

Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informàtica de Sistemes y Computadoras (DISCA)
Universitat Politècnica de València

Bloque Temático 1: Introducción

Unidad Temática 2: Concepto de Llamada al Sistema

fSO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objetivos**

- Proporcionar una visión global del **funcionamiento del computador**, destacando los aspectos que afectan de forma directa al sistema operativo
- Presentar el concepto de **llamada al sistema**, como el mecanismo necesario para obtener los servicios del sistema operativo
- Describir los **servicios** que el sistema operativo proporciona a los usuarios y a los procesos

- **Bibliografía**

- A. Silberschatz, P. B. Galvin. “Sistemas Operativos”. 7ª ed. Capítulos. 1 y 2

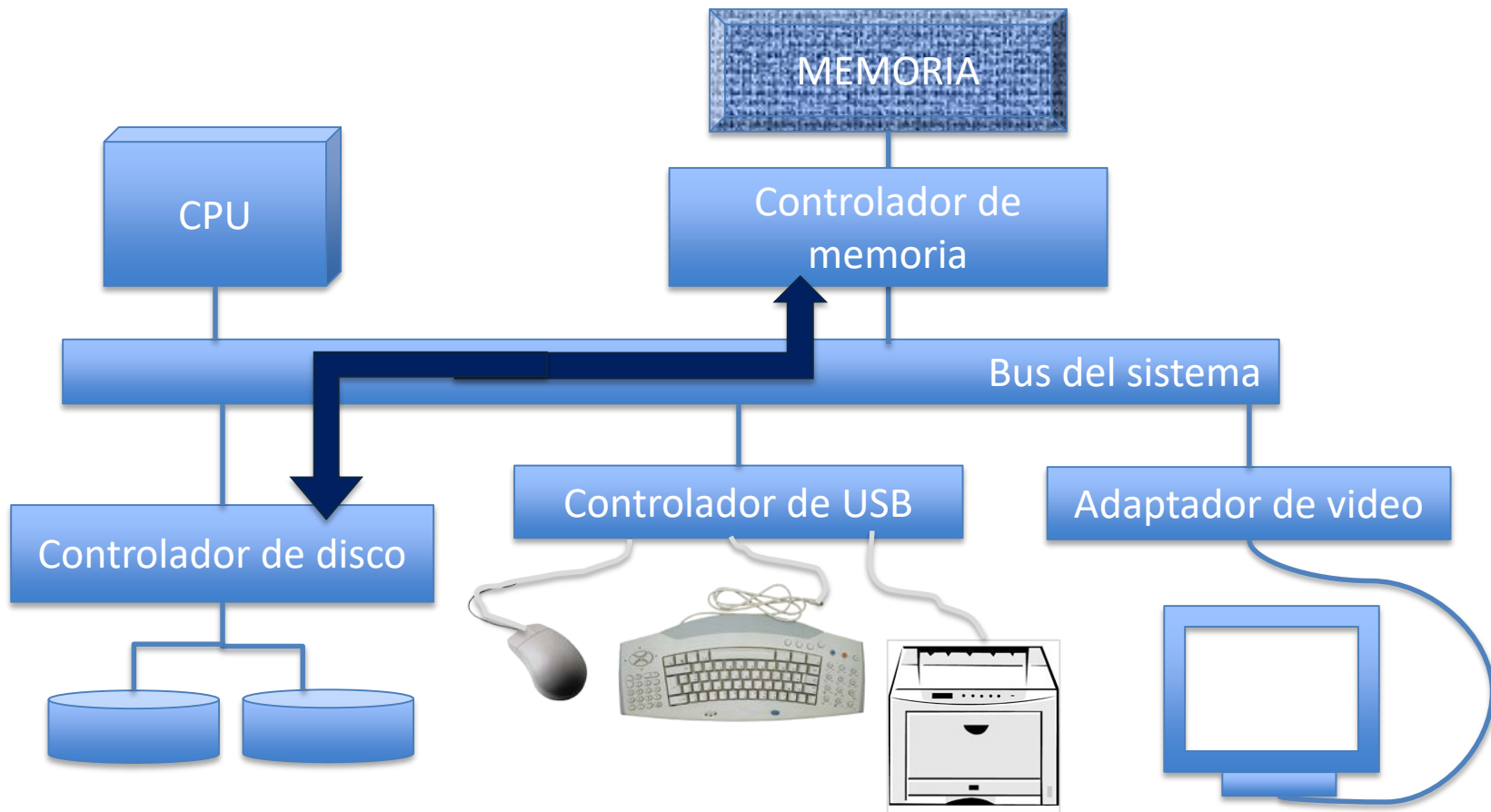
- **Arquitectura Hardware del Computador**
- Interrupciones
- Modos de Ejecución
- Llamadas al Sistema
- Utilidades del Sistema

Nomenclatura:

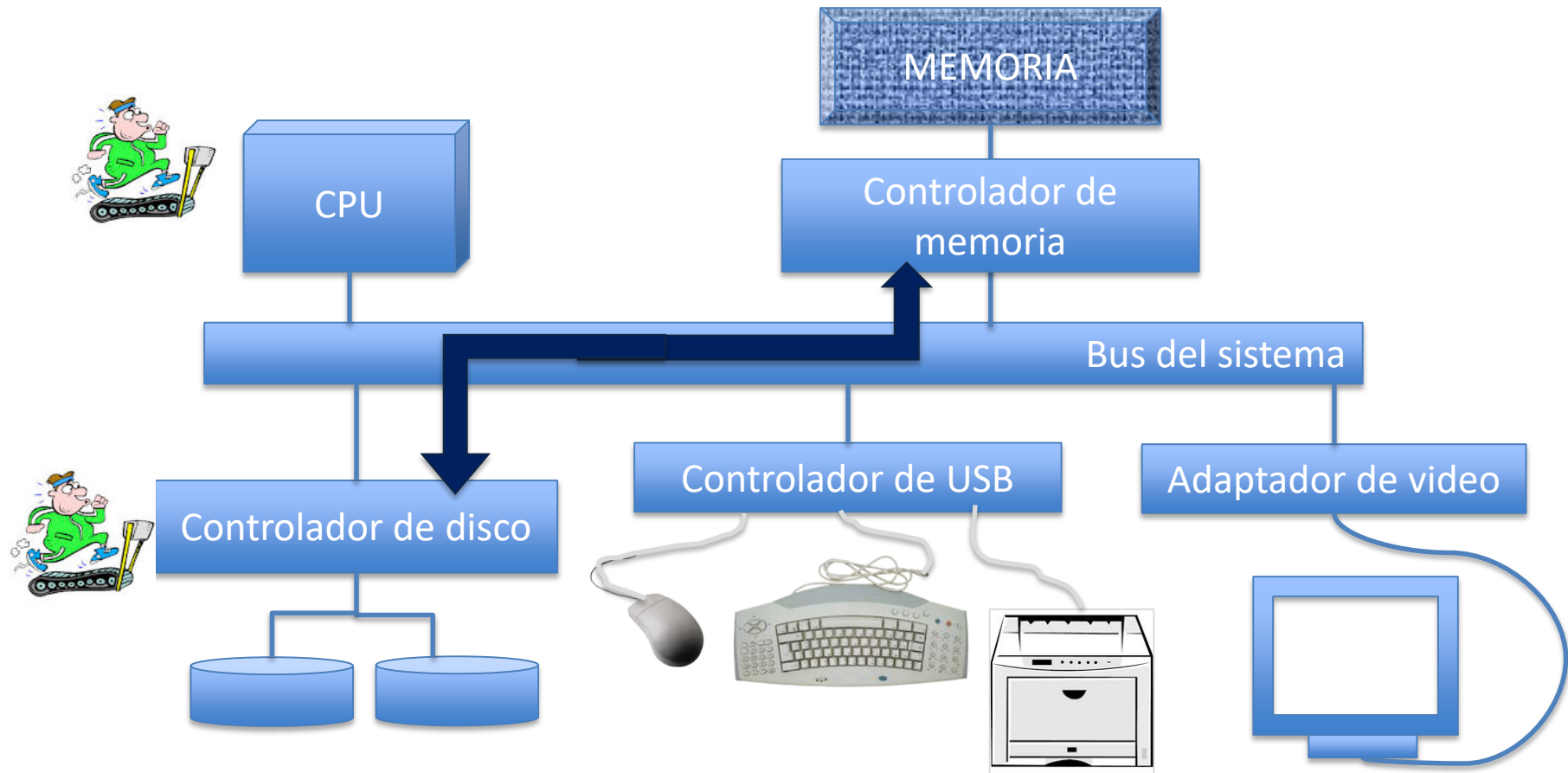
DMA	Acceso directo a memoria (<i>direct memory access</i>)
SO	Sistema Operativo
USB	Bus universal en serie (Universal Serial Bus)
INT	Interrupción
POSIX	Portable Operating System Interface

- Concurrencia entre CPU y Entrada/Salida
 - Dispositivos de **E/S más lentos que procesadores**
 - Ejemplo : tiempo de acceso a información almacenada en disco
 - Un procesador moderno puede ejecutar **miles de millones de instrucciones máquina** en el tiempo que se requiere para acceder a disco
 - Mientras se realizan operaciones de E/S el procesador debe poder ejecutar simultáneamente instrucciones útiles, esperar a que finalice la solicitud de E/S →
Concurrencia entre CPU y E/S

- Funcionamiento de un Sistema Informático



- Funcionamiento de un Sistema Informático



Concurrencia entre CPU y Entrada/Salida

CPU y dispositivos de E/S trabajan simultáneamente en el tiempo

• Manejador versus Controlador

Sistema Operativo

Manejador de Dispositivo (*Device Driver*)

- Elemento del sistema Operativo
- Software con capacidad
 - Para programa los controladores
 - Proporciona una interfaz amigable para uso del controlador



Hardware de dispositivos de E/S

Controlador de Dispositivo

- Componente hardware
- Registros de control
- *Buffer* de almacenamiento
- Registro de Estado
- Capacidad de DMA



Controlador de Dispositivo

Registro
de control

Registro
de Estado

Buffer de
datos

- Arquitectura Hardware del Computador
- **Interrupciones**
- Modos de Ejecución
- Llamadas al Sistema
- Utilidades del Sistema

- El Sistema Operativo es un **programa dirigido por eventos**
- Estos eventos son las **interrupciones hardware**, las **interrupciones software** y las **excepciones**
- El SO actúa como un programa servidor a la espera de que le soliciten trabajo mediante interrupciones
- Los procesos y los dispositivos de E/S solicitan servicios al SO



- ¿Quién genera una solicitud de interrupción?
- ¿Cuándo se genera una solicitud de interrupción?

INTERRUPCIÓN



Interrupción de E/S:

Generada por controladores de dispositivos de E/S

Interrupción de Reloj:

El SO entra a ejecutarse cada cierto intervalo

Excepción hardware:

Error de paridad en memoria, corte de corriente...



Interrupciones *Traps*:

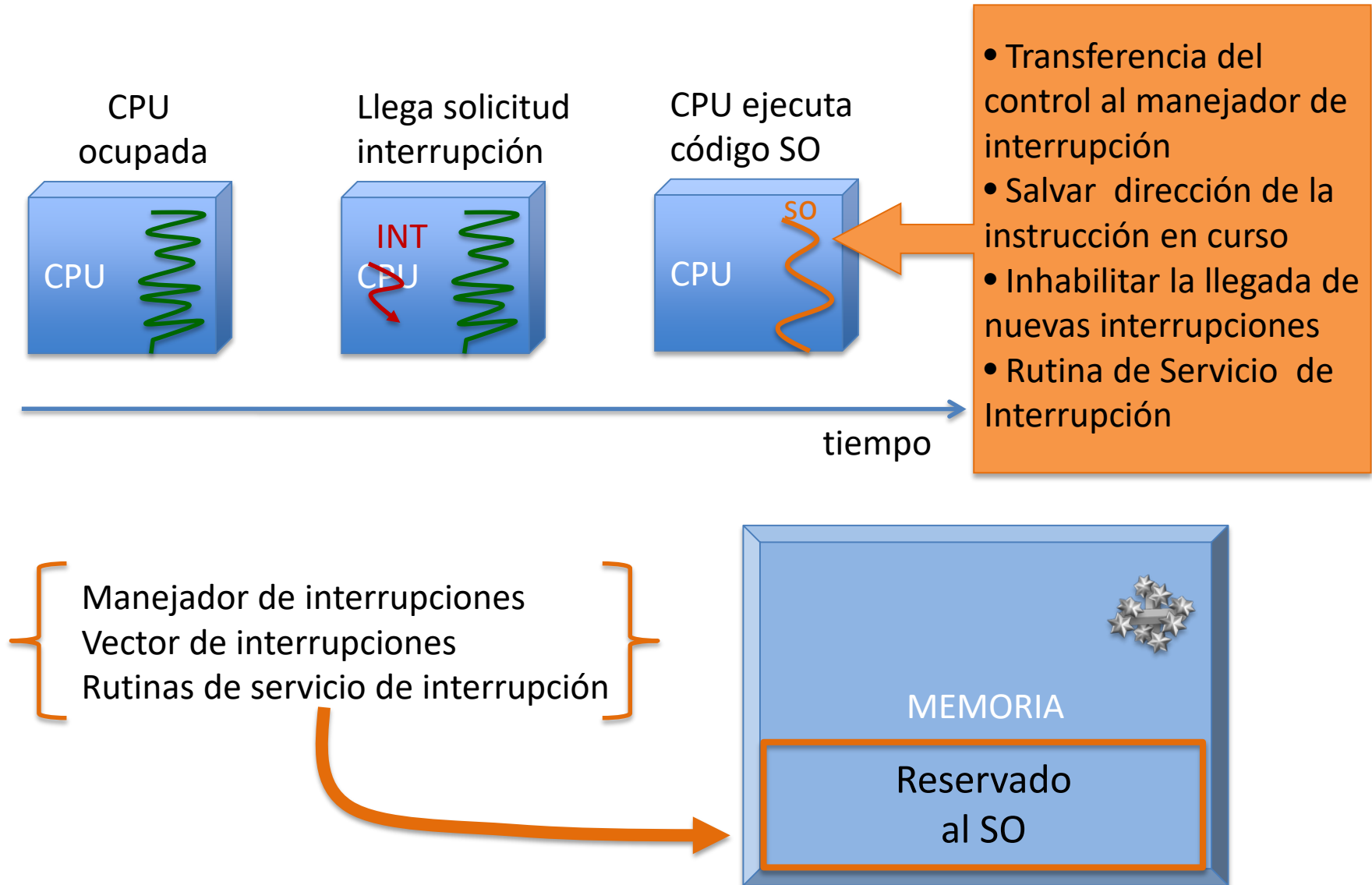
Utilizadas por los programas para solicitar servicios al SO

Excepciones software:

Se generan durante la ejecución de programas cuando ocurre un error, como una división por cero, un desbordamiento con operaciones aritméticas, o se direcciona una posición de memoria prohibida



- Mecanismo de interrupción

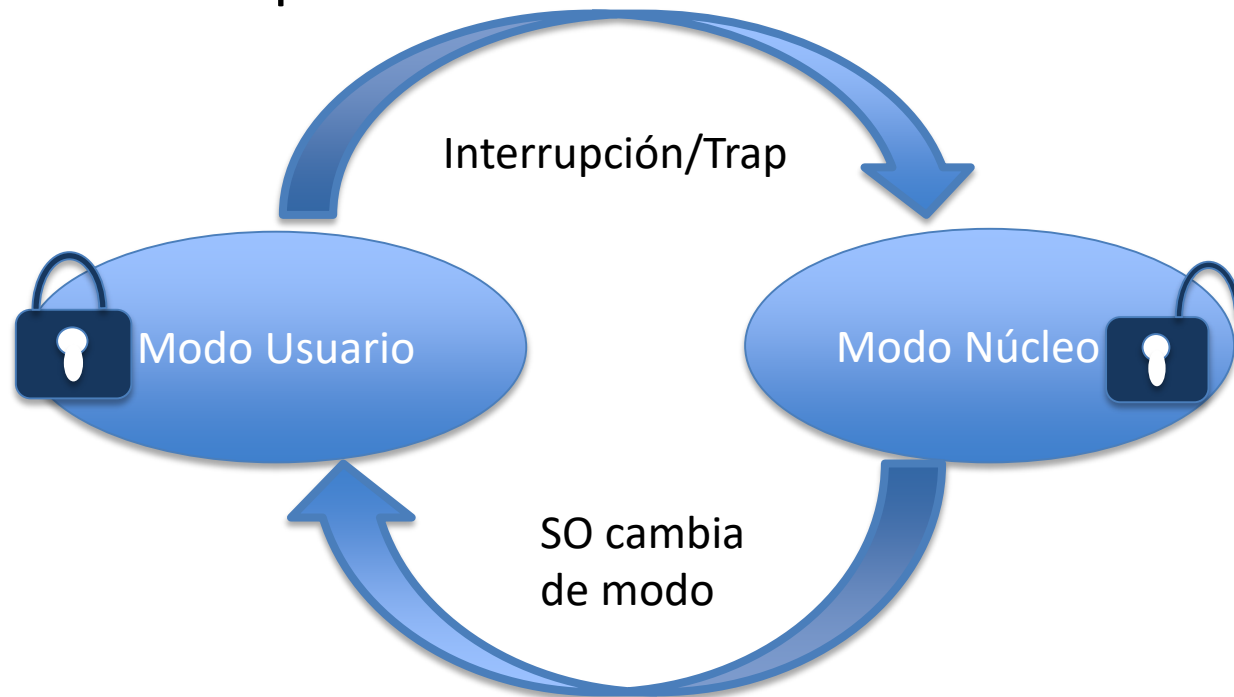


- Arquitectura Hardware del Computador
- Mecanismo de Interrupción a la CPU
- **Modos de Ejecución**
- Llamadas al Sistema
- Programas del Sistema

- Modos de ejecución del procesador
 - Los procesadores presentan **dos o más modos de ejecución**
 - Los modos de ejecución se incluyen para dar **soporte al SO**
 - Los procesos que se ejecutan simultáneamente comparten recursos de la máquina → **necesidad de protección**
 - Protección del acceso al hardware para impedir que los programas de usuarios
 - Accedan a la **memoria** del sistema libremente
 - Monopolicen el uso **de CPU**
 - Accedan a ciertos **registros del sistema**
 - Accedan directamente a los **dispositivos de E/S**
 - Los procesadores disponen de **un bit de modo**, para indicar el modo actual: **Núcleo(0) o Usuario (1)**



- Dos modos de ejecución
 - **Modo Usuario o Regular:** Tiene restringido el juego de instrucciones
 - **Modo Núcleo o Privilegiado:** Permite ejecutar todo tipo de operaciones hardware, acceder a memoria y a los dispositivos de E/S
- Modo dual de Operación

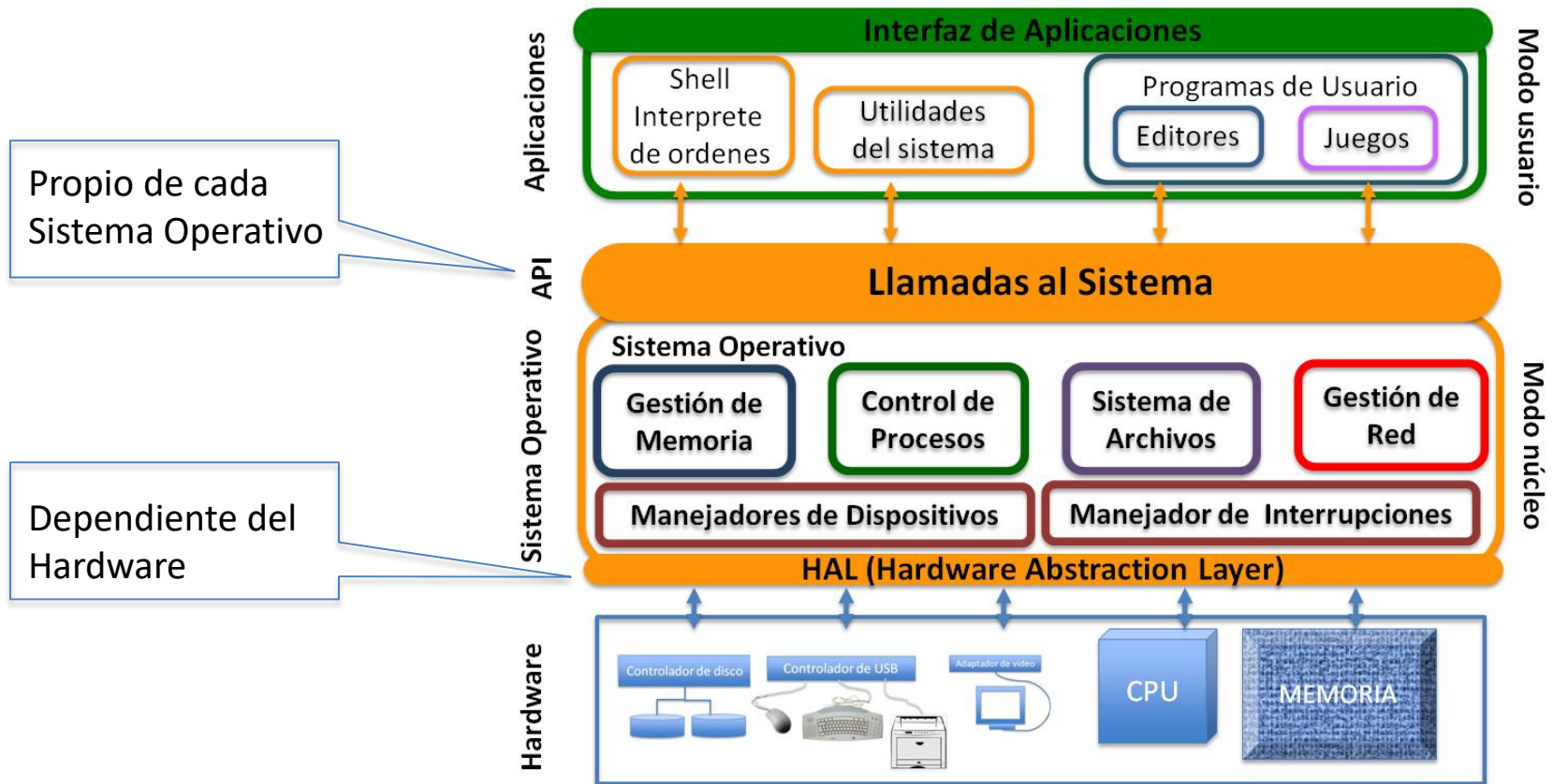




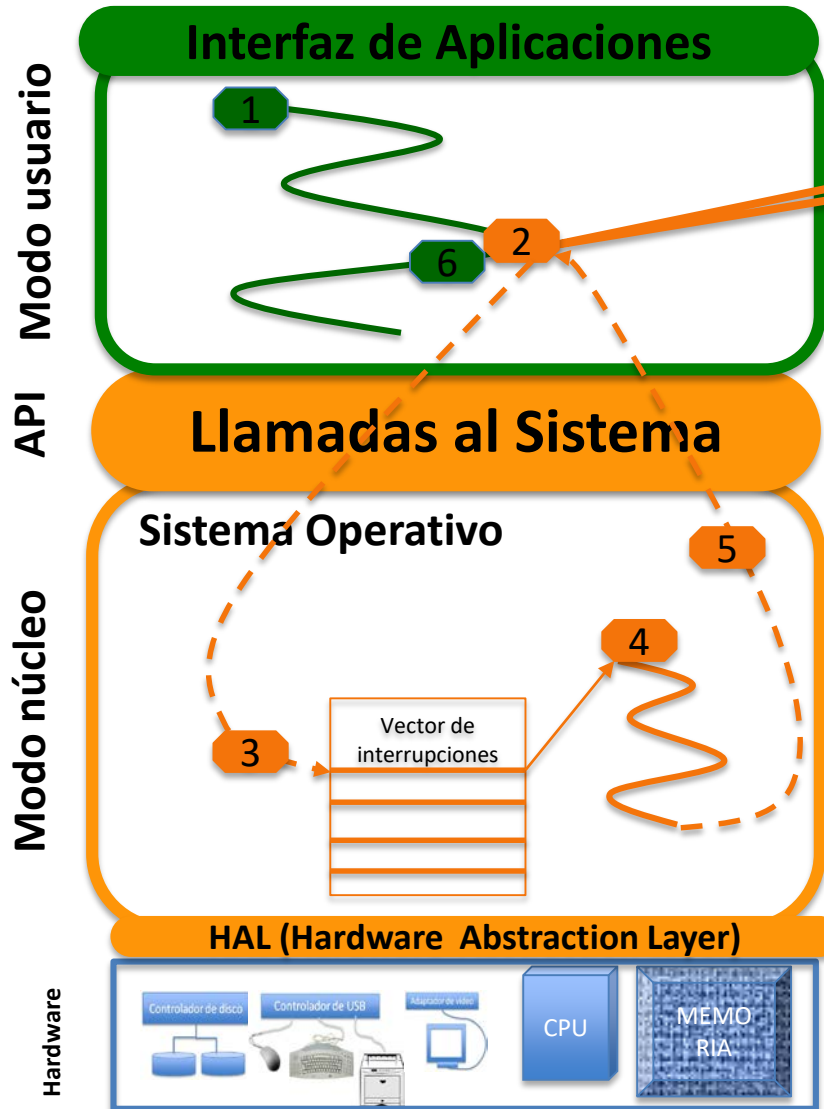
- Instrucciones privilegiadas
 - Son aquellas instrucciones **disponibles en modo Núcleo** y que no lo están en modo usuario
 - Las instrucciones privilegiadas están asociadas principalmente a tres tipos de protección:
 - Protección de la Entrada/Salida
 - Protección de la memoria
 - Protección del procesador

- Arquitectura Hardware del Computador
- Mecanismo de Interrupción a la CPU
- Modos de Ejecución
- **Llamadas al Sistema**
- Utilidades del Sistema

- Mecanismos para **solicitar servicios** al sistema operativo
- **Interfaz** proporcionado por el SO para acceder a los recursos hardware de la máquina
- Utilidad en forma de **funciones de biblioteca** para acceder a recursos gestionados por el sistema operativo



- Solicitud de un servicio al sistema operativo



Llamada al sistema

Modo Usuario

1. Programa ejecutándose
2. Llamada a sistema (Trap o Interrupción software)

Modo Núcleo

3. Identificación del servicio solicitado
4. Ejecución del servicio solicitado
5. Datos solicitados, resultado del servicio

Modo Usuario

6. Siguiente instrucción del programa

¡¡AVISO!!

¿Compatibilidad entre SO? NO

Cada sistema operativo tiene sus propias llamadas

- POSIX: Ejemplo de llamadas
- Estándar POSIX (Portable Operating System Interface)

	Procesos
fork	Creación de un proceso hijo
exit	Terminación del proceso en ejecución
wait	Espera la terminación de un proceso
exec	Cambia imagen de memoria por la de un ejecutable (ejecuta programa)
getpid	Obtiene atributos de un proceso
setsid	Modifica atributos de un proceso

	Protección
chmod	Modifica los bits de permiso (rwx, suid,sgid,...) de un fichero
chown	Asigna nuevo propietario y grupo a un fichero
umask	Modifica la mascara de protección de un proceso

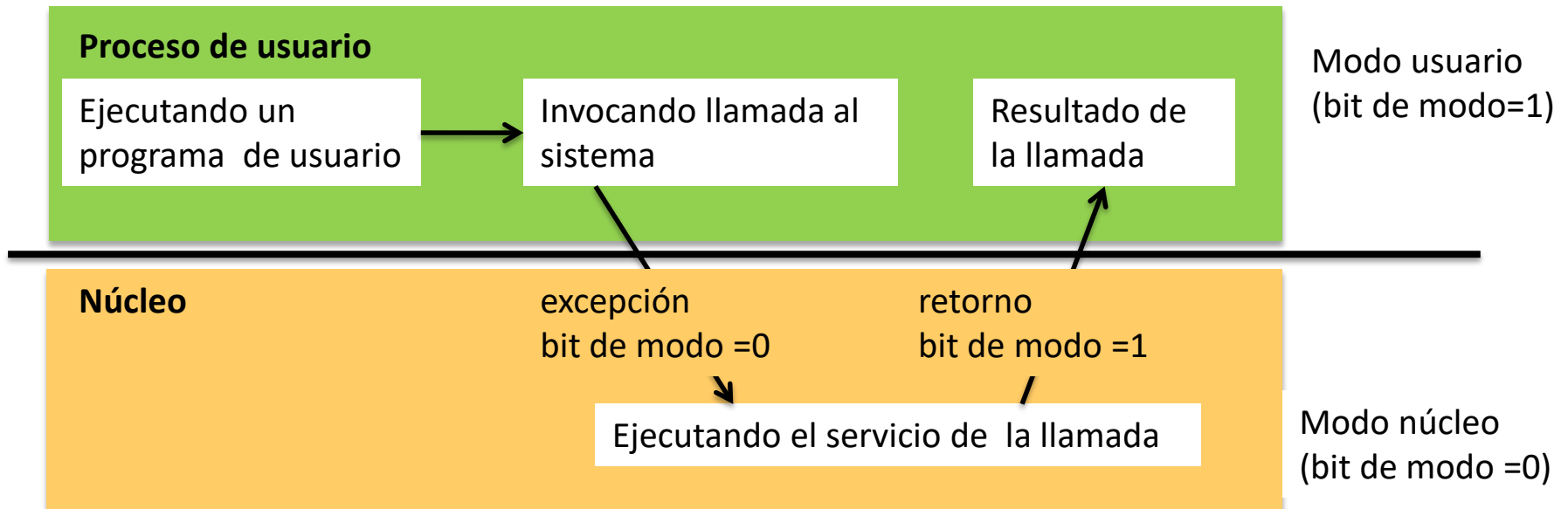
	Directories
mkdir	Create directory
rmdir	Remove an empty directory
opendir	Open directory
readdir	Get the next directory entry
closedir	Close directory
link	Get information about file i-node
unlink	Remove a directory entry

	Señales
Kill	Enviar señales
alarm	Generar una alarma (señal de reloj)
sigemptyset	Iniciar una máscara para que no tenga señales seleccionadas
sigfillset	Iniciar una máscara para que contenga señales seleccionadas
sigaddset	Añadir una señal concreta a un conjunto de señales
sigdelset	Borrar una señal concreta de un conjunto de señales
sigismember	Consultar si una señal concreta pertenece a un conjunto de señales
sigprocmask	Examinar /Modificar /Establecer una máscara de señales
sigaction	Capturar/Manejar una señal
sigsuspend	Esperar la captura señales

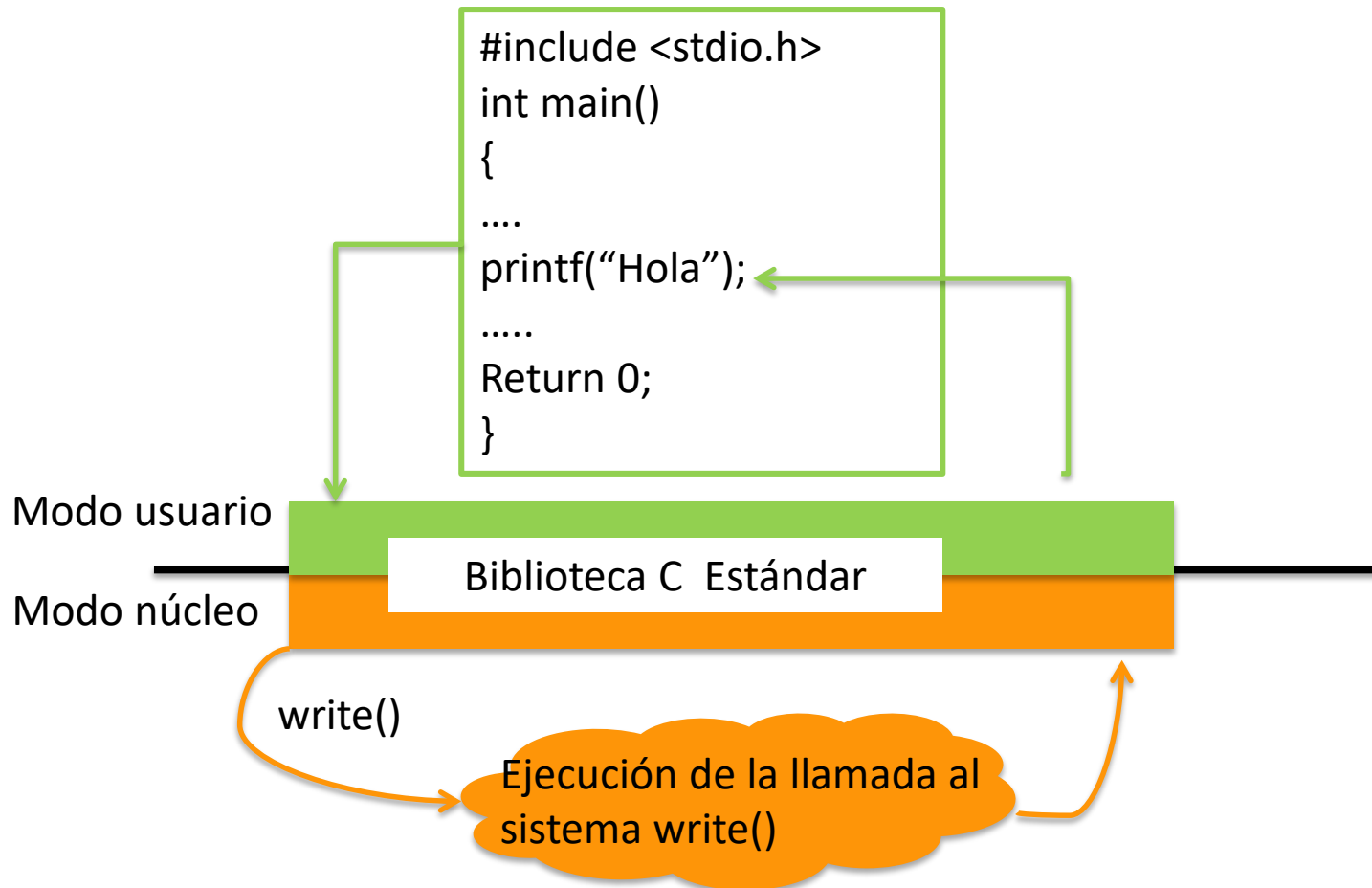
	Ficheros
Open	Abrir/Crear ficheros
read	Lectura de ficheros
write	Escritura de ficheros
close	Cerrar ficheros
lseek	Posicionamiento de lectura/escritura en fichero
stat	Obtener información del nodo-i de un fichero
dup2	Duplica un descriptor de fichero
pipe	Creación de tubo
mkfifo	Creación de tubo con nombre (fifo)

Las llamadas al sistema utilizan el cambio de modo para poder acceder a los recursos del sistema

- Con el **bit de modo** se diferencia entre tareas ejecutadas por el SO y las ejecutadas a nivel de usuario
- Cuando se arranca el sistema, el hardware se inicia en modo núcleo (bit de modo = 0), entonces se carga el sistema operativo y se inician las aplicaciones de usuario en modo usuario (bit de modo=1)



- **Bibliotecas del lenguaje C estándar**
 - Proporcionan una interfaz portable a muchas llamadas al sistema.



- Arquitectura Hardware del Computador
- Mecanismo de Interrupción a la CPU
- Modos de Ejecución
- Llamadas al Sistema
- **Utilidades del Sistema**

- **Utilidades del sistema operativo**
 - Se ejecutan como procesos de usuario y proporcionan un **entorno más cómodo** de trabajo
 - Se proporciona como parte del SO, pero no son imprescindibles para el funcionamiento de la máquina
 - Ejemplos en UNIX
 - Tratamiento de ficheros: mkdir, cp, mv, ls
 - Filtros: grep, sort, head, tail
 - Editores, compiladores, ensambladores, editores de enlace ...
 - Sistemas de ventanas: X11
 - Comunicaciones: mail, ftp, rlogin
 - Intérpretes de órdenes: sh, ksh, bash

- Ejemplo “*mi_copy*”

```
#define BUFSIZE 1024

void copy(char *from, char *to)
{
    int fromfd, tofd, nread;
    char buf[BUFSIZE];

    if ((fromfd = open(from, O_RDONLY)) == -1)
        { perror(from); exit(1) }
    if ((tofd = creat(to, 0666)) == -1)
        { perror(to); exit(1); }
    while ((nread = read(fromfd, buf, sizeof (buf))) > 0)
        if (write(tofd, buf, nread) != nread)
            { perror("write"); exit(1); }
    if (nread == -1) perror("read");
    if (close(fromfd) == -1 || close(tofd) == -1) perror("close");
}
```

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
int main(int argc, char *argv[])
{
    if (argc!=3) {
        fprintf(stderr, "Uso: %s forigen"
            " fdestino\n", argv[0]);
        exit(1); }
    copy(argv[1], argv[2]);
    return 0;
}
```