

3.3 El Sistema de Entrada y Salida

Gestión de la comunicación entre el sistema informático y los dispositivos periféricos



Índice de Contenidos



3.3 El sistema de entrada y salida

- Definición y función fundamental
- Importancia en el sistema operativo
- Clasificación de dispositivos de E/S
- Funciones esenciales
- Estructura



3.3.1 Acceso directo a memoria

- Concepto y definición
- Funcionamiento del DMA
- Ventajas del DMA
- Comparación con métodos tradicionales



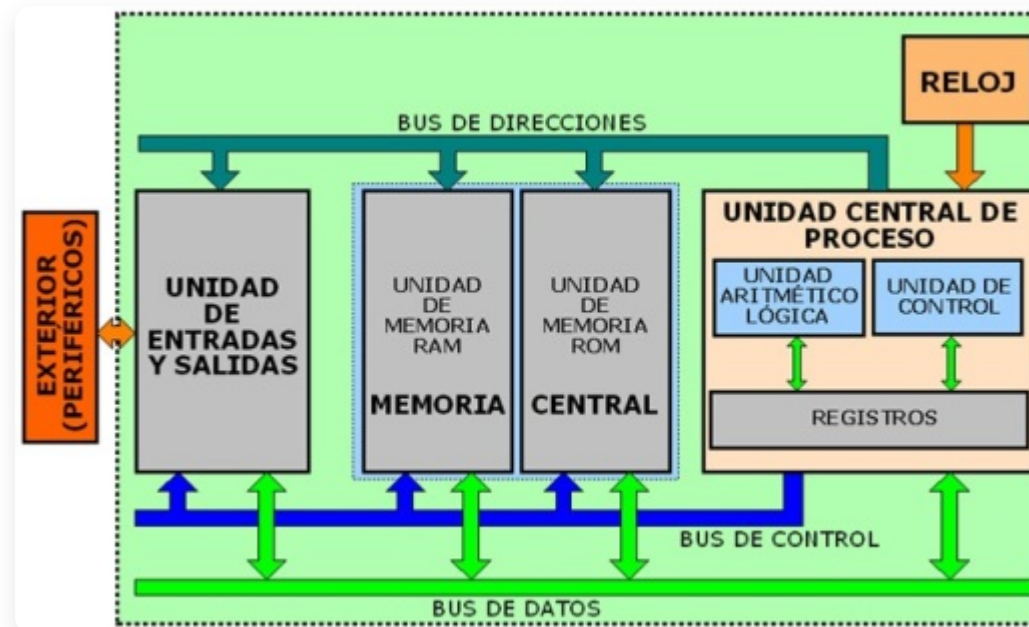
3.3.2 Manejadores de dispositivos

- Definición y propósito
- Arquitectura en capas
- Funcionamiento
- Gestión de colas de peticiones
- Relación con controladores

Definición y función fundamental



El sistema de **entrada y salida (E/S o I/O)** es una de las funciones más importantes y complejas que realiza el sistema operativo. Su función principal es **gestionar la comunicación** entre el sistema informático y los dispositivos periféricos, actuando como intermediario entre la CPU y los dispositivos externos.



Esquema del sistema de computadora mostrando la interacción entre CPU, memoria y unidades de E/S

Importancia en el sistema operativo



Los dispositivos periféricos son **significativamente más lentos** que la CPU



Existe una **gran variedad de dispositivos** con características y protocolos diferentes



La **eficiencia del sistema global** depende en gran medida de la eficacia del sistema de E/S



Los **fallos en dispositivos de E/S** son comunes y deben gestionarse adecuadamente



Diversidad de dispositivos periféricos que requieren gestión eficiente

Clasificación de dispositivos de E/S



Dispositivos orientados a caracteres

- Trabajan con **flujos de caracteres individuales**
- Ejemplos: teclados, ratones, impresoras
- No tienen estructura de bloques
- Acceso secuencial



Dispositivos orientados a bloques

- Trabajan con **bloques de datos de tamaño fijo**
- Ejemplos: discos duros, unidades USB, tarjetas de memoria
- Tienen estructura de bloques
- Permiten acceso aleatorio
- Soportan DMA (Acceso Directo a Memoria)

Categorías de dispositivos

- Dispositivos de bloques

- » Bloques de tamaño fijo
- » L/E de cada bloque de forma independiente
- » Se puede direccionar cada bloque

DISCOS

- Dispositivos de caracteres

- » Sin estructura de bloques
- » L/E de flujos de caracteres

TERMINALES

IMPRESORAS

Funciones esenciales del sistema de E/S



Enviar comandos y datos

Formatear órdenes según **especificaciones** de cada dispositivo



Conocer estado de dispositivos

Verificar si están **listos, ocupados o en error**



Detectar interrupciones

Responder a señales cuando dispositivos **completan operaciones**



Controlar errores

Detectar y gestionar errores en **operaciones de E/S**



Gestionar buffers

Almacenar temporalmente datos para **compensar diferencias de velocidad**



Proporcionar interfaz uniforme

Ocultar **peculiaridades de hardware** a las aplicaciones

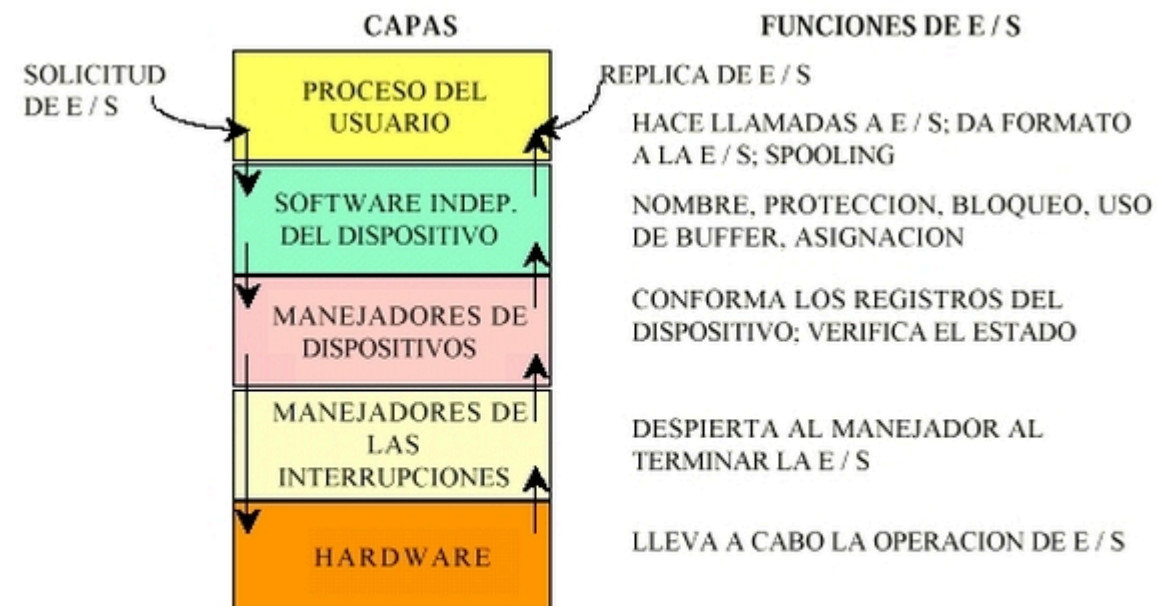


Figura 5.3: Capas del sistema de entrada / salida y las principales funciones de cada capa.

Esquema del sistema de E/S mostrando sus funciones principales

Estructura del sistema de E/S

1

Nivel de usuario

Interfaz para **aplicaciones y usuarios**

2

Nivel independiente del dispositivo

Proporciona una **interfaz uniforme** para todos los dispositivos

3

Nivel de controlador de dispositivo

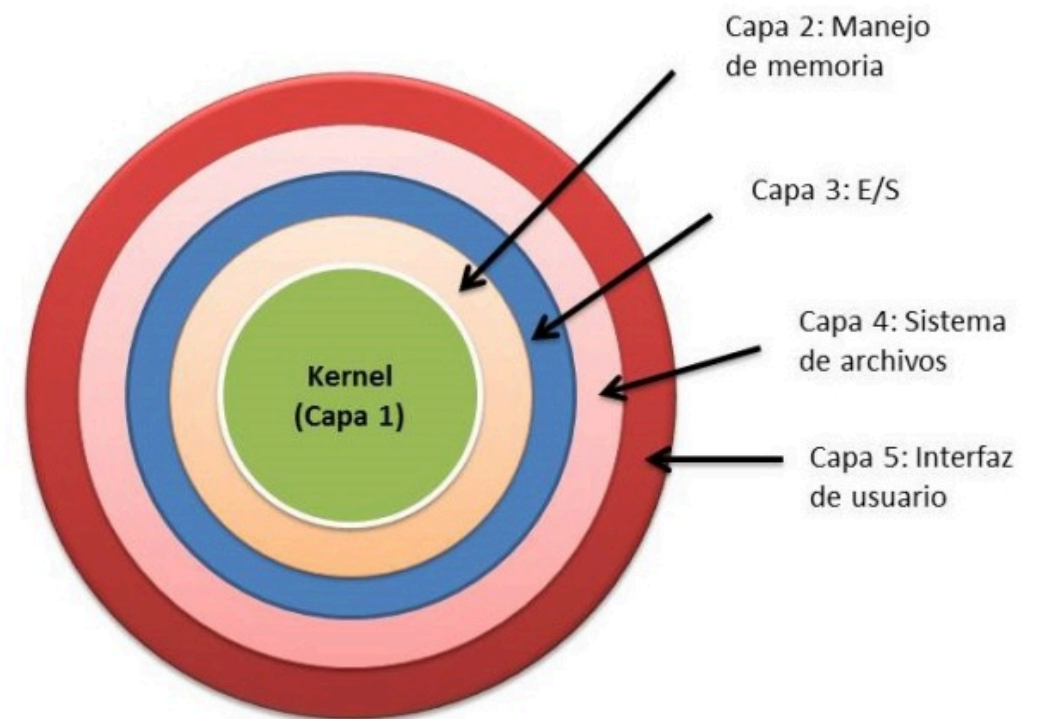
Traduce solicitudes abstractas a **comandos específicos** del hardware

4

Hardware del dispositivo

Los **dispositivos físicos** y sus circuitos de control

El controlador del dispositivo se encarga de mostrar al sistema operativo una interfaz estándar (dispositivo virtual) que hace transparentes al sistema operativo las peculiaridades del hardware.



Sistema operativo

Acceso directo a memoria - Concepto y definición

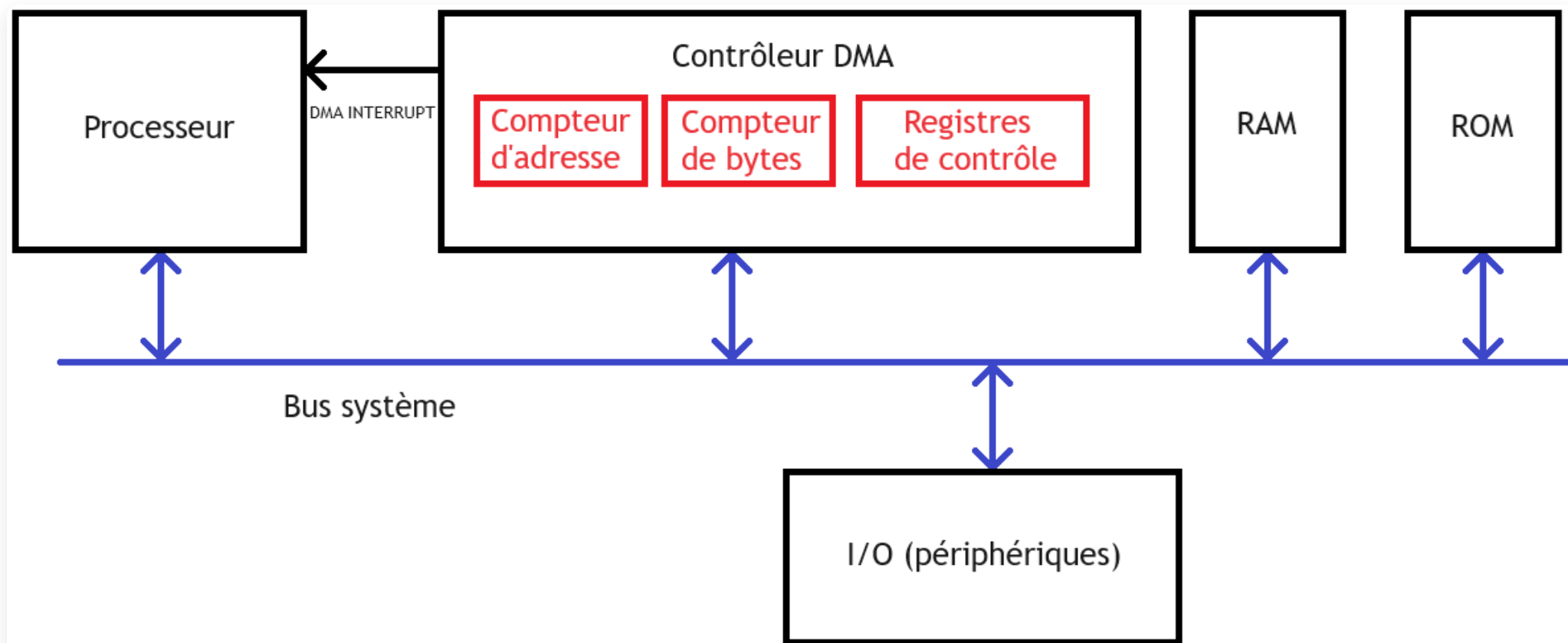


El **Acceso Directo a Memoria (DMA - Direct Memory Access)** es una técnica que permite a ciertos dispositivos de hardware acceder directamente a la memoria principal **sin intervención constante de la CPU**. Es especialmente útil para dispositivos de E/S de alto rendimiento.

 Discos duros

 Tarjetas de red

 Almacenamiento masivo



Esquema del sistema informático mostrando la integración del controlador DMA

Funcionamiento del DMA

1 Solicitud de transferencia

- Dirección de memoria
- Número de datos
- Dirección del dispositivo

2 Transferencia independiente

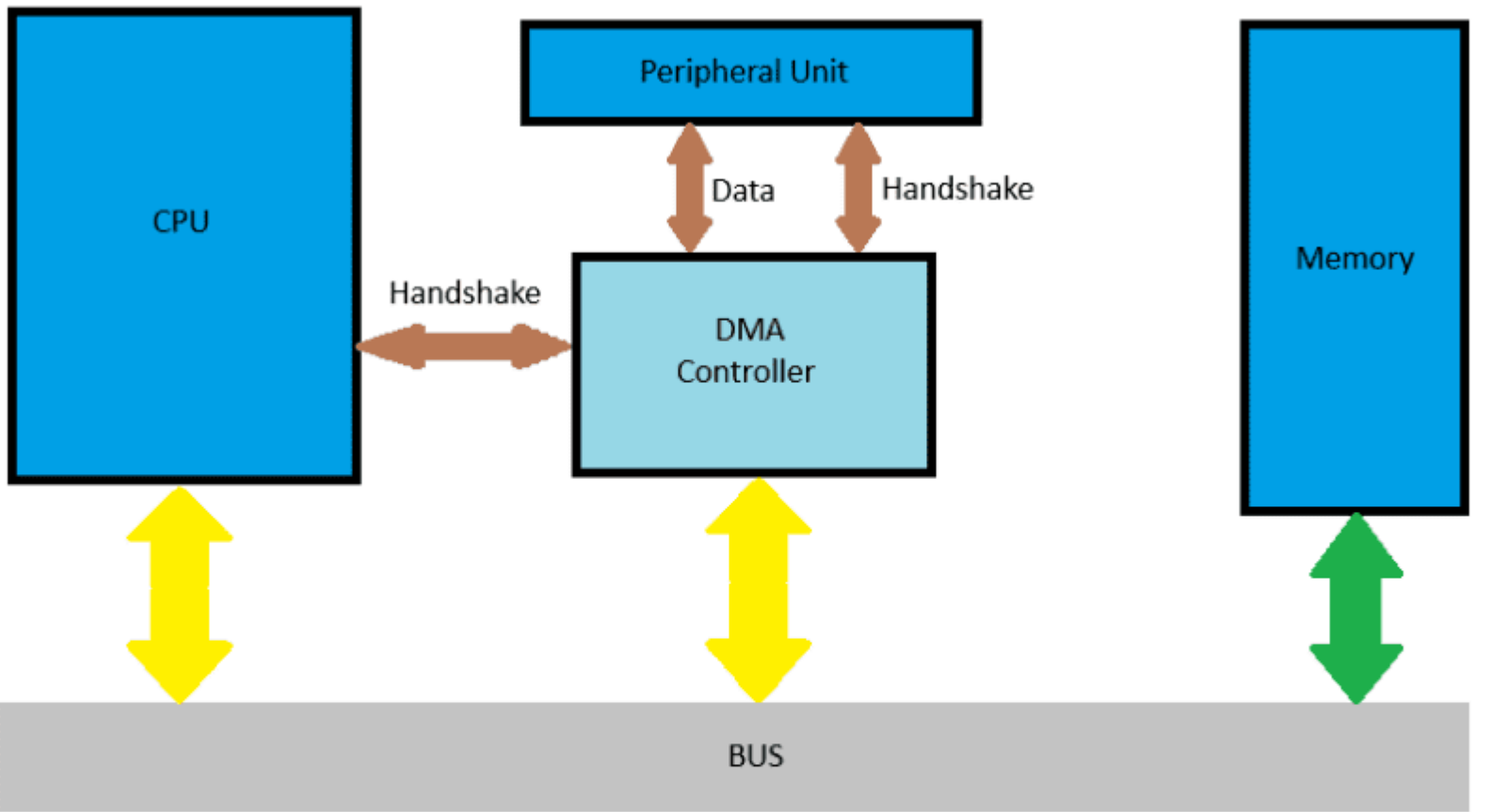
- CPU continúa otras tareas

3 Operación de E/S

- Lectura desde dispositivo
- Escritura directa en memoria
- Sin intervención de CPU

4 Interrupción final

- Notificación a CPU
- Operación completada



- Master Interface
- Slave Interface

Esquema de conexión entre CPU, controlador DMA, memoria y dispositivos periféricos

Ventajas del acceso directo a memoria



Solapa operaciones

Permite que la CPU **realice otras tareas** mientras se realiza la transferencia



Reduce carga de CPU

Elimina la necesidad de gestionar **cada byte** transferido



Aumenta rendimiento

Mejora significativamente la **eficiencia** en operaciones de E/S intensivas



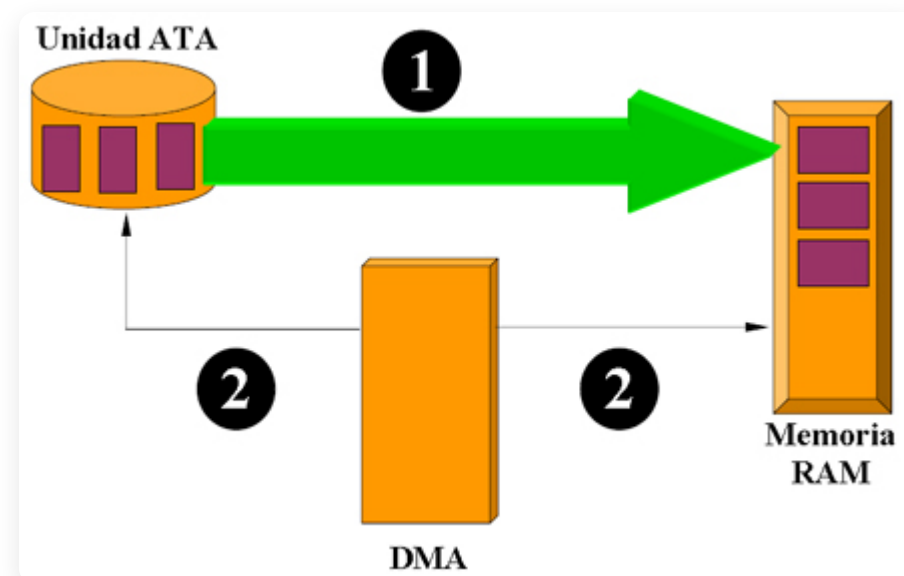
Disminuye latencia

Permite transferencias **más rápidas** al eliminar sobrecarga de CPU



Eficiencia energética

Reduce el trabajo de CPU, importante en **dispositivos móviles**



Esquema de transferencia de datos mediante DMA mostrando la eficiencia del proceso

Comparación con métodos tradicionales de E/S

⌚ Métodos tradicionales sin DMA

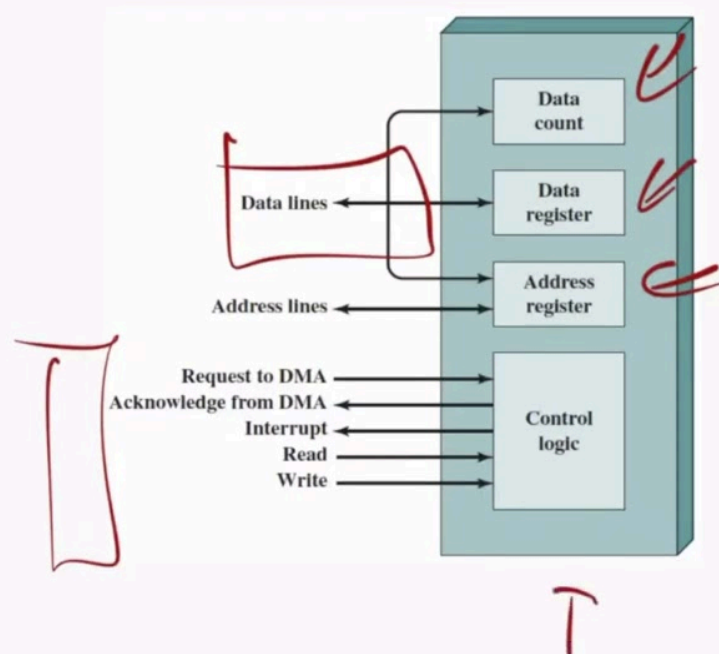
- ▶ La CPU **debe gestionar** cada transferencia de datos
- ▶ Para cada byte/word transferido:
 - Leer el dato del dispositivo
 - Almacenar el dato en memoria
 - Repetir para el siguiente dato

🔧 Con DMA

- ▶ La CPU **configura una única operación** para toda la transferencia
- ▶ El controlador DMA **gestiona la transferencia completa**
- ▶ La CPU solo se notifica **al finalizar** la operación

Esta diferencia es especialmente significativa en operaciones de E/S grandes, donde el ahorro de ciclos de CPU puede ser del 90% o más, permitiendo que la CPU se dedique a tareas más críticas.

Funcionamiento del DMA



Para dar una instrucción al DMA se debe de enviar la siguiente información:

- Para solicitar una transacción se utilizan las líneas de *read* y *write* de la lógica de control.
- La dirección del dispositivo de I/O involucrado.
- La dirección inicial en memoria para leer o escribir a través de las líneas de datos.
- El número de datos a leer o escribir a través de la línea de datos.

Diagrama que ilustra el funcionamiento del Acceso Directo a Memoria (DMA)

Manejadores de dispositivos - Definición y propósito



Los **manejadores de dispositivos** (también conocidos como drivers) son programas especializados que permiten al sistema operativo comunicarse con los dispositivos de hardware. Su propósito principal es **aceptar solicitudes abstractas** y ponerse en contacto con el controlador para realizar esa petición.



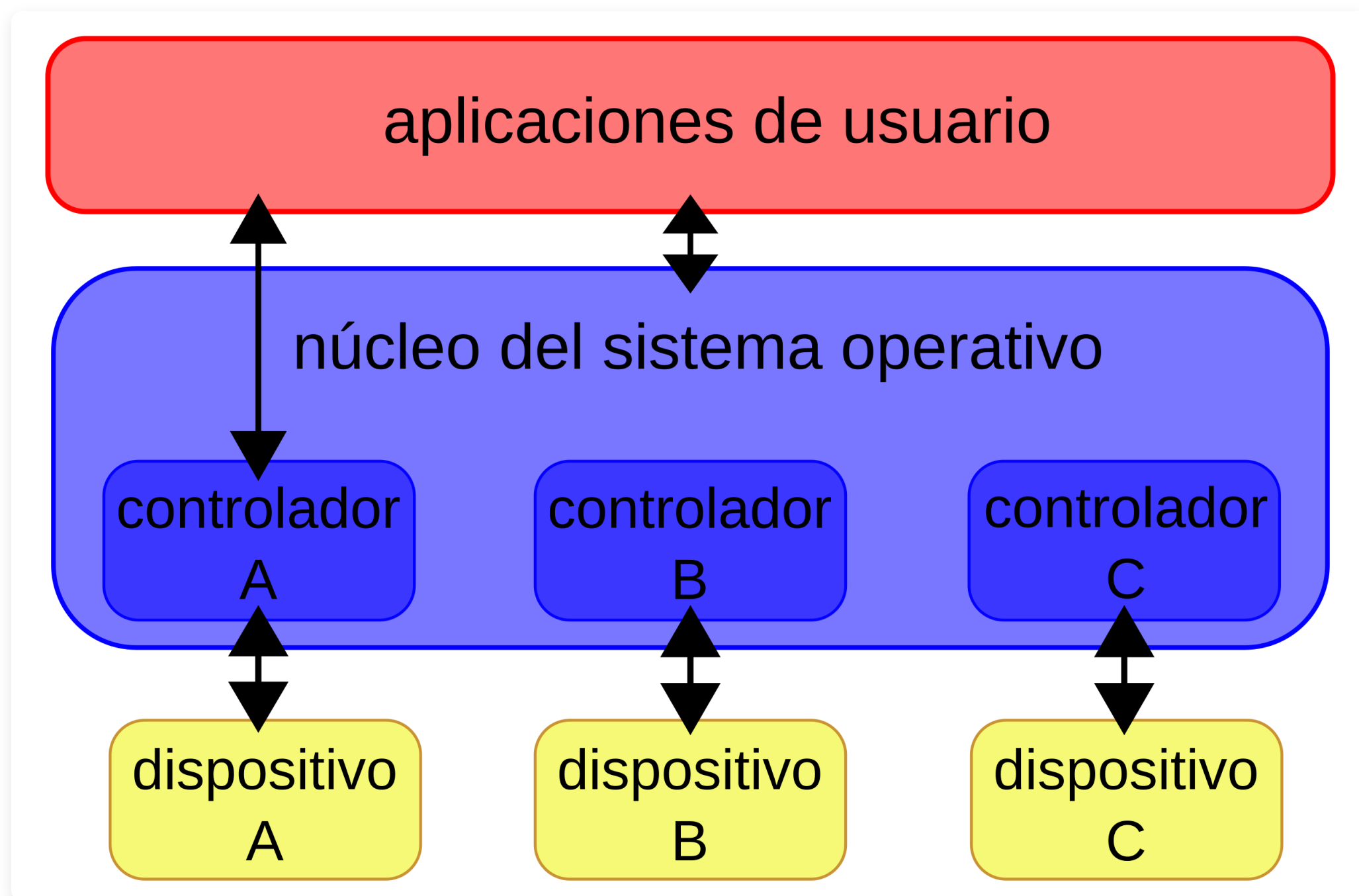
Intermediario entre SO y hardware



Traduce órdenes abstractas



Gestiona dispositivos específicos



Arquitectura del sistema operativo mostrando la relación entre aplicaciones, kernel y controladores de dispositivos

Arquitectura en capas del software de E/S

1 **Capa de usuario**
Interfaz para **aplicaciones y usuarios**

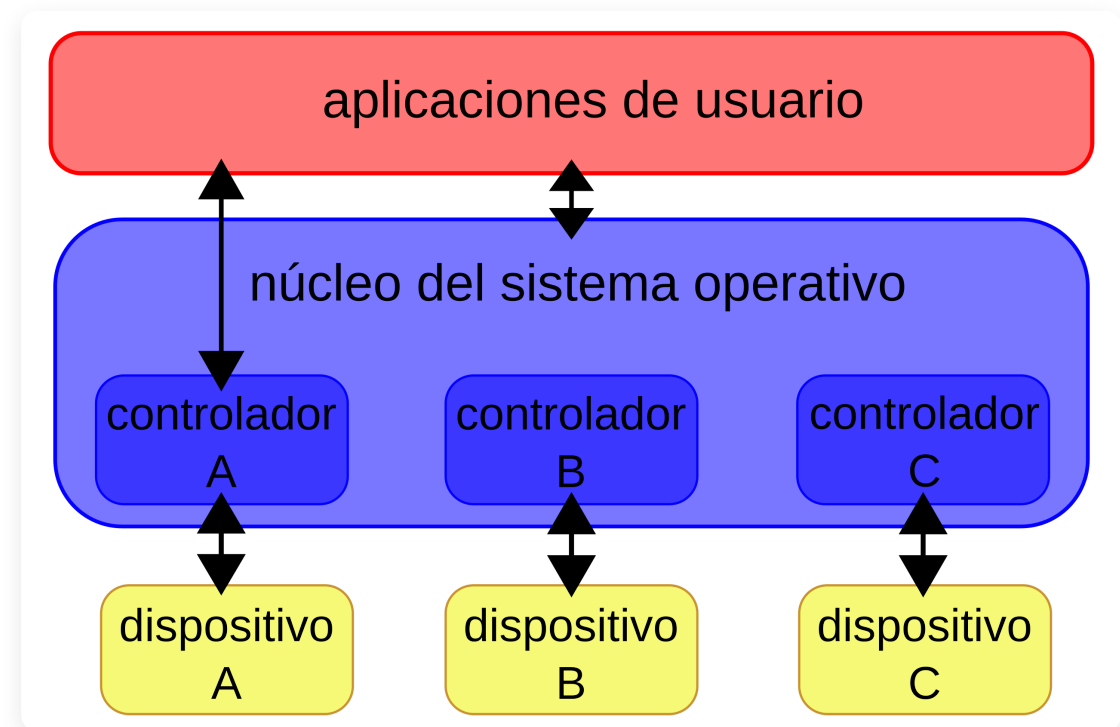
2 **Capa independiente del dispositivo**
Proporciona una **interfaz uniforme** para todos los dispositivos

3 **Capa de controlador de dispositivo**
Traduce solicitudes abstractas a **comandos específicos** del hardware

4 **Hardware del dispositivo**
Los **dispositivos físicos** y sus circuitos de control

★ Ventajas del diseño en capas

- ✓ Oculta peculiaridades del hardware
- ✓ Interfaz amigable para usuarios
- ✓ Sistema modular y fácil de mantener
- ✓ Añadir dispositivos sin modificar el sistema



Funcionamiento de los manejadores



Traducción de solicitudes

Convierte solicitudes genéricas en **comandos específicos** para el dispositivo



Gestión de estado

Mantiene información sobre el estado actual: **listo, ocupado, en error**



Manejo de interrupciones

Procesa señales del dispositivo para **notificar** el estado de las operaciones



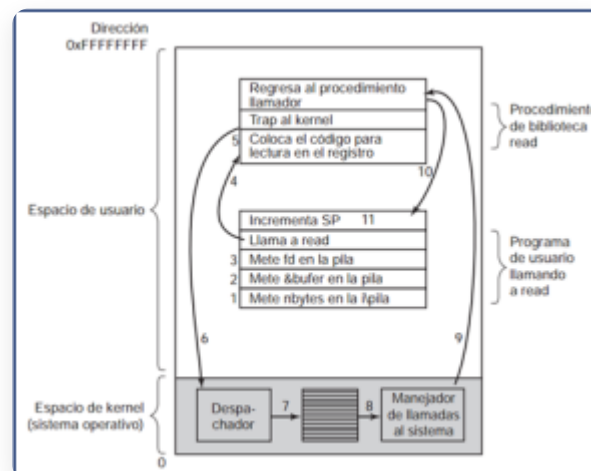
Buffering

Gestiona búferes para **compensar diferencias** de velocidad entre CPU y dispositivo



Control de errores

Detecta y gestiona errores en las **operaciones de E/S**



Esquema que muestra el flujo de una llamada al sistema read a través de las capas del manejador

Gestión de colas de peticiones

+ Dispositivo ocupado

Cuando el dispositivo está ocupado, el manejador **gestiona una cola** de peticiones para darles paso tan pronto como sea posible

≡ Ordenación de peticiones

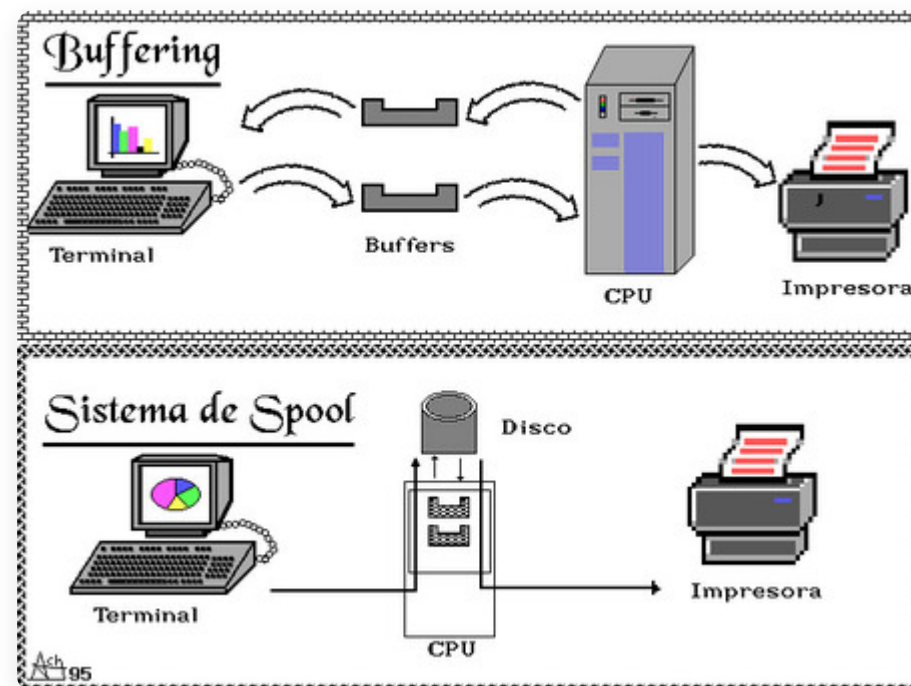
Algunos manejadores implementan **algoritmos de ordenación** óptima (ej: algoritmo del elevador en discos duros para minimizar movimiento del cabezal)

! Priorización

Las peticiones pueden tener **diferentes niveles de prioridad**, y el manejador debe gestionarlas en consecuencia

≡ Ejemplo de cola de peticiones

- 1 Petición de alta prioridad (interrupción)
- 2 Petición de media prioridad (lectura)
- 3 Petición de baja prioridad (escritura)
- 4 Petición de baja prioridad (lectura)



Esquema que ilustra el sistema de gestión de colas y buffers en dispositivos de E/S

Relación con los controladores de dispositivos

<> Manejador de dispositivo

- ✓ **Software** que se ejecuta en el espacio del sistema operativo
- ✓ Proporciona una interfaz **estandarizada**
- ✓ Traduce órdenes abstractas a comandos específicos

🔧 Controlador de dispositivo

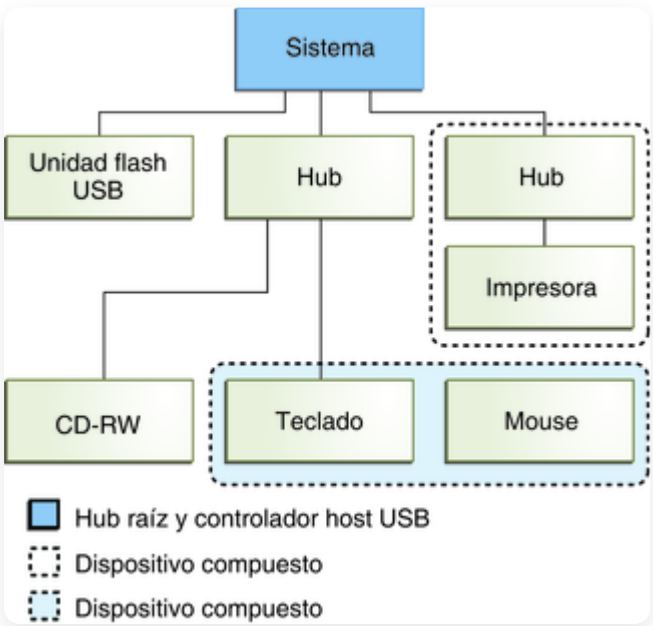
- ✓ Componente de **hardware** físico
- ✓ Se comunica **directamente** con el dispositivo físico
- ✓ Implementa protocolos específicos del dispositivo

↔ Interacción entre manejador y controlador

El manejador se comunica con el controlador para realizar las operaciones físicas en el dispositivo. Muchos controladores, en particular los correspondientes a dispositivos de bloque, permiten el **DMA**, lo que mejora significativamente el rendimiento del sistema al solapar operaciones de CPU y E/S.

🖥 Gestión en Windows

En Windows, los manejadores de dispositivos pueden gestionarse a través del **Administrador de dispositivos**, donde se pueden ver las propiedades, actualizar drivers, resolver conflictos de recursos y gestionar el estado de los dispositivos del sistema.



Esquema de un sistema USB mostrando la relación entre controladores y dispositivos