

# *Introducción a los Sistemas Operativos*

## **Introducción – I**

**Profesores:**

Lía Molinari

Juan Pablo Pérez

Macia Nicolás



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

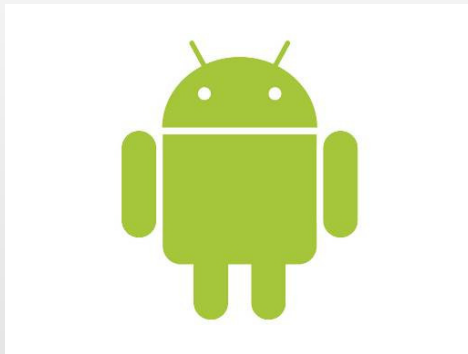
# I.S.O.

- ✓ Versión: Marzo 2013
- ✓ Palabras Claves: Sistemas Operativos, Hardware, Interrupciones, Registros

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts)



# *¿Qué es un Sistema Operativo?*



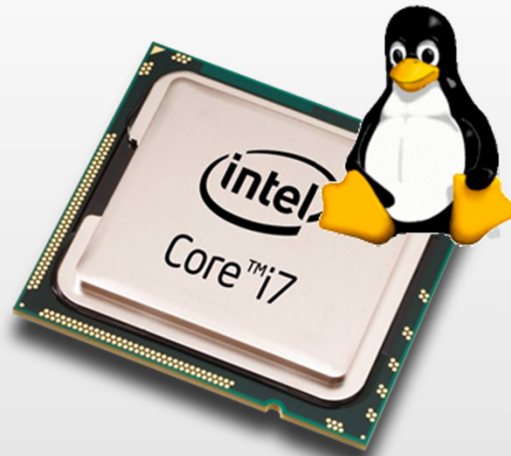
¿SO?



# Sistema Operativo

☑ Es software:

✓ necesita procesador y memoria para ejecutarse



☑ Dos perspectivas

✓ de arriba hacia abajo

✓ de abajo hacia arriba



# *Perspectiva de arriba hacia abajo*

- ✓ Abstracción con respecto a la arquitectura
- ✓ El SO “oculta” el HW y presenta a los programas abstracciones más simples de manejar.
- ✓ Arquitectura: conjunto de instrucciones, organización de memoria, E/S, estructura de bus)
- ✓ Los programas de aplicación son los “clientes” del SO.
- ✓ Comparación: uso de escritorio y uso de comandos de texto
- ✓ Comodidad, “amigabilidad” (friendliness)



# *Perspectiva de abajo hacia arriba*

- ✓ Visión del SO como un administrador de recursos
- ✓ Administra los recursos de HW de uno o más procesadores
- ✓ Provee un conjunto de servicios a los usuarios del sistema
- ✓ Maneja la memoria secundaria y dispositivos de I/O.
- ✓ Ejecución simultánea de programas
- ✓ Multiplexación en tiempo (CPU) y en espacio (memoria)



# *Elementos Básicos de una computadora*

- ☑ Procesador
- ☑ Memoria Principal
  - ✓ Volátil
  - ✓ Se refiere como memoria real o primaria
- ☑ Componentes de I/O
  - ✓ Dispositivos de memoria secundaria
  - ✓ Equipamiento de comunicación
  - ✓ terminales
- ☑ Bus Sistema
  - ✓ comunicación entre procesadores, memoria, dispositivos de I/O



# Componentes de alto nivel

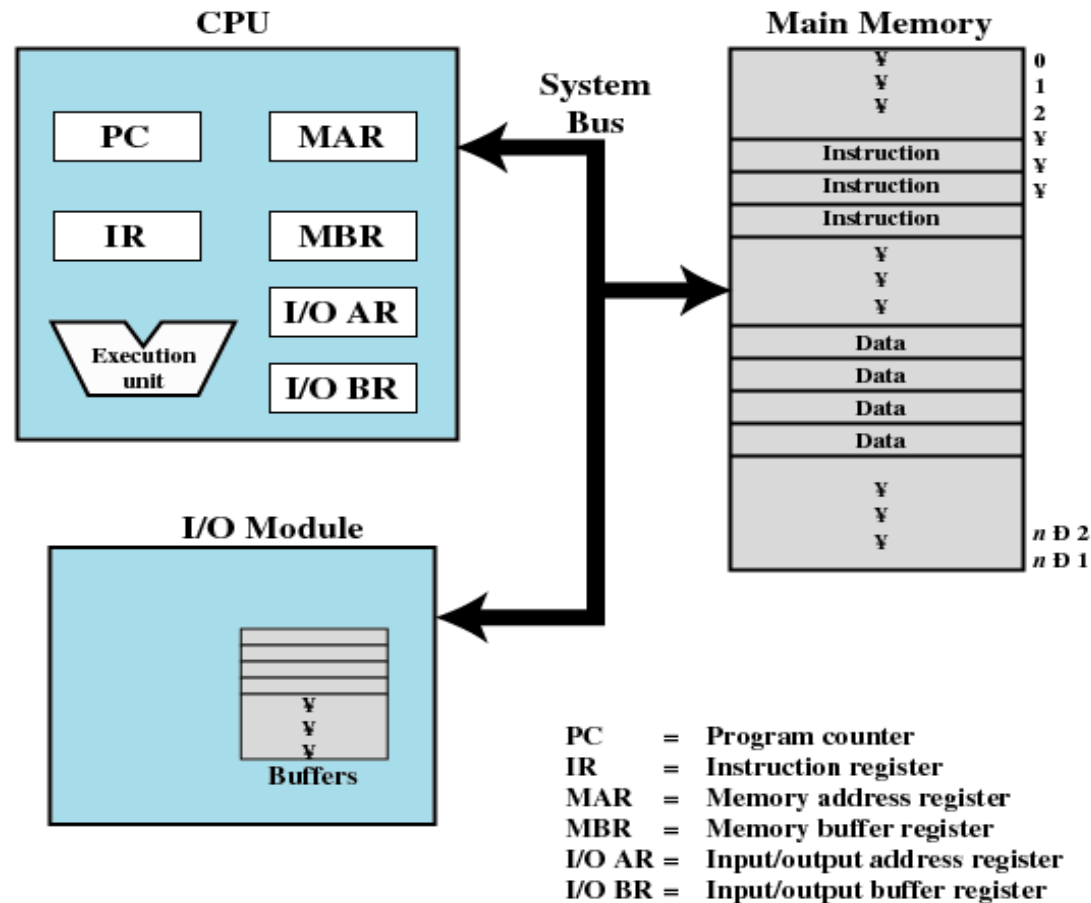


Figure 1.1 Computer Components: Top-Level View





# 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 programas



# *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/user)



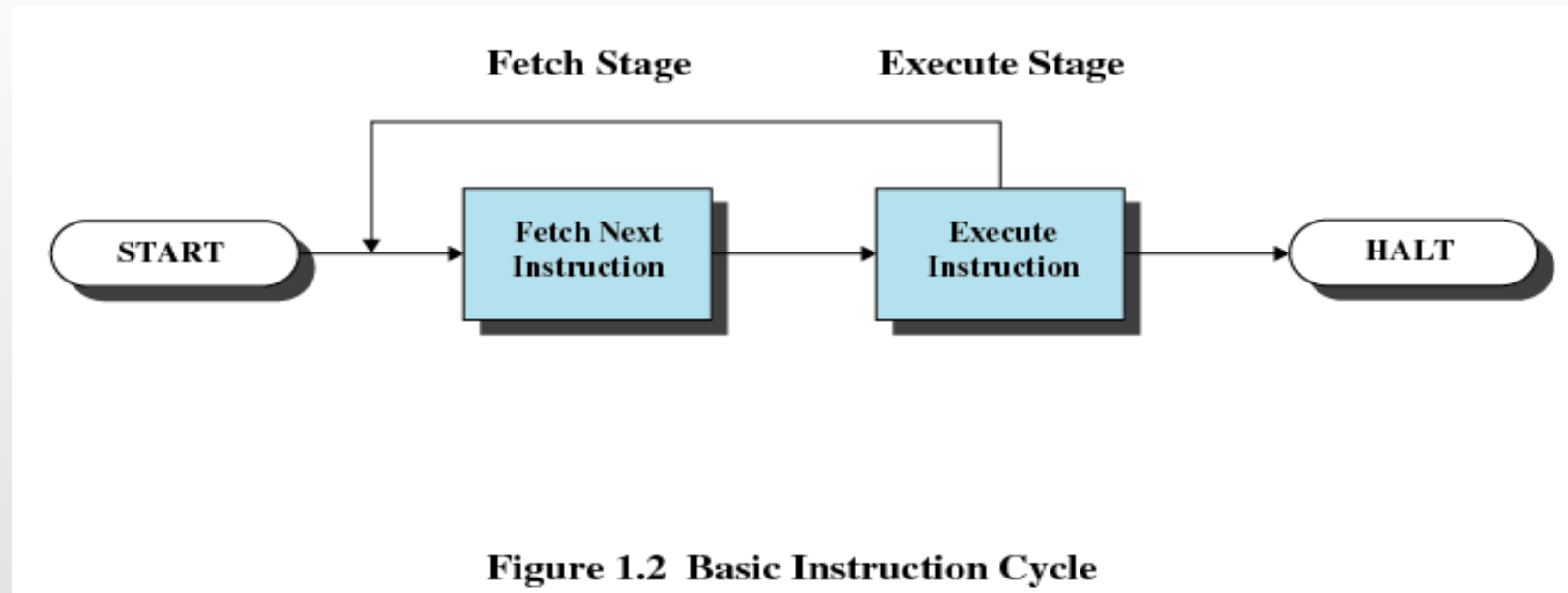
# *Ejecución de Instrucción*

## ☑ Dos pasos

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



# Ciclo Instrucción



# *Instrucción: Fetch y Execute*

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

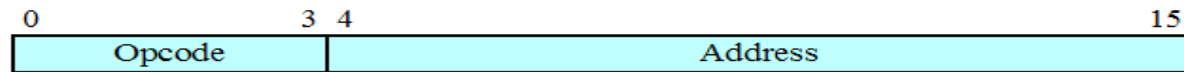


# *Instruction Register*

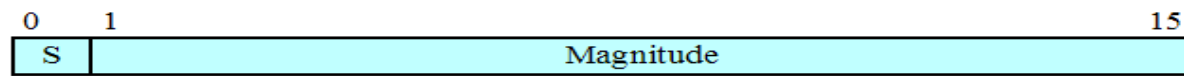
- ☑ La instrucción referenciada por el PC se almacena en el IR y se ejecuta
- ☑ Categorías de instrucciones
  - ✓ Procesador-memoria
    - ◆ Transfiere datos entre procesador y memoria
  - ✓ Procesador-I/O
    - ◆ Transfiere datos a/o desde periféricos
  - ✓ Procesamiento de Datos
    - ◆ Operaciones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

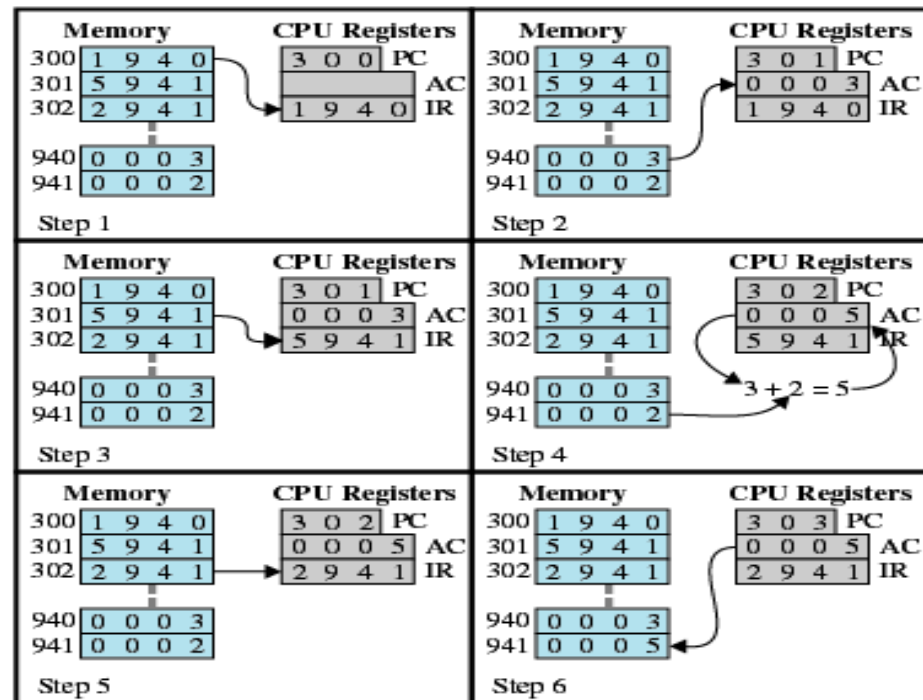
**(d) Partial list of opcodes**

**Figure 1.3 Characteristics of a Hypothetical Machine**





# Ejemplo de una ejecución de programa



**Figure 1.4 Example of Program Execution**  
(contents of memory and registers in hexadecimal)



# Interrupciones

- ☑ Interrumpen el secuenciamiento del procesador en la ejecución de un proceso
- ☑ Dispositivos de I/O más lentos que el procesador
  - ✓ Procesador debe esperar al dispositivo



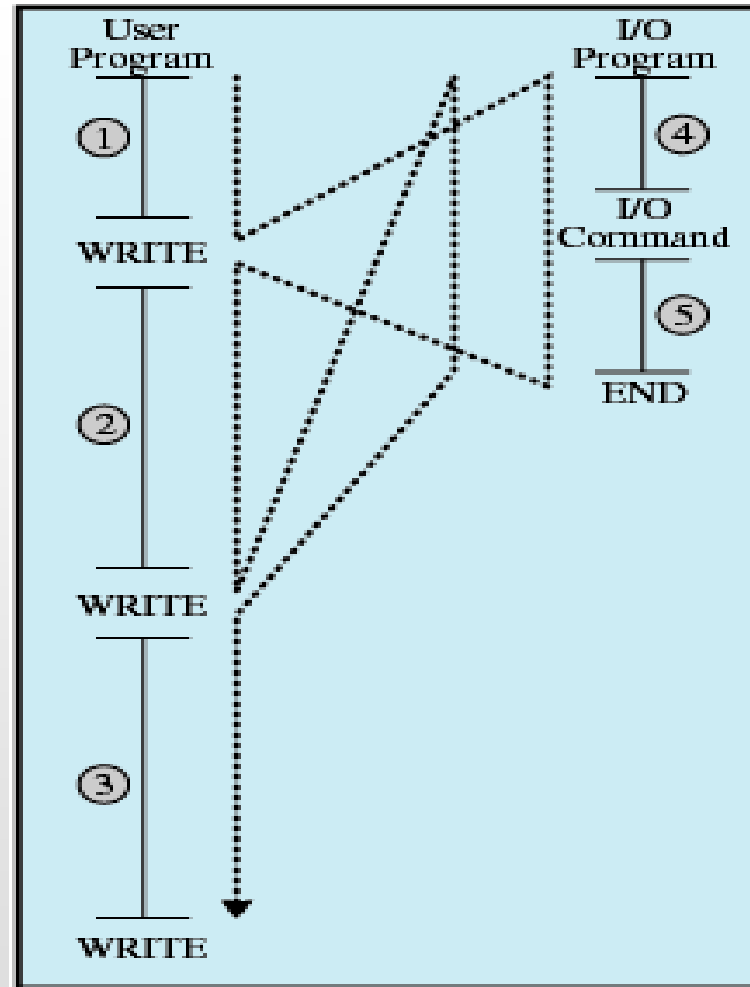
# Clases de Interrupciones

**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.



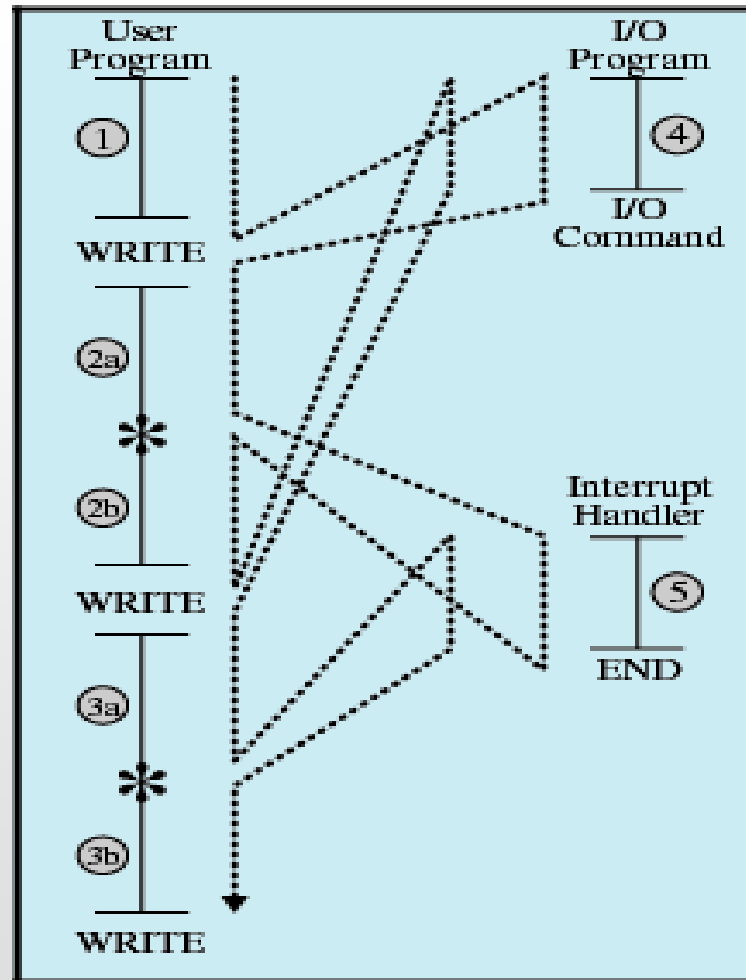
# Flujo de control SIN interrupciones



(a) No interrupts



# Flujo de control CON interrupciones



(b) Interrupts; short I/O wait



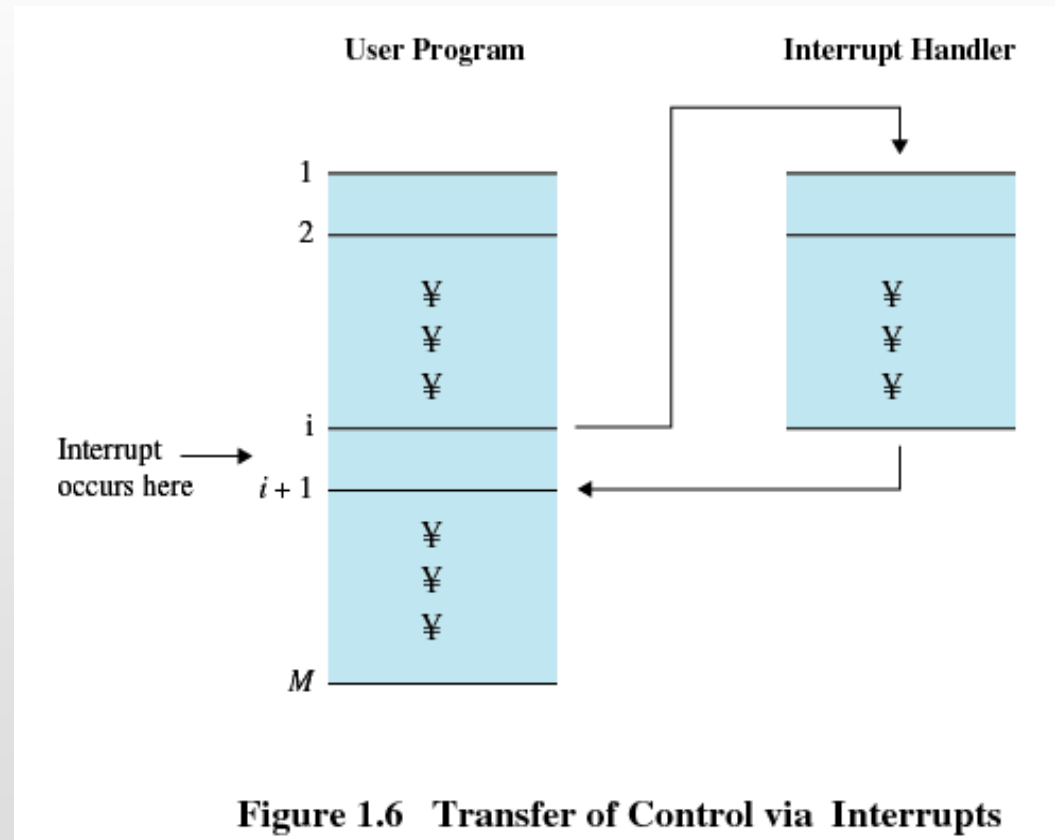
# *Interrupt Handler*

- ☑ Programa (o rutina) que atiende una determinada interrupción
  - ✓ Por ejemplo, para un dispositivo particular de I/O
- ☑ Generalmente es parte del SO



# Interrupciones

☑ Suspende la secuencia normal de ejecución



# Ciclo de interrupción

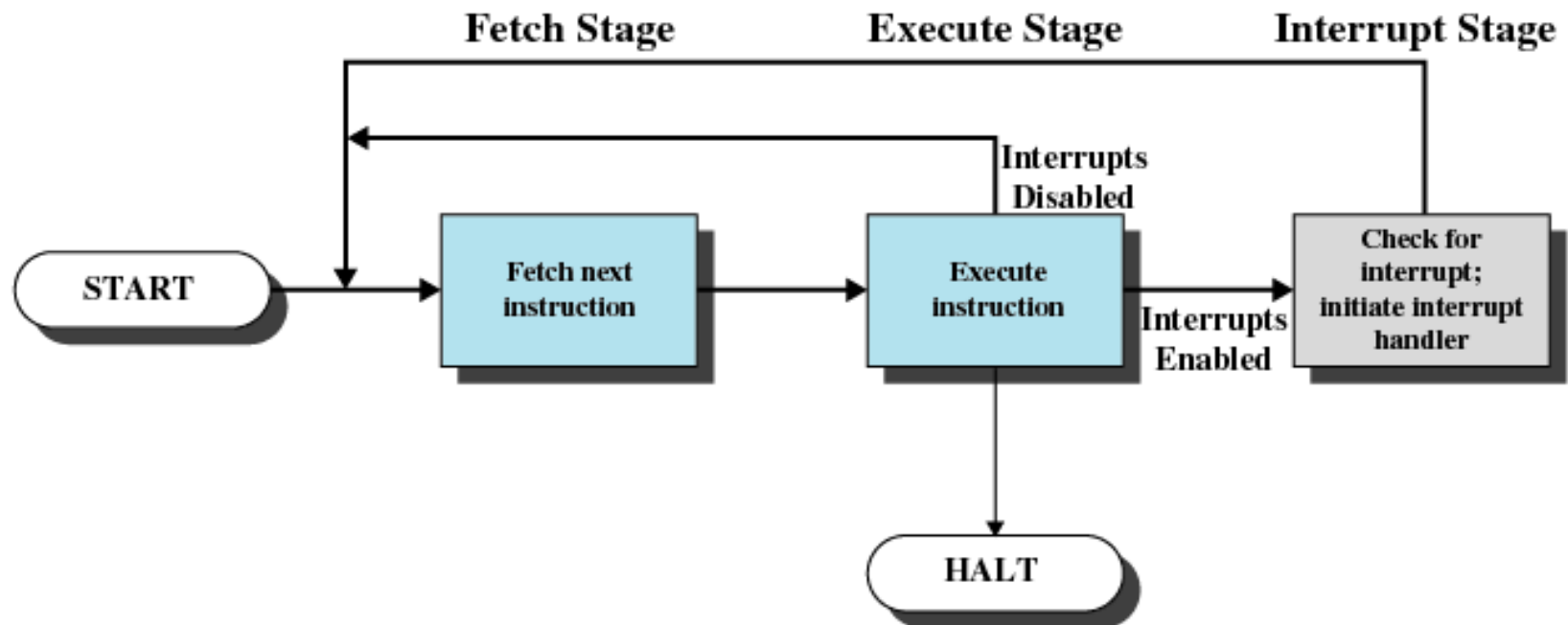


Figure 1.7 Instruction Cycle with Interrupts





# *Ciclo de interrupción*

- ✓ El procesador chequea la existencia de interrupciones.
- ✓ Si no existen interrupciones, la proxima 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

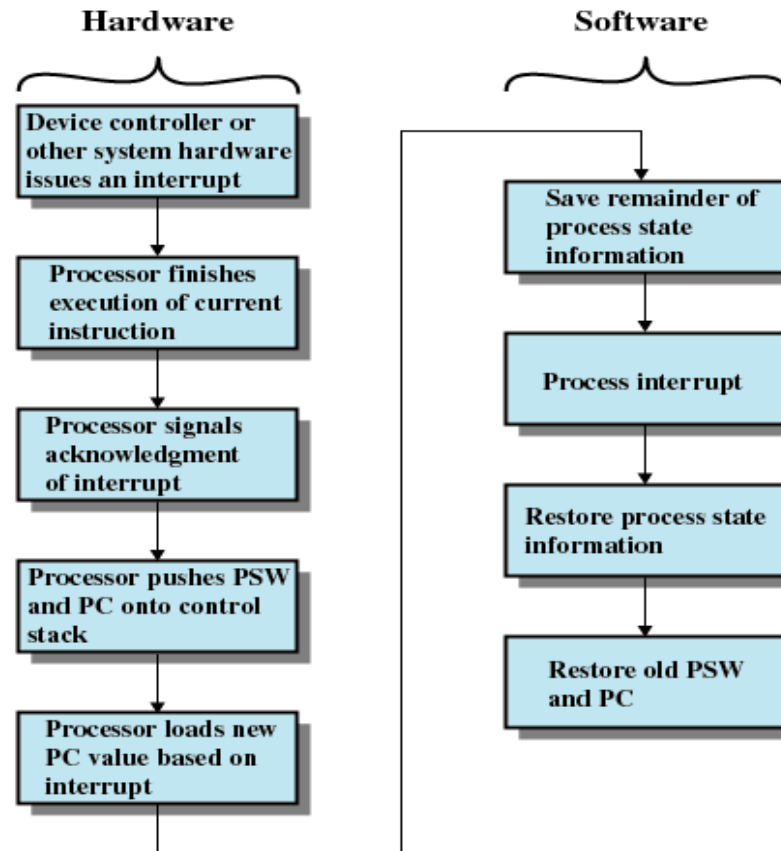
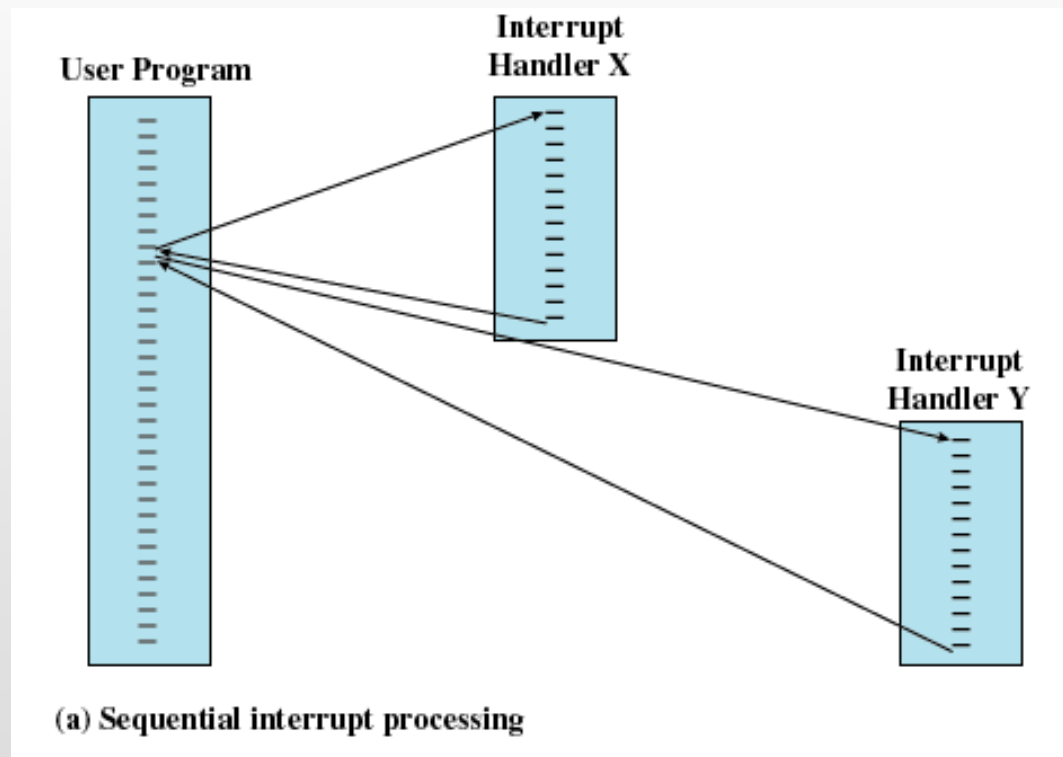


Figure 1.10 Simple Interrupt Processing



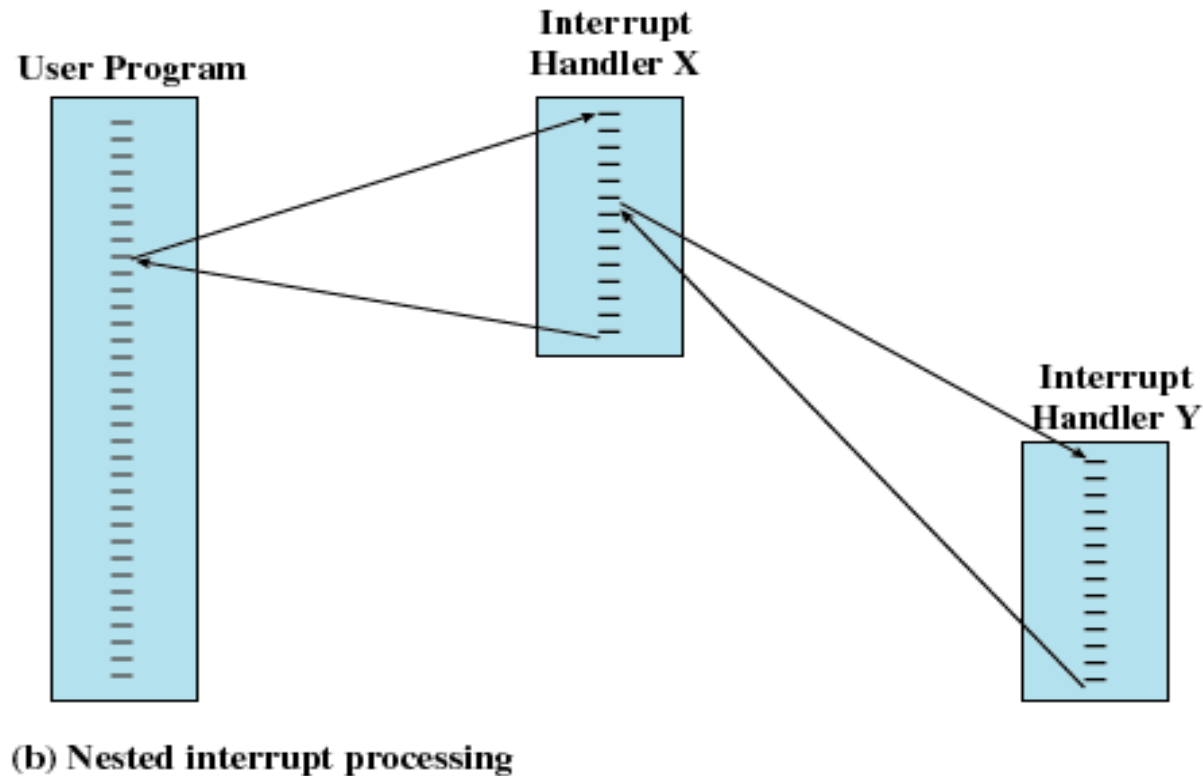
# Multiples Interrupciones

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

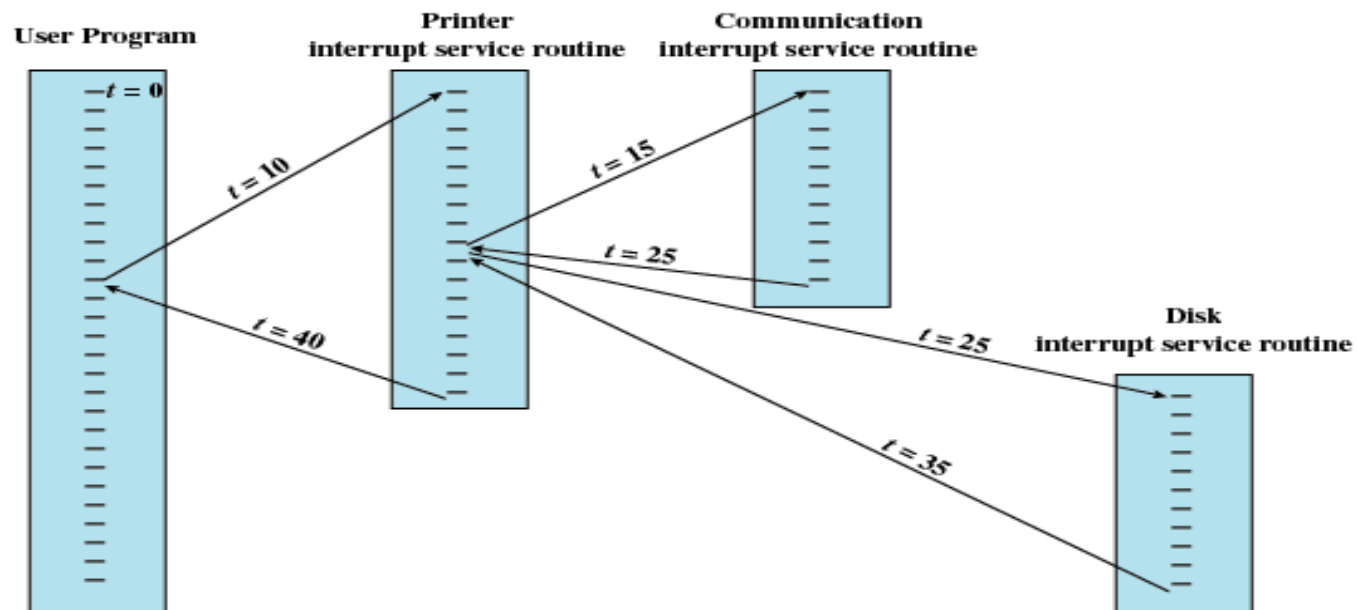


# Multiples Interrupciones

- ✓ Definir prioridades a las interrupciones



# Multiples Interrupciones



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



# *Direct Memory Access (DMA)*

- ☑ El intercambio de I/O ocurre directamente con la memoria
- ☑ El procesador le da autoridad al dispositivo de I/O para leer o escribir a memoria
- ☑ Releva al procesador de la responsabilidad del intercambio

