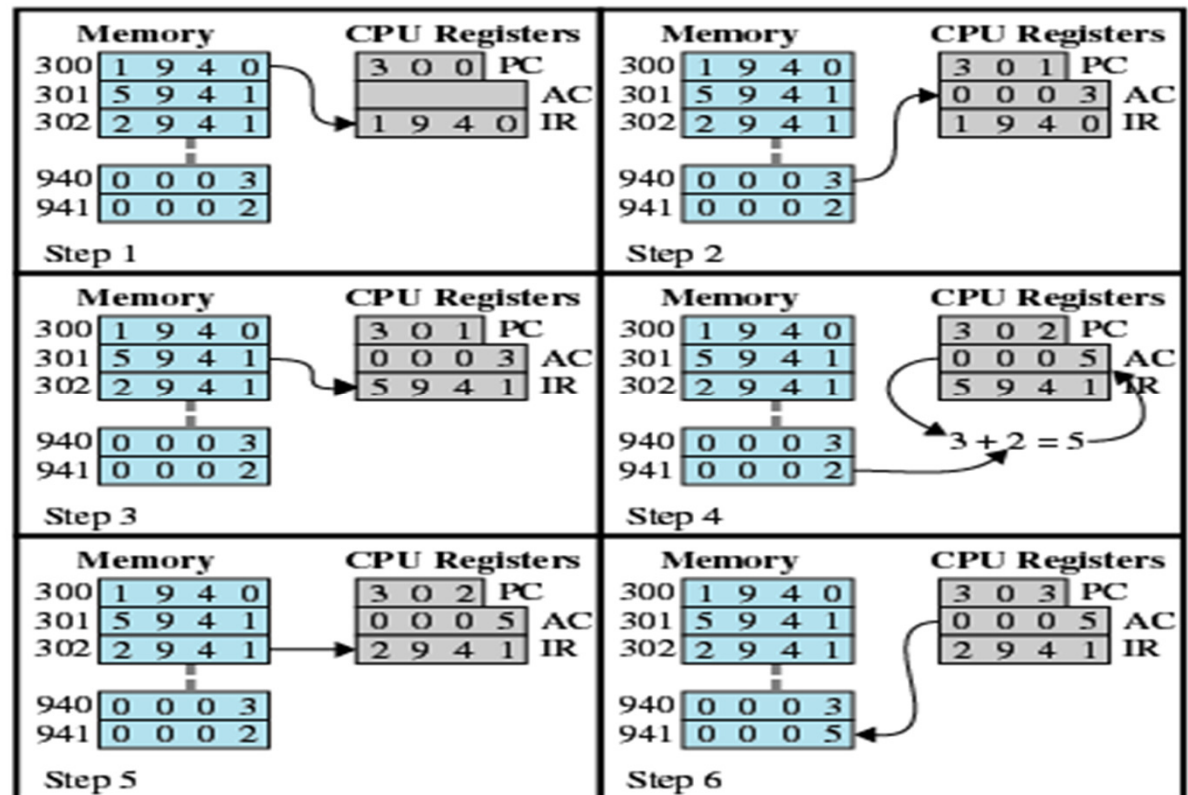


Figure 1.2 Basic Instruction Cycle

Figure 1.3 Characteristics of



# Interrupciones

- ☑ Interrumpen el secuenciamiento del procesador durante la ejecución de un proceso
- ☑ Dispositivos de E/S más lentos que el procesador
  - ✓ Procesador debe esperar al dispositivo



# Clases de Interrupciones

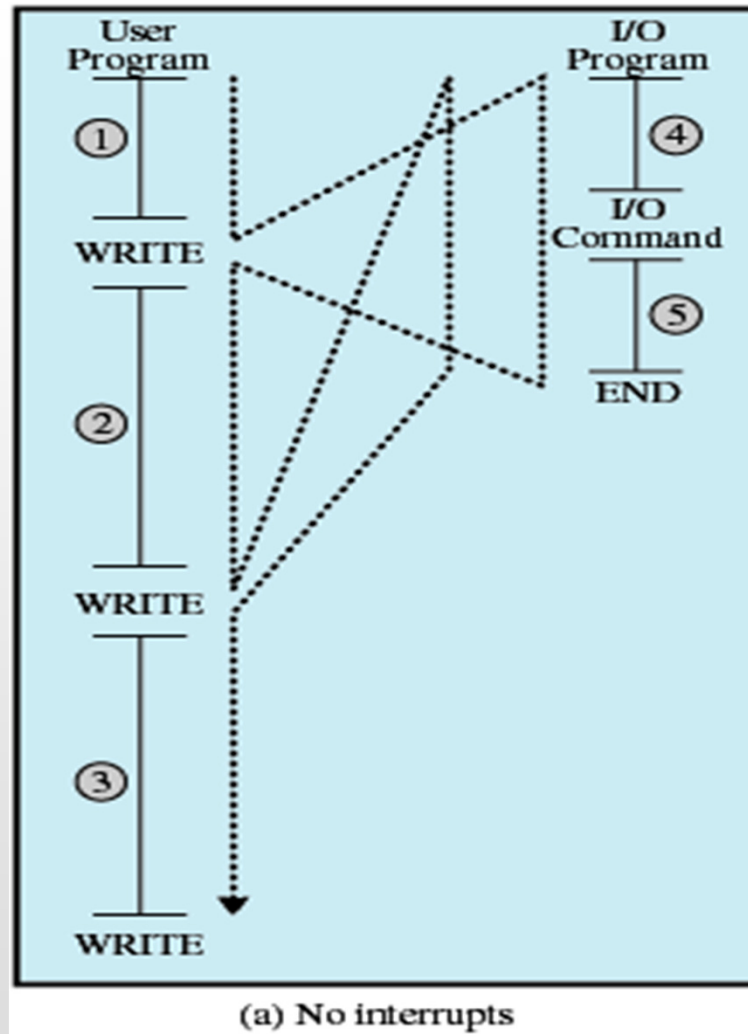
**Table 1.1**    **Classes of Interrupts**

<b>Program</b>	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
<b>Timer</b>	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
<b>I/O</b>	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
<b>Hardware failure</b>	Generated by a failure, such as power failure or memory parity error.



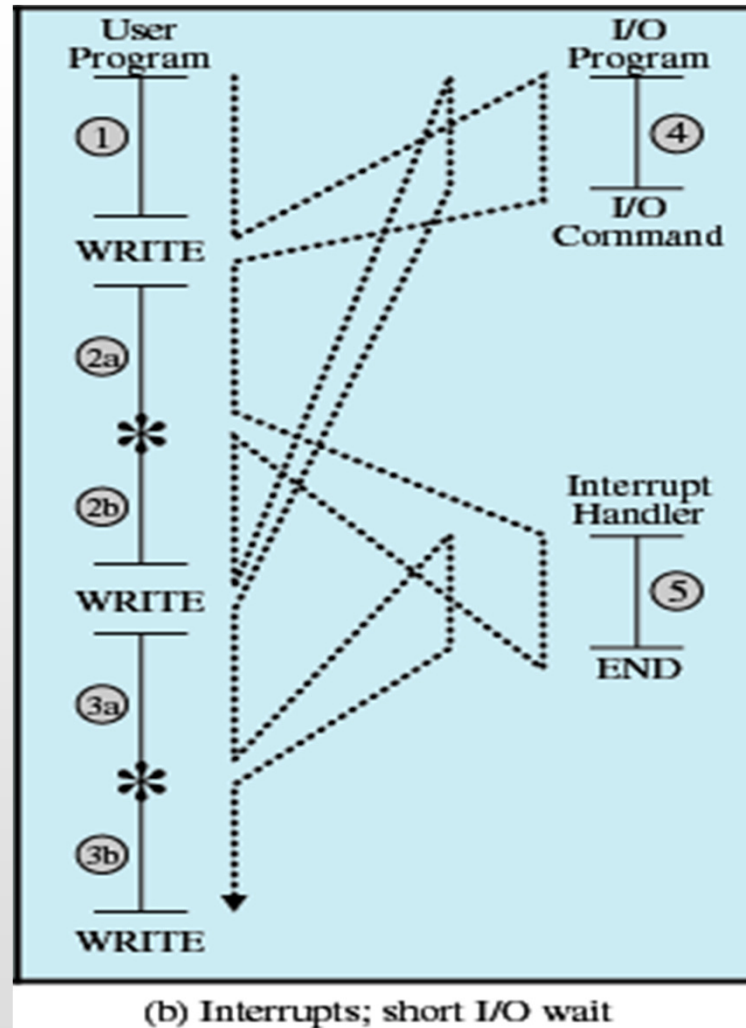


# Flujo de control SIN interrupciones





# Flujo de control CON interrupciones



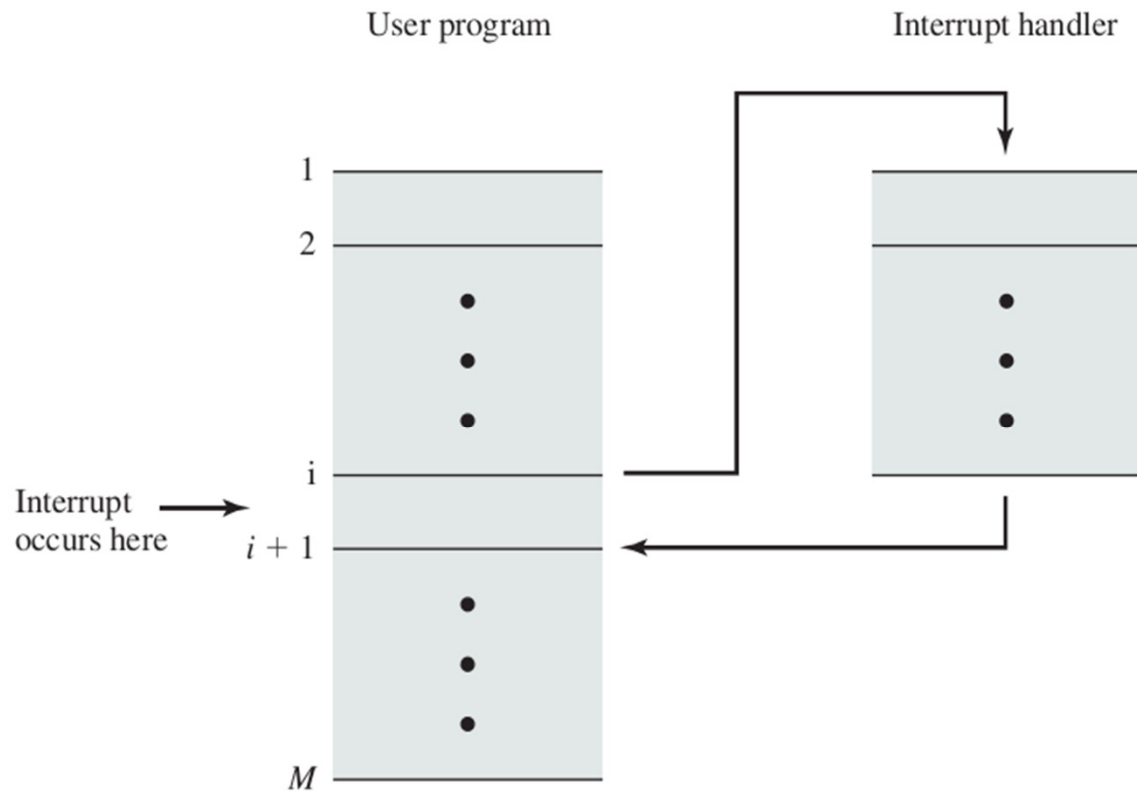
# *Interrupt Handler*

- ☑ Programa (o rutina) que determina la naturaleza de una interrupción y realiza lo necesario para atenderla
  - ✓ Por ejemplo, para un dispositivo particular de E/S
- ☑ Generalmente es parte del SO



# Interrupciones

☑ Suspende la secuencia normal de ejecución



**Figure 1.6** Transfer of Control via Interrupts



# Ciclo de interrupción

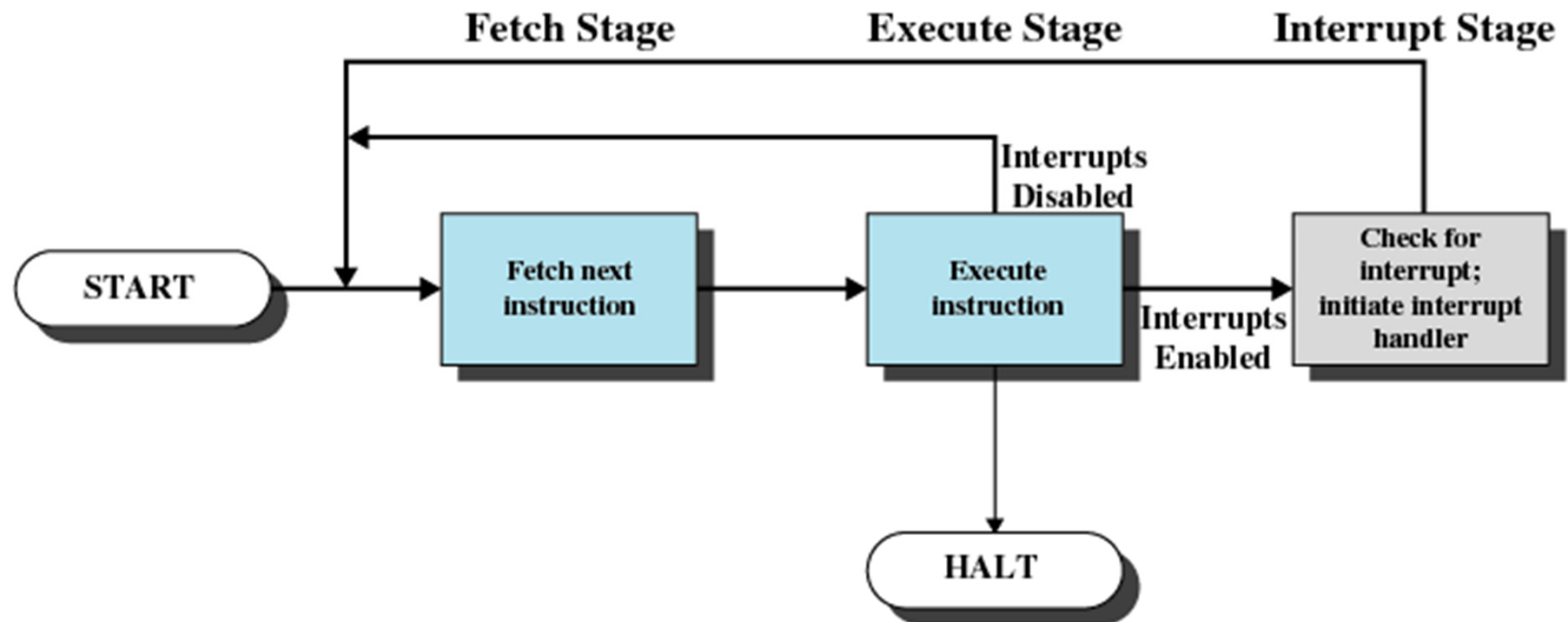


Figure 1.7 Instruction Cycle with Interrupts



# *Ciclo de interrupción*

- ✓ El procesador chequea la existencia de interrupciones.
- ✓ Si no existen interrupciones, la próxima instrucción del programa es ejecutada
- ✓ Si hay pendiente alguna interrupción, se suspende la ejecución del programa actual y se ejecuta la rutina de manejo de interrupciones.



# Simple Interrupt Processing

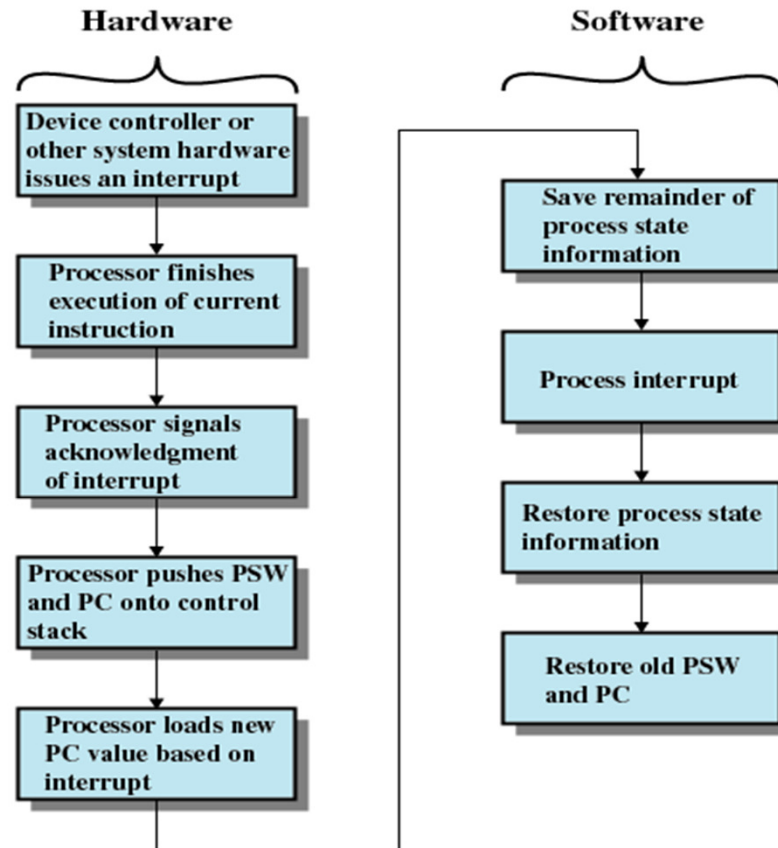
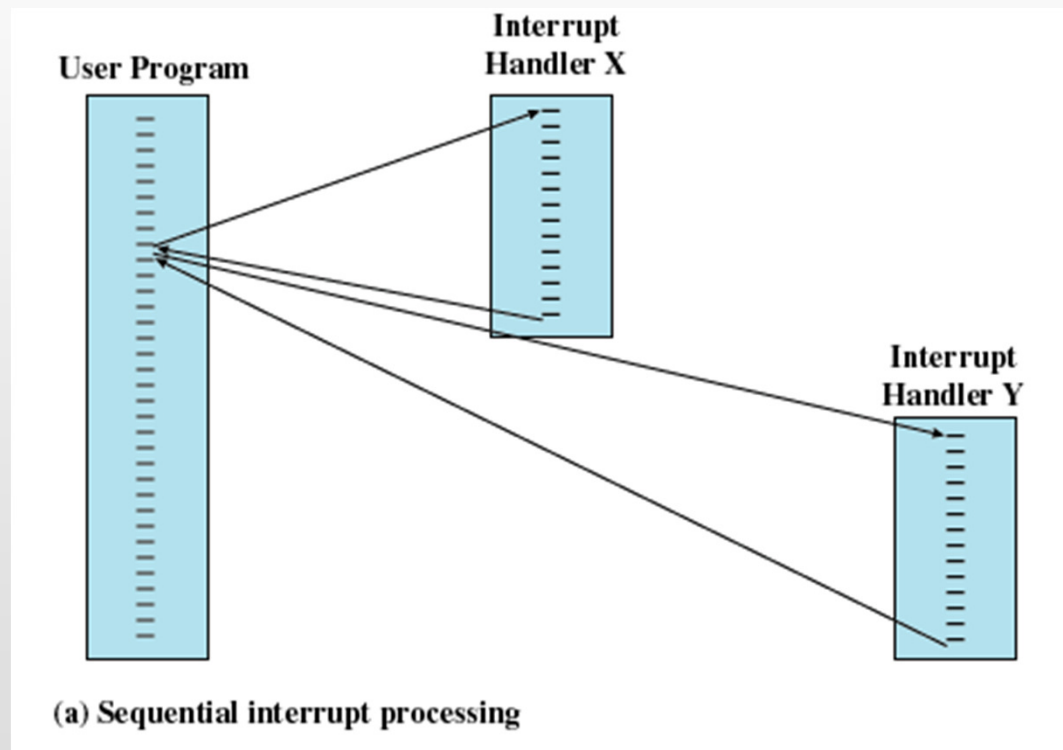


Figure 1.10 Simple Interrupt Processing



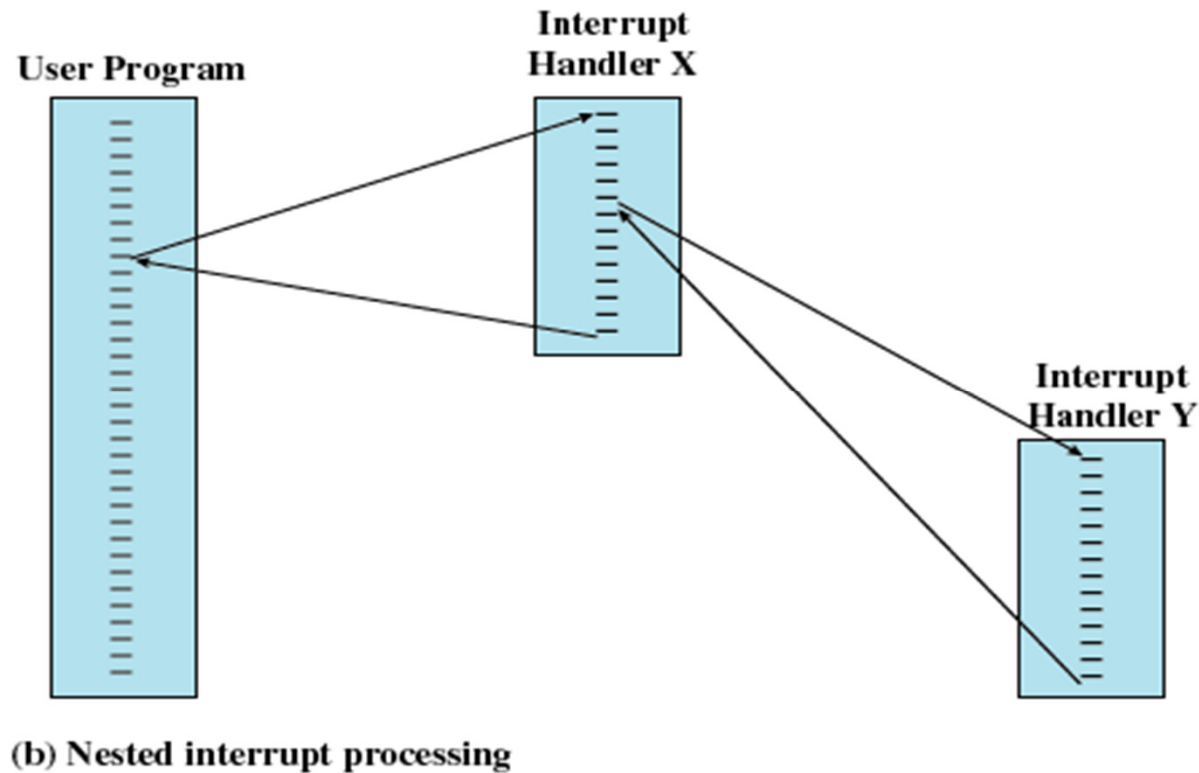
# Multiples Interrupciones

- ✓ Deshabilitar las interrupciones mientras una interrupción está siendo procesada.



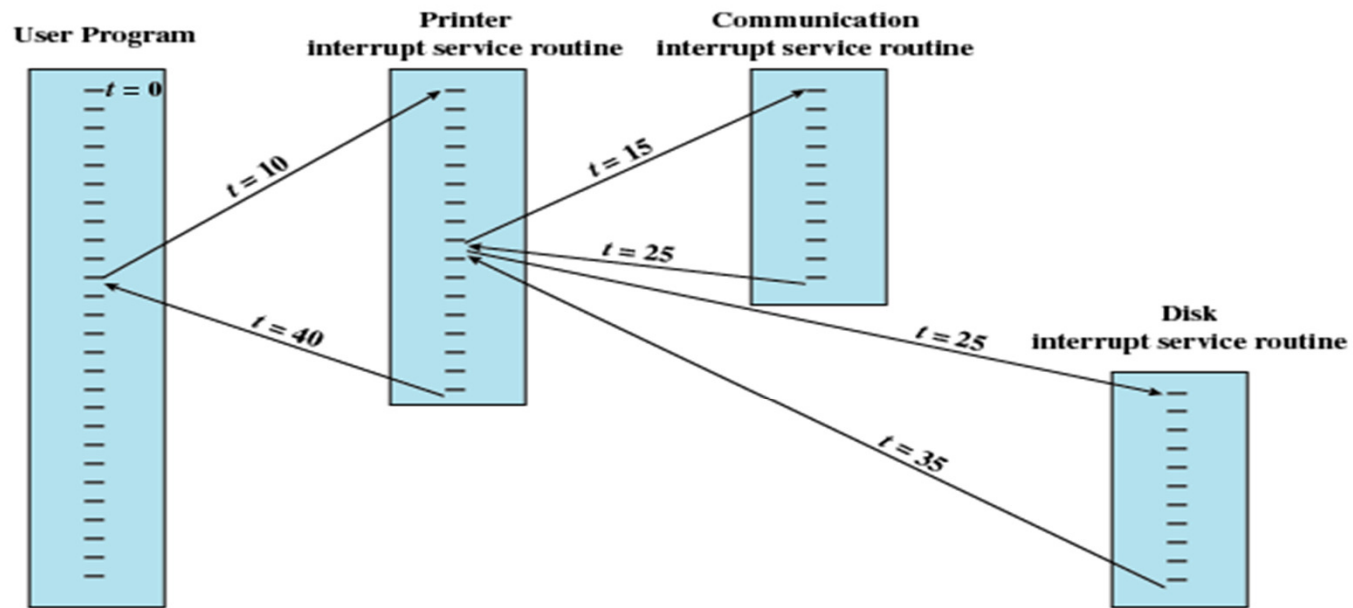
# Multiples Interrupciones

✓ Definir prioridades a las interrupciones





# Multiples Interrupciones



**Figure 1.13 Example Time Sequence of Multiple Interrupts**

