

# ENTRADA/SALIDA

---

Universidad San Pablo-CEU  
Escuela Politécnica Superior  
Rodrigo García Carmona

# OBJETIVOS

- Entender qué son los **dispositivos de Entrada/Salida** y ser consciente de los problemas que implica su manejo.
- Analizar las diferentes **estrategias** de Entrada/Salida.
  - Programada.
  - Mediante interrupciones.
  - Usando DMA.
- Comprender el por qué de la **estructura en capas** de los sistemas de control de Entrada/Salida.
- Entender las diferencias entre los diversos **tipos de buffer** y sus implicaciones de diseño.

# CONTENIDO

- Entrada/Salida
- Estructura en capas
- Acceso Directo a Memoria
- Modelos de *buffer*

## Bibliografía

- W. Stallings:  
**Sistemas Operativos.**
  - Capítulo 11.
- A.S. Tanenbaum:  
**Modern Operating Systems.**
  - Capítulo 5.

# ENTRADA/SALIDA

# ENTRADA/SALIDA

- Componentes de un ordenador:
  - CPU
  - Memoria principal
  - **Dispositivos de Entrada/Salida**
- Categorización:
  - **Interacción con el usuario:** teclado, ratón, pantalla...
  - **Interacción entre máquinas:** disco duro, impresora...
  - **De comunicación:** Ethernet, Wi-Fi...
- Una gran diversidad de dispositivos...
  - Es preciso manejarlos de forma **uniforme**.
- ...de baja velocidad comparados con la CPU y la memoria principal.
  - Es preciso manejarlos de forma **eficiente**.

# CARACTERÍSTICAS

- Los dispositivos de Entrada/Salida pueden clasificarse mediante varias características:
  - Velocidad de transferencia.
  - Funciones.
  - Usuarios.
  - Complejidad de control.
  - Codificación de datos.
  - Modo de transferencia:
    - Dispositivos **de bloque**.
    - Dispositivos **de flujo de caracteres**.
  - Control de errores.

# TIPOS DE GESTIÓN DE LA ENTRADA/SALIDA

- **Entrada/Salida programada:**

- Un proceso solicita el acceso a un dispositivo.
- El procesador envía una orden a un módulo de Entrada/Salida.
- El proceso realiza espera activa.
- El procesador gestiona el intercambio.

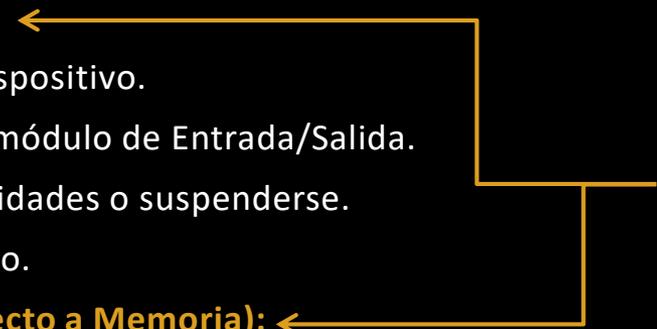
- **Entrada/Salida mediante interrupciones:**

- Un proceso solicita el acceso a un dispositivo.
- El procesador envía una orden a un módulo de Entrada/Salida.
- El proceso puede realizar otras actividades o suspenderse.
- El procesador gestiona el intercambio.

- **Entrada/Salida usando DMA (Acceso Directo a Memoria):**

- Un proceso solicita el acceso a un dispositivo.
- El procesador delega en un módulo DMA.
- El proceso puede realizar otras actividades o suspenderse.
- El procesador no gestiona el intercambio.

**Eficiencia**

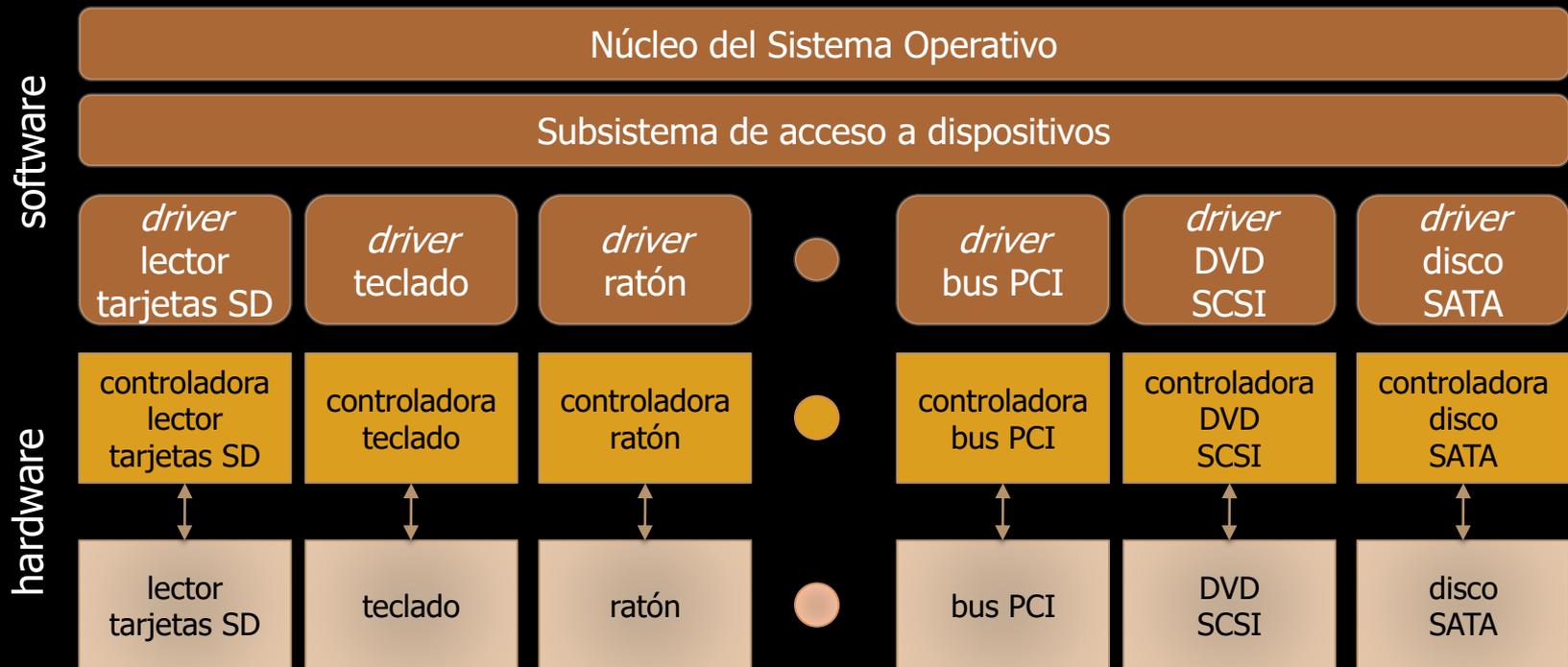


# ESTRUCTURA EN CAPAS

# ESTRUCTURA EN CAPAS

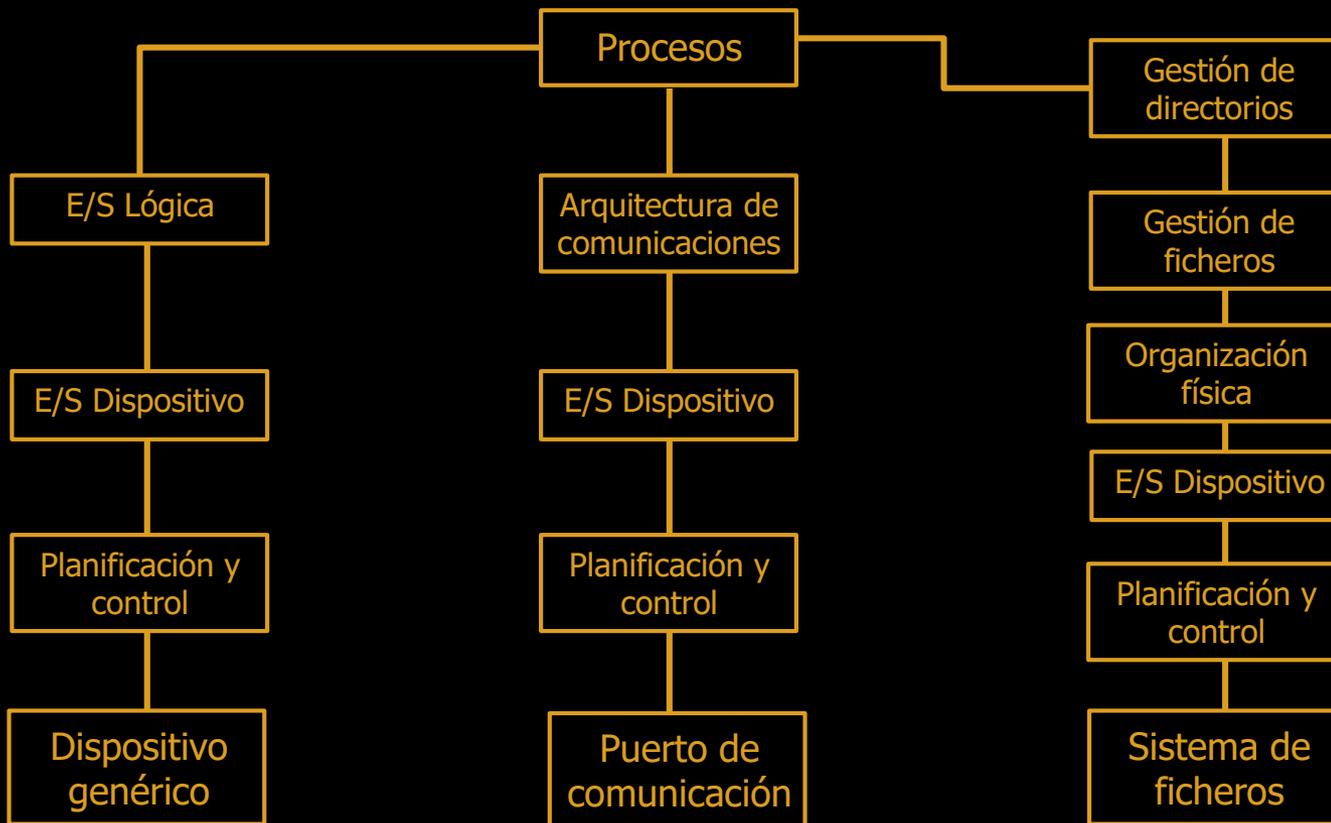
- Hay una gran variedad de dispositivos y cada uno exige un manejo distinto.
- Es preciso abstraer en capas, apiladas una encima de otra.
  - Niveles inferiores -> Menos abstracción
- Los módulos *software* que se encargan de controlar un dispositivo se denominan **drivers o manejadores**.
- Ofrecen una **interfaz común** al sistema operativo. ← **Uniformidad**
- Estructura en capas básica:
  - **Entrada/Salida lógica:** Dispositivo como recurso lógico.
    - Tareas generales: abrir, cerrar, leer, escribir...
  - **Entrada/Salida de dispositivo:** Dispositivo como ente físico.
    - Operaciones concretas de E/S, órdenes y control de datos.
  - **Planificación y control:** Gestión real del dispositivo.
    - Organización de las tareas a realizar. Control de interrupciones y estado.

# DRIVERS Y SUBSISTEMA DE ACCESO A DISPOSITIVOS



# MODELOS DE ORGANIZACIÓN

- Diferencias en la estructura de capas en función del dispositivo a manejar.



# ACCESO DIRECTO A MEMORIA

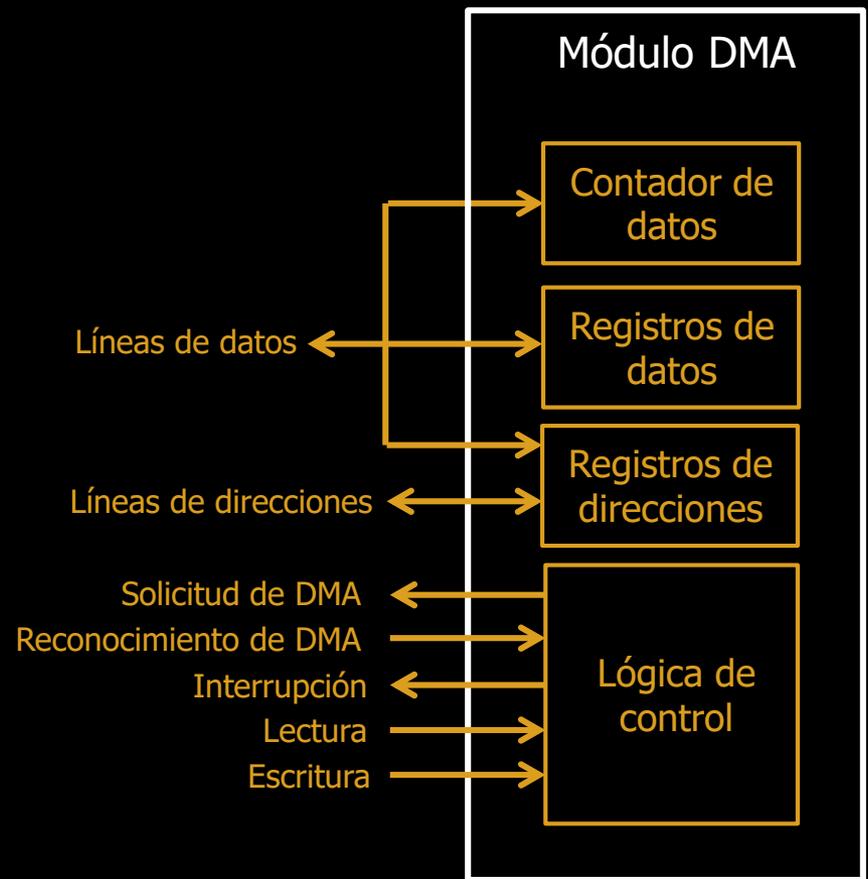
# ACCESO DIRECTO A MEMORIA

- Módulo DMA:
  - Componente separado del procesador.
  - Se encarga de las operaciones de Entrada/Salida.
  - Posee acceso directo a la memoria principal
- Descarga al procesador de la tarea del intercambio de datos.
  - Sin módulo DMA:  
El procesador supervisa cada copia de datos.
  - Con módulo DMA:  
El procesador simplemente da la orden de hacer la copia.

# FUNCIONAMIENTO DEL MÓDULO DMA (I)

Para leer/escribir datos el procesador realiza las siguientes operaciones sobre el módulo DMA.

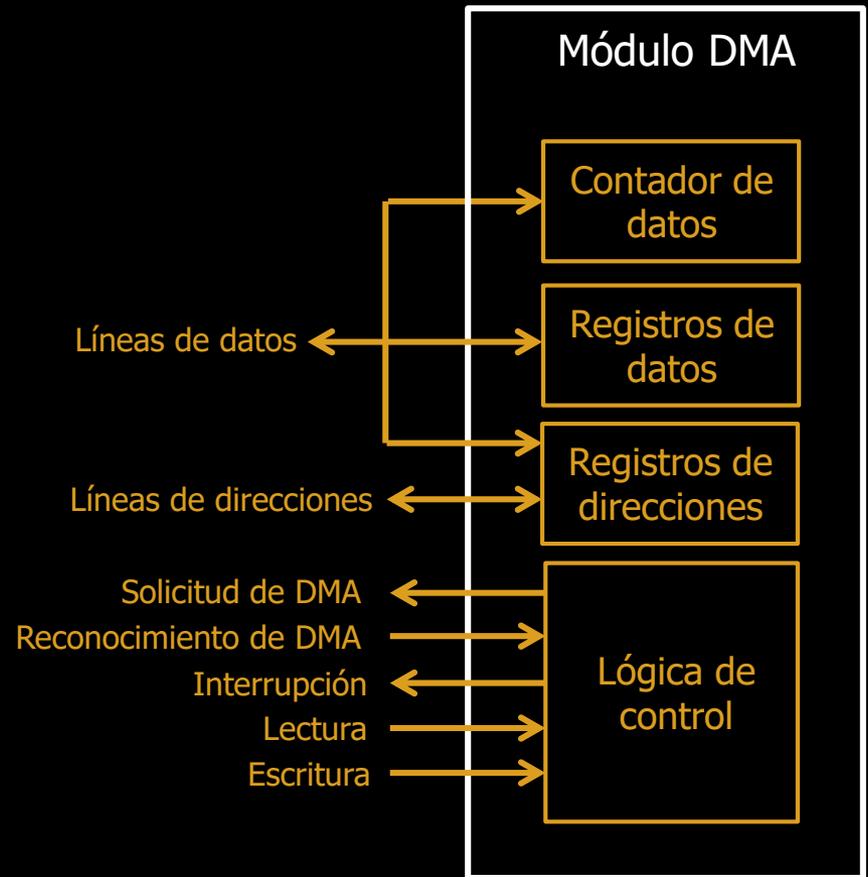
1. Marcar **lectura** o **escritura**.
2. Escribir en **registros de direcciones** usando las **líneas de datos** la dirección del dispositivo.
3. Escribir en **registros de direcciones** usando las **líneas de datos** la dirección de memoria principal.
4. Escribir en **contador de direcciones** usando las **líneas de datos** la cantidad de datos a leer/escribir.



# FUNCIONAMIENTO DEL MÓDULO DMA (II)

A continuación el módulo DMA lleva a cabo las siguientes operaciones:

1. Copia datos del/al dispositivo a/desde la memoria principal utilizando los **registros de datos**. La comunicación con la memoria principal se hace a través de las **líneas de datos** y las **líneas direcciones**.
2. Envía **interrupción**.
3. Envía **solicitud de DMA**.
4. Espera **respuesta de DMA**.



# EVOLUCIÓN HISTÓRICA

1. Entrada/Salida programada con control directo desde el procesador.
2. Entrada/Salida programada con control a través de un *driver* o manejador.
3. Entrada/Salida mediante interrupciones.
4. Gestión de la Entrada/Salida mediante DMA.
5. Gestión de la Entrada/Salida mediante DMA implementado como procesador independiente.
  - El módulo DMA puede ejecutar programas.
  - **Canal de Entrada/Salida.**
6. Gestión de la Entrada/Salida mediante DMA implementado como procesador independiente y con memoria independiente.
  - El módulo DMA reduce al mínimo los accesos a memoria principal.
  - **Procesador de Entrada/Salida**

# MODELOS DE *BUFFER*

# PROBLEMAS DE LA ENTRADA/SALIDA

- El proceso que la invoca debe esperar a que termine.
- La Entrada/Salida interfiere con las decisiones del Sistema Operativo.
  - Las posiciones de memoria involucradas no pueden modificarse durante el intercambio.
  - No se puede expulsar por completo de memoria al proceso que espera.
  - Podría producirse un interbloqueo:
    1. Un proceso hace una petición de Entrada/Salida.
    2. Es suspendido mientras espera el resultado y, por tanto, la dirección de la que debe leer/escribir el dispositivo de E/S expulsada de la memoria antes del inicio de la operación.
    3. La operación se bloquea esperando a que el proceso vuelva memoria...  
...y el proceso se bloquea esperando a que se produzca la operación.
- Una solución es la utilización de *buffers*.

# TIPOS DE *BUFFERS*

- Nomenclatura:
  - **T**: Tiempo de lectura de un bloque.
  - **P**: Tiempo entre peticiones de lectura (tiempo para procesar la lectura).
  - **M**: Tiempo de movimiento de datos del *buffer* a la memoria.
- **Sin *buffer***:
  - Tiempo de ejecución de bloque:  $T + P$
- **Buffer único**:
  - Un *buffer* en el que se almacenan los datos leídos/escritos del/al dispositivo.
  - Permite sacar al proceso de memoria.
  - Permite realizar lectura/escritura anticipada.
  - Tiempo de ejecución de bloque:  $\text{MAX}[T,P] + M$
- **Buffer doble**: Dos *buffers*.
  - Permite leer/escribir de un *buffer* mientras el SO escribe/lee en el otro.
  - Tiempo de ejecución de bloque:  $\text{MAX}[T,P]$
- **Buffer circular**: Más de dos *buffers*.
  - Modelo productor/consumidor con *buffer* acotado.

# ESQUEMAS DE *BUFFERS*

