

# *Introducción a los Sistemas Operativos*

## *Anexo – Arquitectura de Computadoras*



- ✓ Versión: Agosto 2024
- ✓ 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) y y Conceptos de sistemas operativos (Silberschatz, Galvin, Gagne)



# Objetivo de la presentación

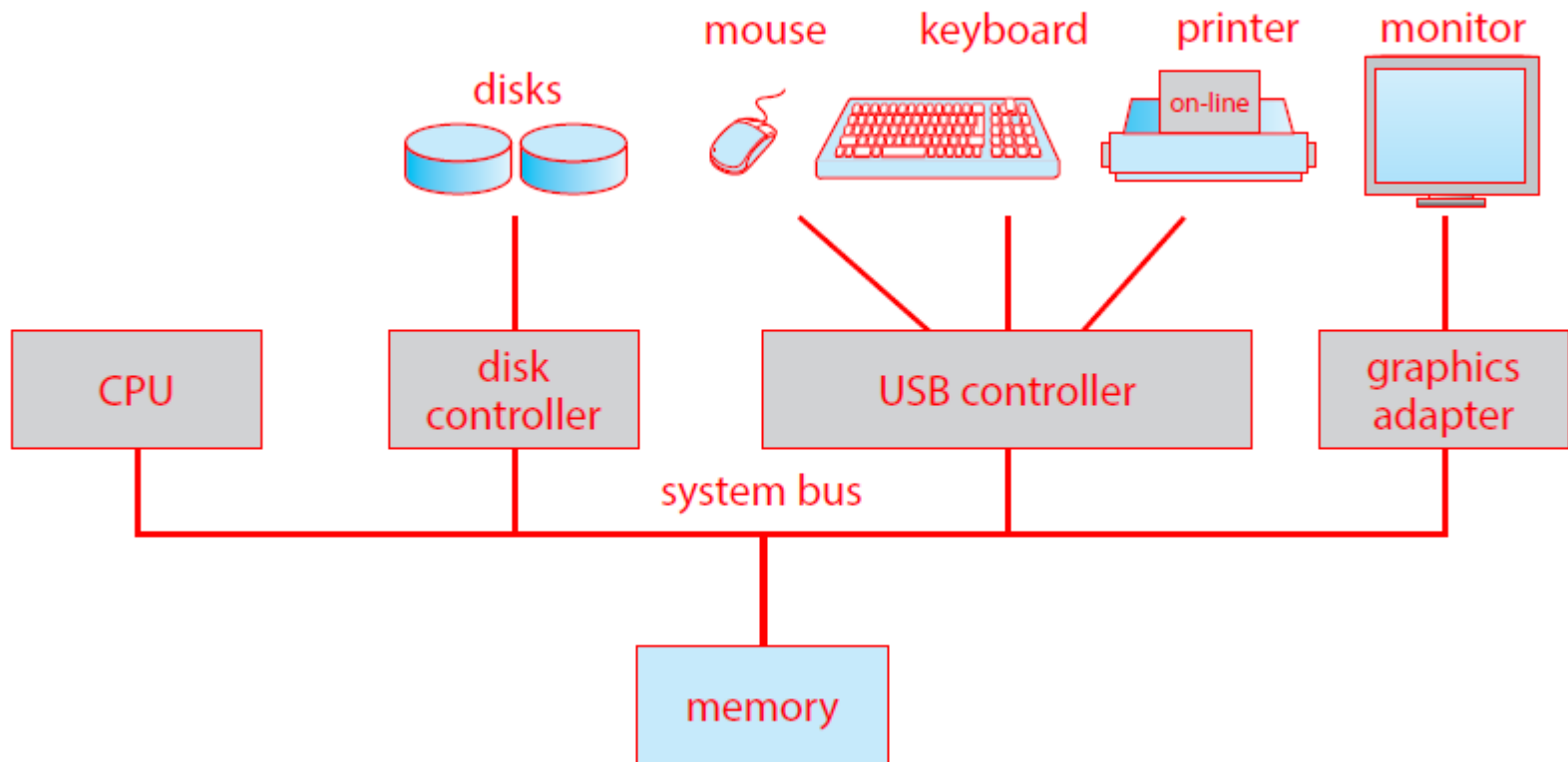
- ☑ Incluye los conceptos vistos en asignatura/s previa/s y que son necesarios tener presentes para conceptos que se verán durante la cursada.



# *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





Fuente: Operating System Concepts. Silberschatz, Galvin, Gagne



# Componentes de alto nivel

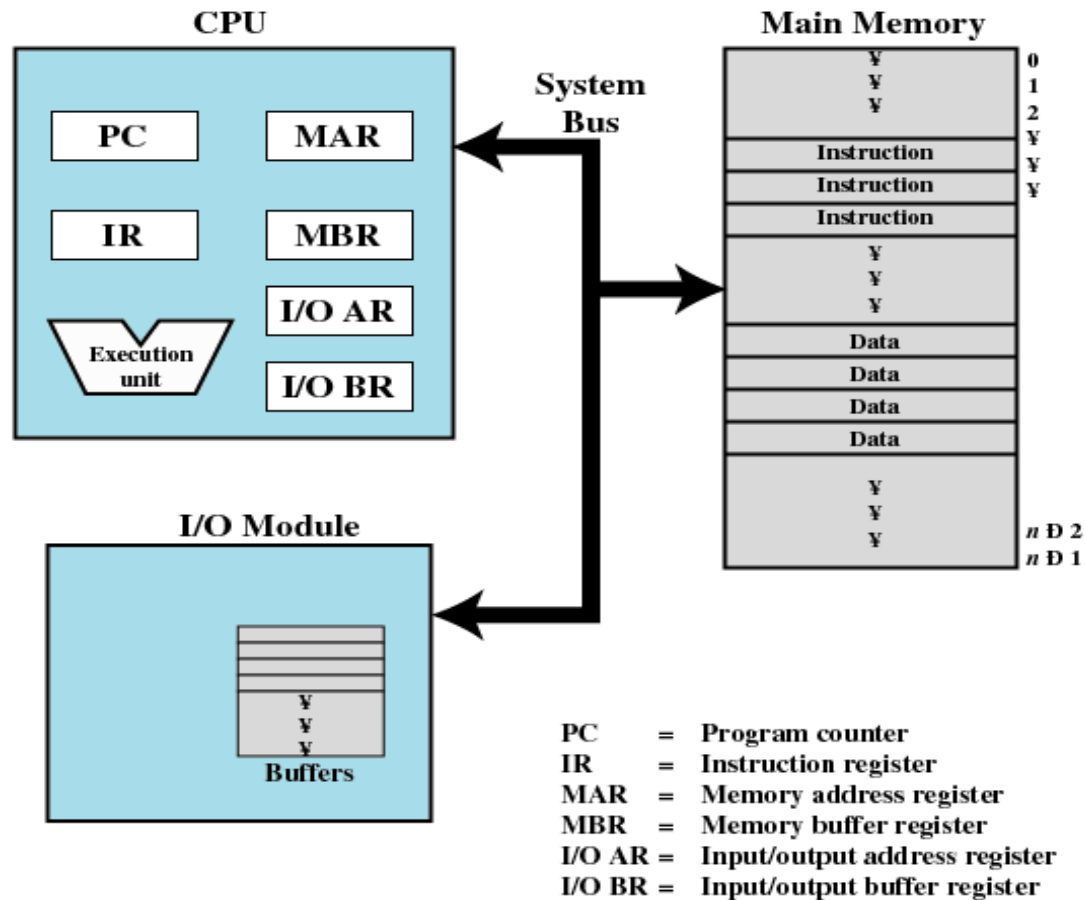


Figure 1.1 Computer Components: Top-Level View



# 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 proxima 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)



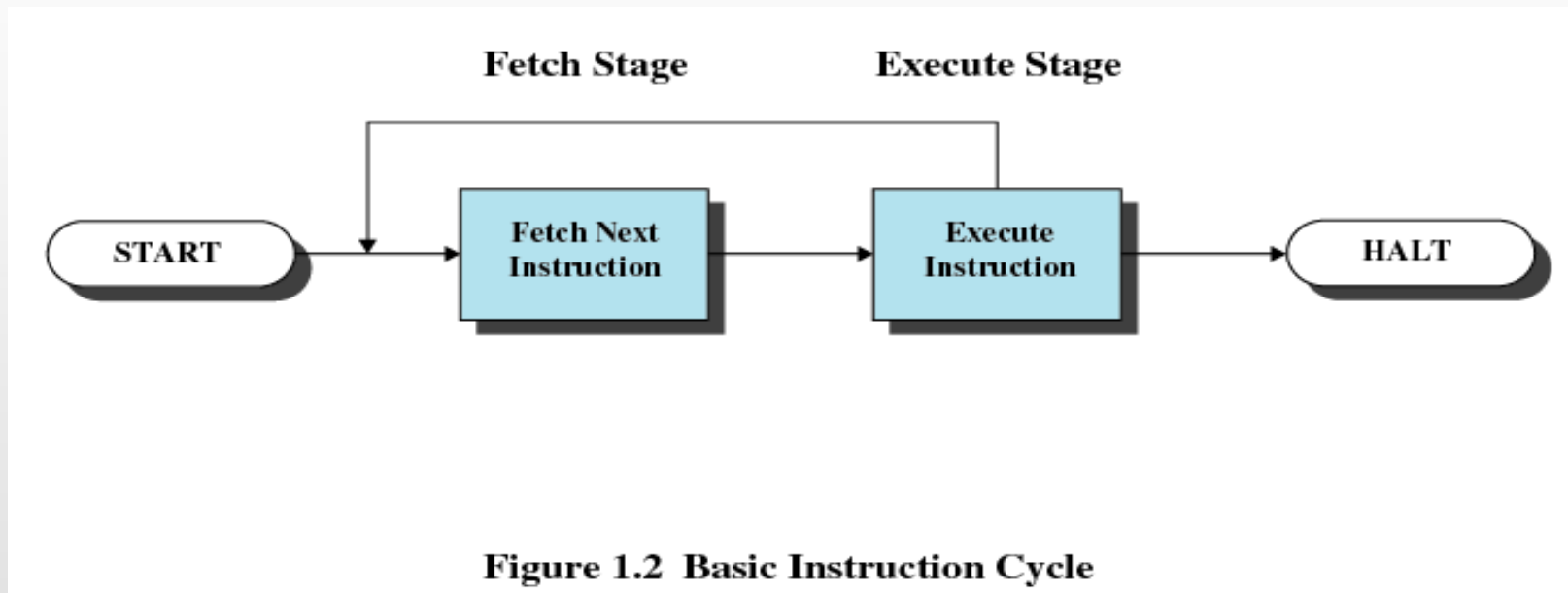
# *Ciclo Ejecución de Instrucción*

## ☑ 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) → IR
- ✓ El PC se incrementa después de cada fetch para apuntar a la próxima instrucción
  - $PC = PC + 4$



# IR - Instruction Register

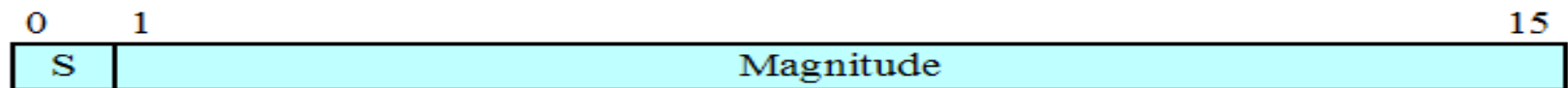
- ☑ La instrucción referenciada por el PC se almacena en el IR y se ejecuta
- ☑ Categorías de instrucciones
  - ✓ Procesador - Memoria
    - ◆ Transfiere datos entre procesador y memoria
  - ✓ Procesador - E/S
    - ◆ Transfiere datos a/o desde periféricos
  - ✓ Procesamiento de Datos
    - ◆ Operaciones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

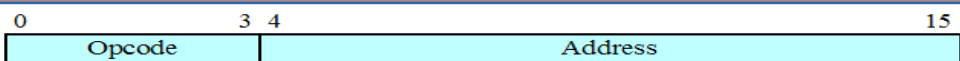
0001 = Load AC from Memory  
0010 = Store AC to Memory  
0101 = Add to AC from Memory

**(d) Partial list of opcodes**

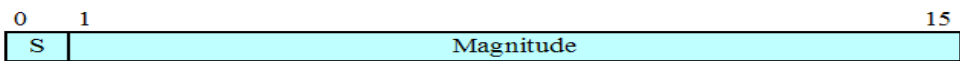
**Figure 1.3 Characteristics of a Hypothetical Machine**



# 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 instructions

Figure 1.3 Characteristics of a

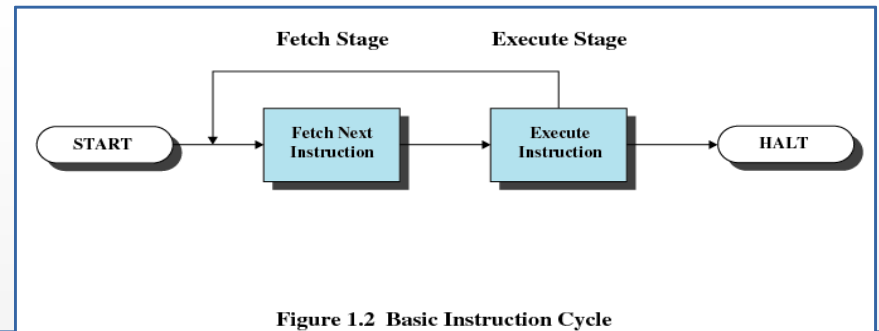
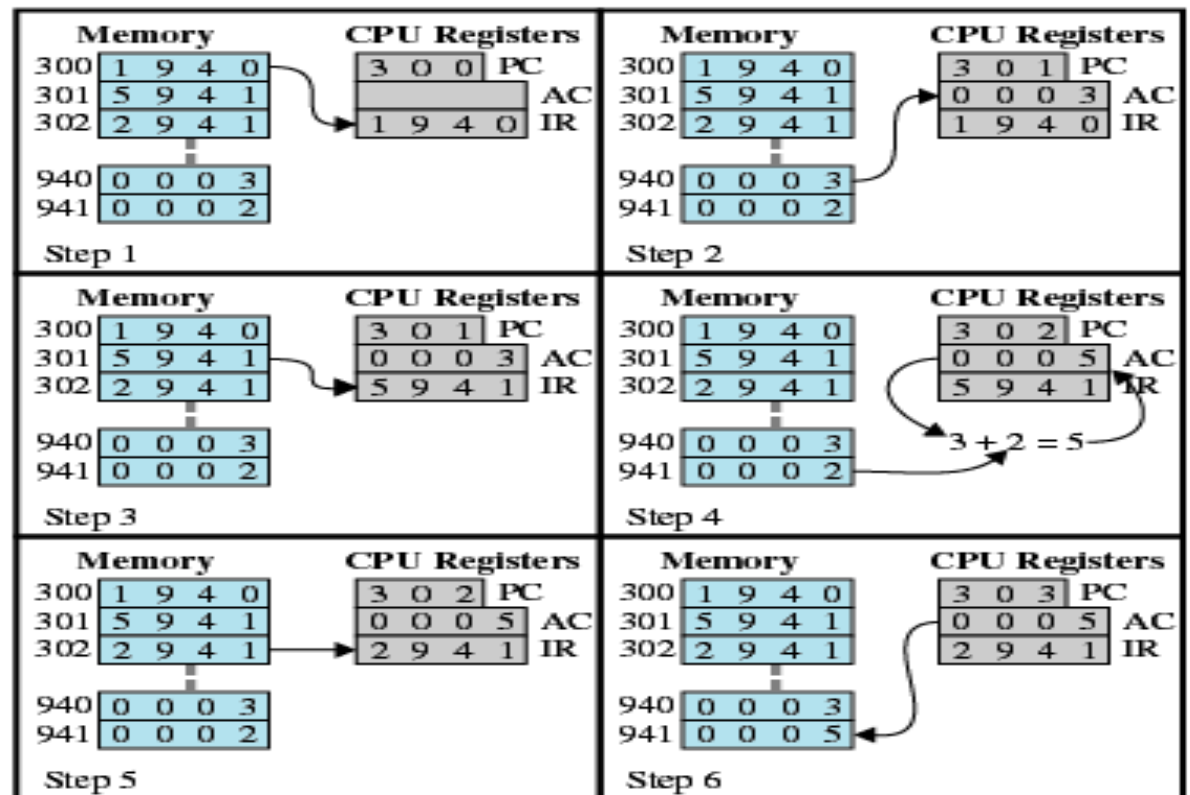


Figure 1.2 Basic Instruction Cycle



# 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





# Necesidades del SO

- ✓ Postergar el manejo de una interrupción en momentos críticos
- ✓ Atender en forma eficiente: la rutina de atención adecuada según el dispositivo
- ✓ Tener varios niveles de interrupción para que el SO pueda distinguir entre interrupciones de alta prioridad y de baja prioridad y responder adecuadamente



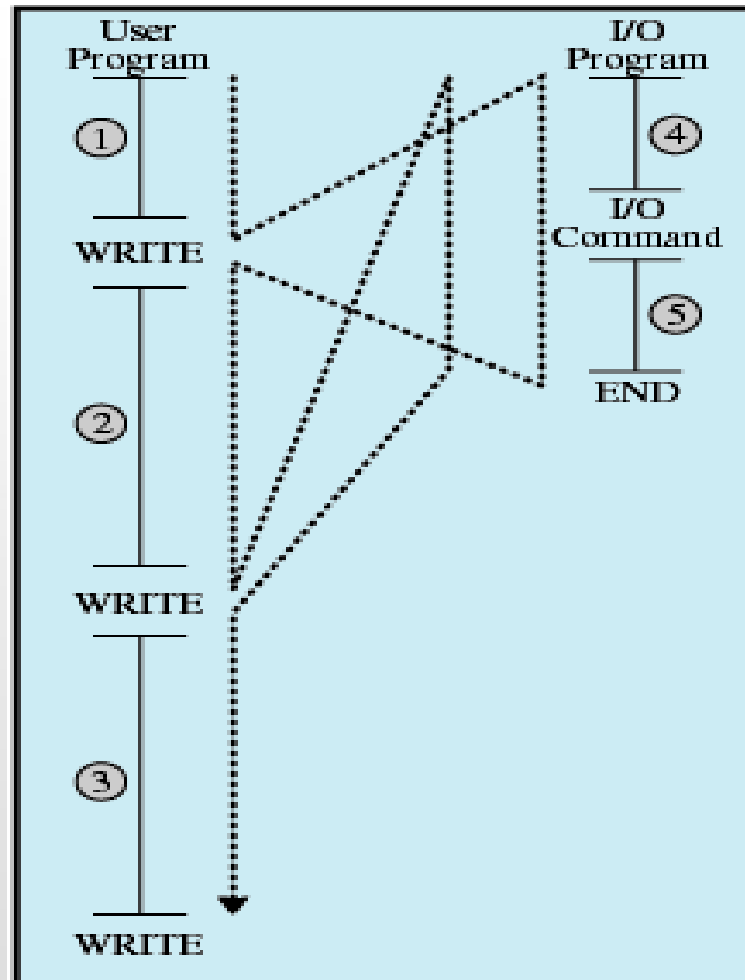
# Clases de Interrupciones

**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.



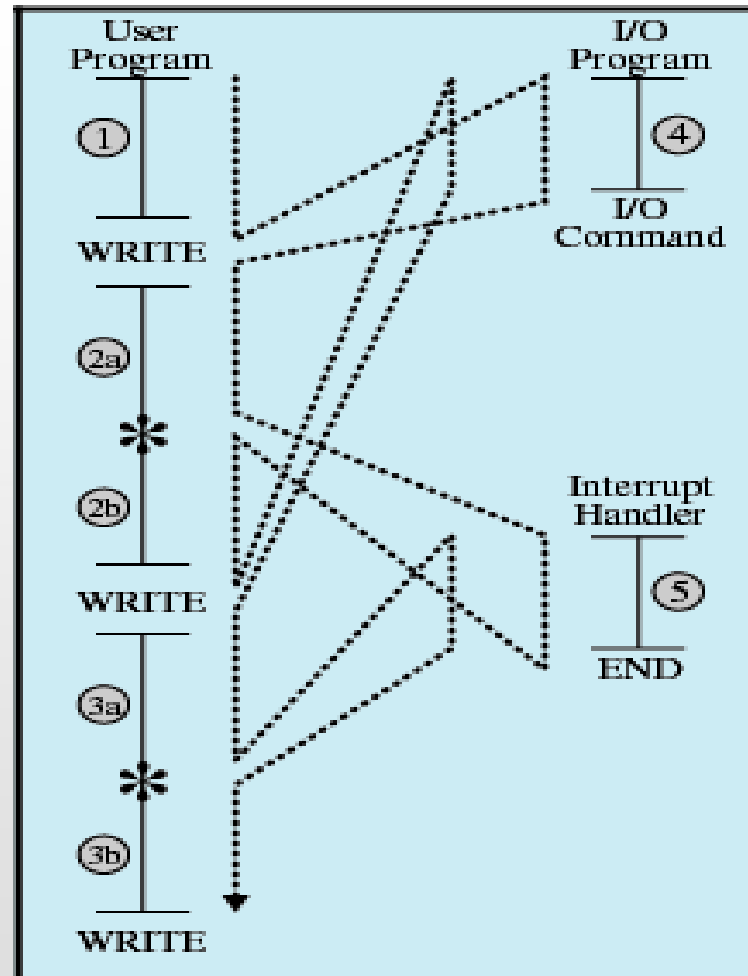
# Flujo de control SIN interrupciones



(a) No interrupts



# Flujo de control CON interrupciones



(b) Interrupts; short I/O wait



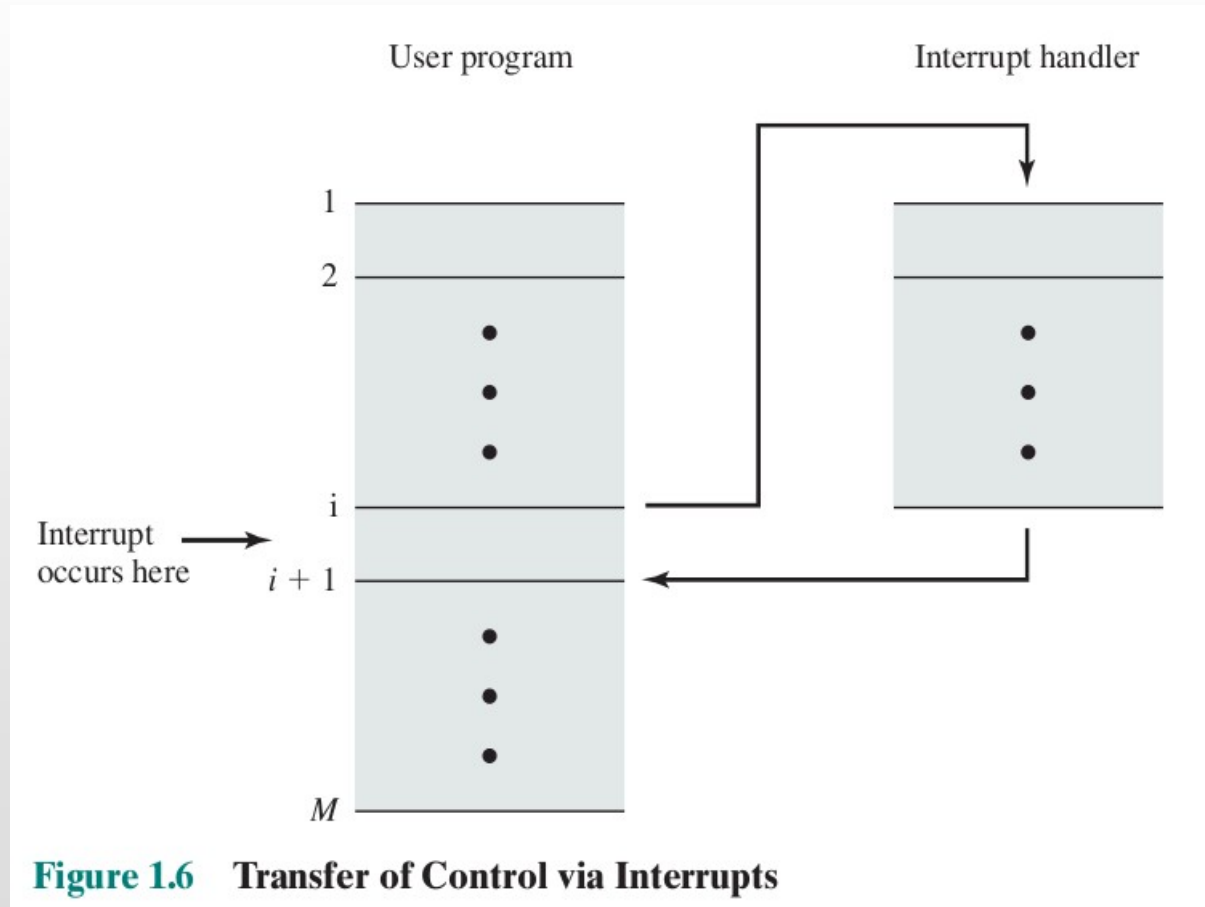
# *Interrupt Handler*

- ✓ 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



# Ciclo de interrupción

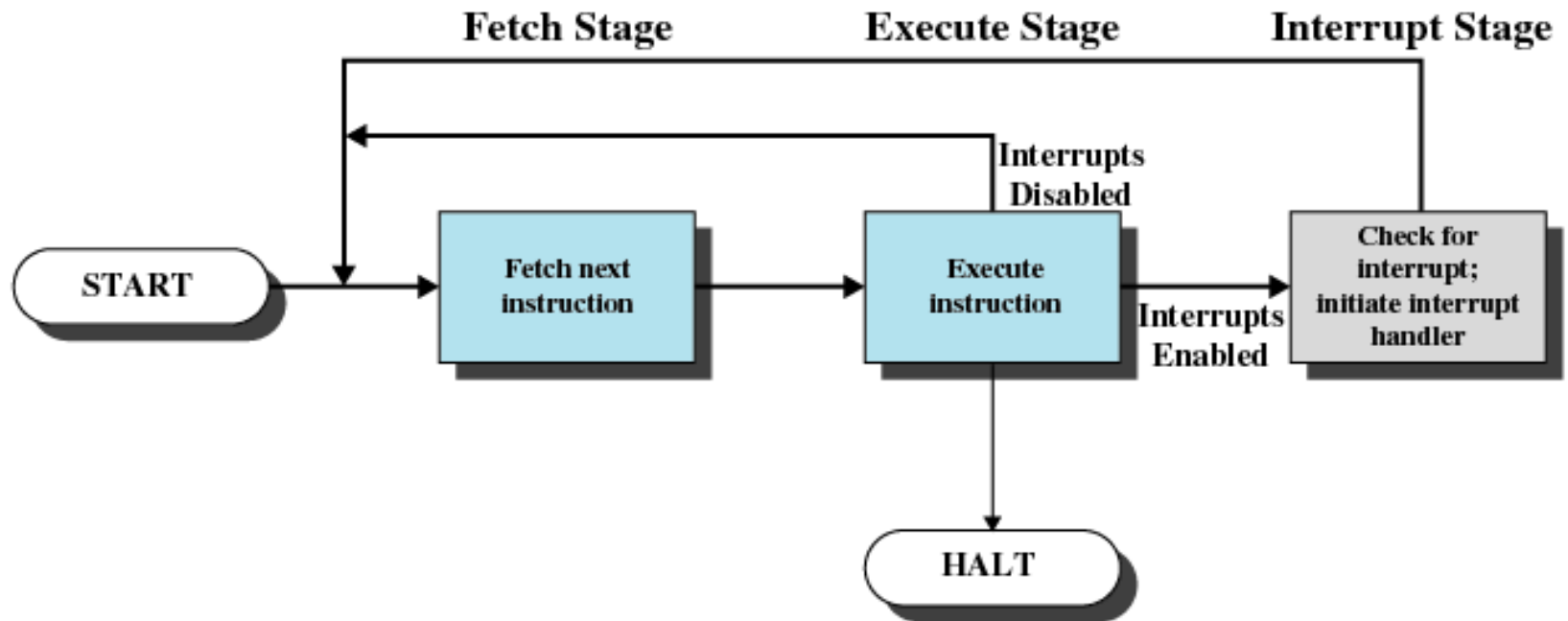


Figure 1.7 Instruction Cycle with Interrupts



# *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.





# Procesamiento simple de una interrupción

IMPORTANTE!!!

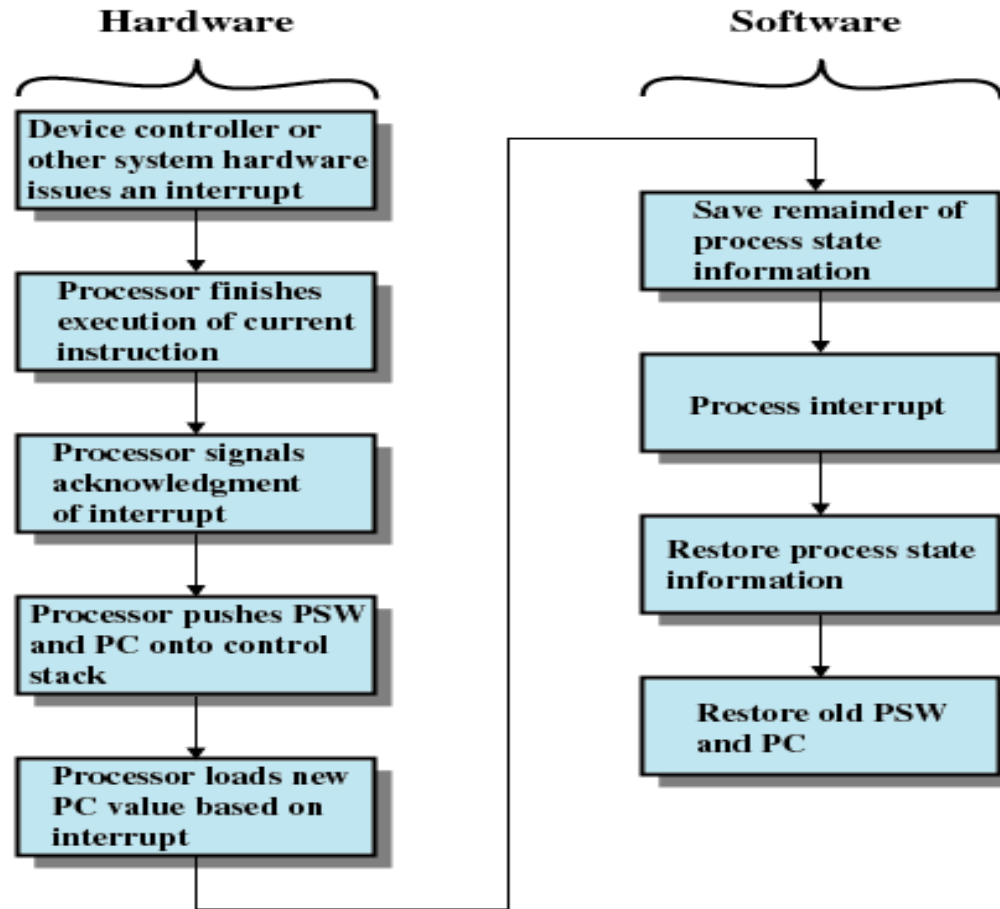
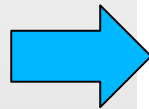


Figure 1.10 Simple Interrupt Processing



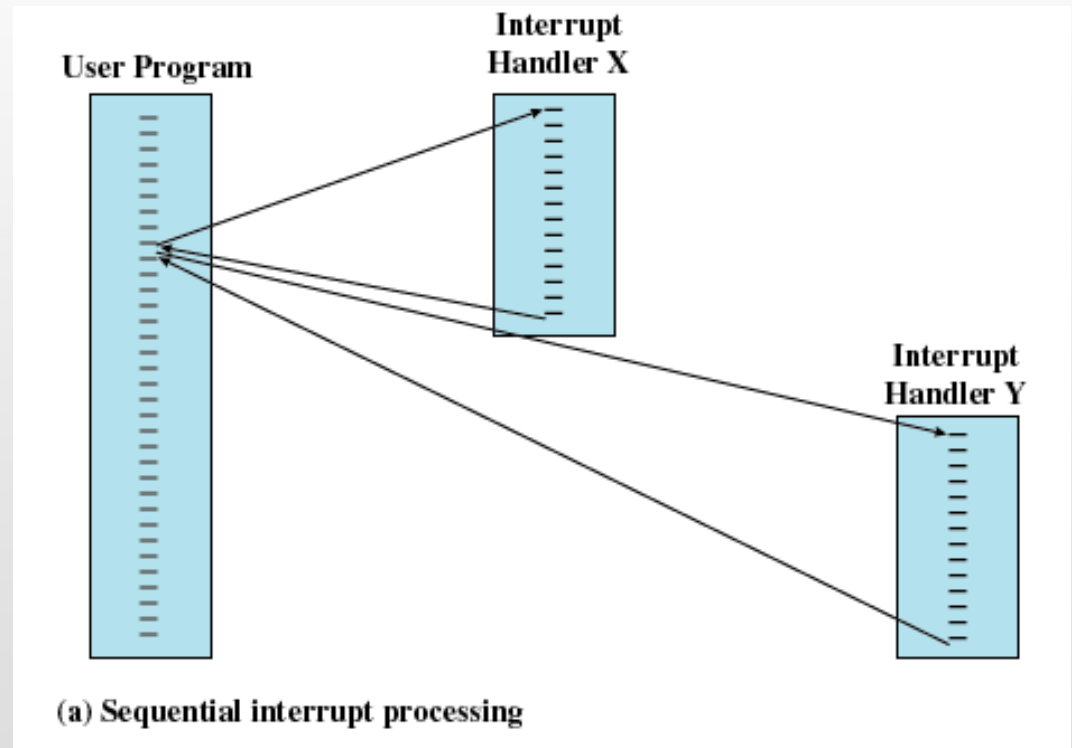
# *Interrupciones no enmascarables y enmascarables*

- ✓ La mayoría de las CPUs tienen dos líneas de requerimiento de interrupciones: la de no enmascarables y las enmascarables.
- ✓ La de no enmascarables se reservan para eventos tales como errores de memoria no recuperables.
- ✓ La de enmascarables puede ser “apagada” por la CPU si en ese momento se está ejecutando una secuencia crítica de instrucciones. Estas son las que usan los controladores de dispositivo cuando necesitan servicio.



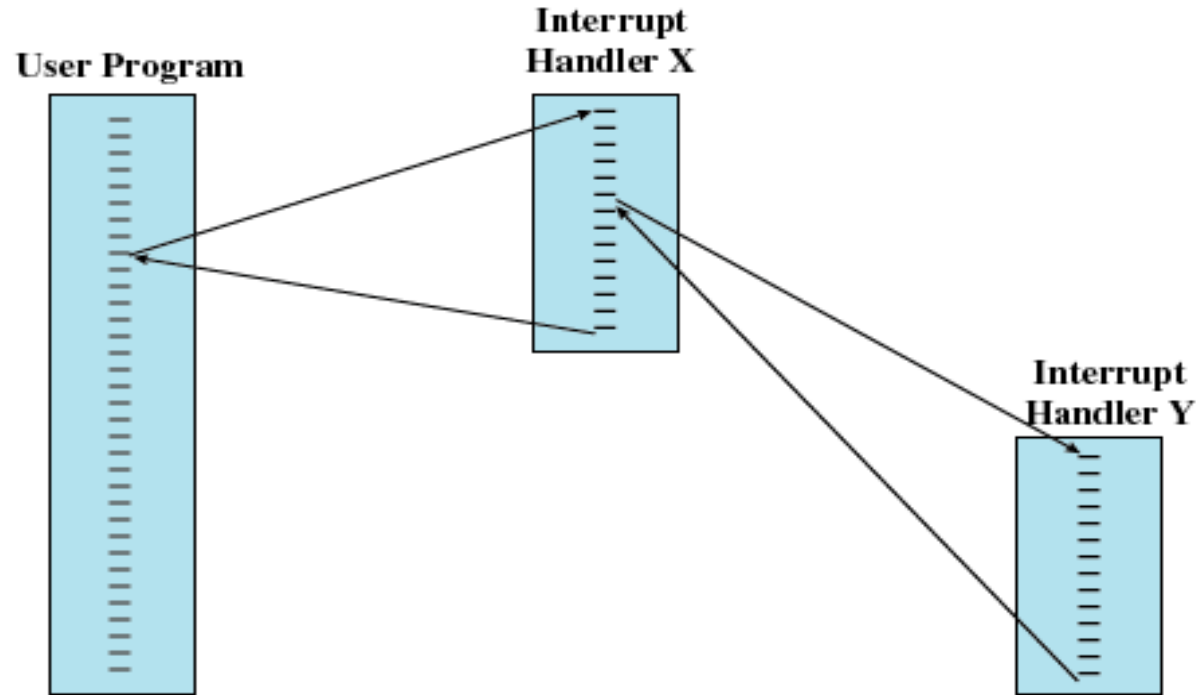
# Multiples Interrupciones

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



# Multiples Interrupciones

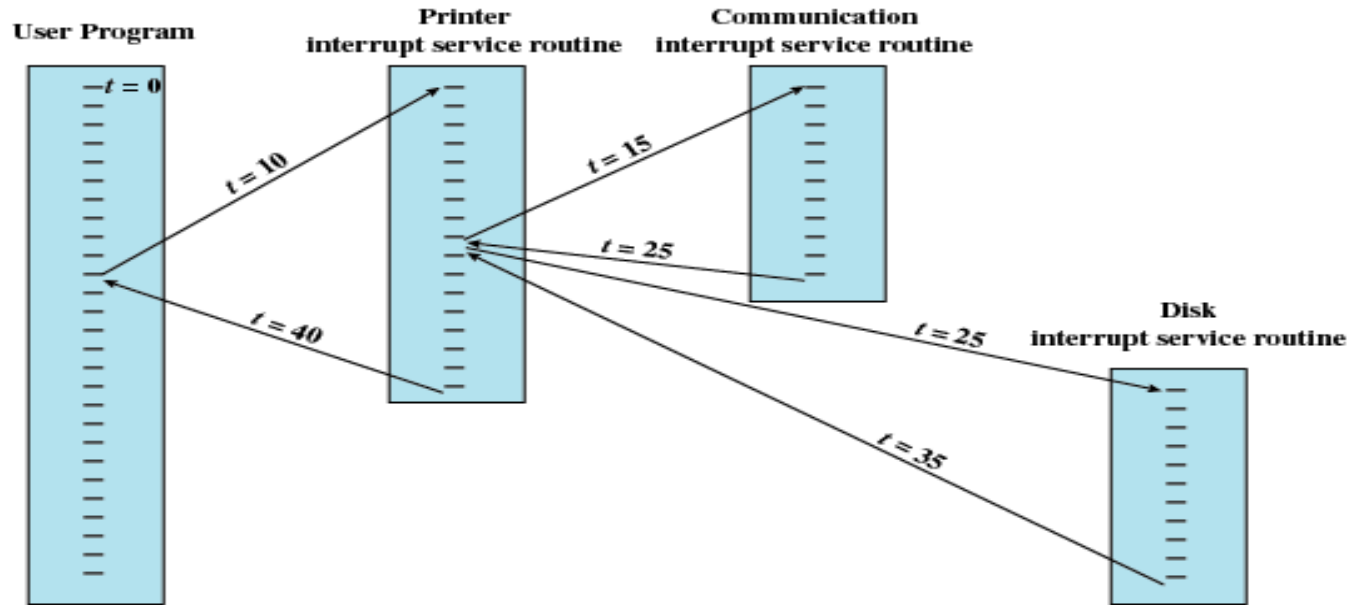
☑ Definir prioridades a las interrupciones



(b) Nested interrupt processing



# Multiples Interrupciones



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



# Descripción de número en el vector de interrupciones de procesador Intel:

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

- De 1 a 31: **no enmascarables**, por ejemplo, errores de condición
- 32 a 255, **enmascarables**, usadas para interrupciones generadas por dispositivos

