

Otro de los sistemas más importantes de un SO es la gestión de E/S.
Vamos a realizar estos ejercicios que os permitirán tener una visión más clara.

El sistema de E/S de un SO se encarga de gestionar el envío y recepción de datos desde los periféricos.

1. ¿Qué tipos de periféricos existen según su dirección?

- **De entrada**: son periféricos que **reciben información** y la transmiten al ordenador para su procesamiento, por ejemplo: el ratón, el teclado, el escáner, etc.
- **De salida**: periféricos que **presentan la información** procesada por el ordenador, por ejemplo: la impresora, el plóter (para impresión de planos y cartografía), etc.
- **De entrada y salida**: Aúnan **ambas funciones**, por ejemplo: el monitor, el disco duro, unidad de lectura y grabación de DVD, etc.

2. ¿Cuál es la principal función que realiza el SO para tratar con las particularidades que presentan los periféricos?

El sistema operativo **abstrae de la complejidad y peculiaridad del hardware de cada periférico** para que las aplicaciones de usuario puedan hacer uso de los periféricos de una manera estandarizada y más sencilla.

3. ¿Qué elemento o componente permite desacoplar al SO de las particularidades de los periféricos?

Tanto el sistema operativo como **los fabricantes de periféricos deben estandarizar el acceso a los dispositivos utilizando** lo que se denominan controladores de dispositivos (**device drivers**).

4. Un periférico, para gestionar la comunicación con el SO se compone de 2 partes principales. Indica el nombre de estos componentes y cuál es su función.

- **Controlador (driver)**: se encarga de la **comunicación con la CPU**
Es un software, generalmente, suministrado por el fabricante del dispositivo o bien por el desarrollador del sistema operativo.
- **dispositivo** mecánico, electromecánico o electromagnético, **que se conecta al PC**

5. Uno de los principales problemas que el SO debe resolver al tratar con la E/S es la amplia variedad de rangos de velocidad que presentan los distintos periféricos.

Para solucionar este problema, ¿qué estructuras de datos se utilizan? Indica cuáles y explica en qué consiste esta estructura de datos y cómo soluciona el problema de ajustar la velocidad entre el SO y el periférico.

- **Spools:** Los datos de salida se almacenan de forma temporal en una cola situada en un dispositivo de almacenamiento masivo (spool), hasta que el dispositivo periférico requerido se encuentre libre, y así evitar que un programa quede retenido porque el periférico no esté disponible
- **Buffers:** Es para dispositivos que pueden atender peticiones de distintos orígenes. Se envían porciones de datos que el buffer retiene de forma temporal, y también se utilizan para acoplar velocidades de distintos dispositivos. De este modo, si un dispositivo lento va a recibir información más rápido de lo que puede atenderla se emplea un buffer para retener temporalmente la información hasta que el dispositivo pueda asimilarla.

6. Para comunicarse con los periféricos, el SO utiliza distintas técnicas, enuméralas y explica brevemente cada una de ellas.

a) E/S programada

- † Cuando el procesador ejecuta un programa y encuentra una instrucción relacionada con la E/S (leer, escribir o comprobar el estado de un dispositivo), ejecuta esa instrucción generando uno o varios mandatos al módulo de E/S apropiado.
- † El módulo de E/S realiza la acción solicitada pero no realiza ninguna acción para avisar al procesador.
- † Por tanto, después de invocar la instrucción E/S, el procesador debe tomar un papel activo para determinar cuándo se completa la instrucción.
- † El procesador comprueba periódicamente el estado del módulo E/S hasta que se completa la operación.
- † Aquí no se ha producido ninguna interrupción, sin embargo, el procesador ha colaborado con el módulo E/S para comunicarse en dirección al dispositivo externo y no al revés.

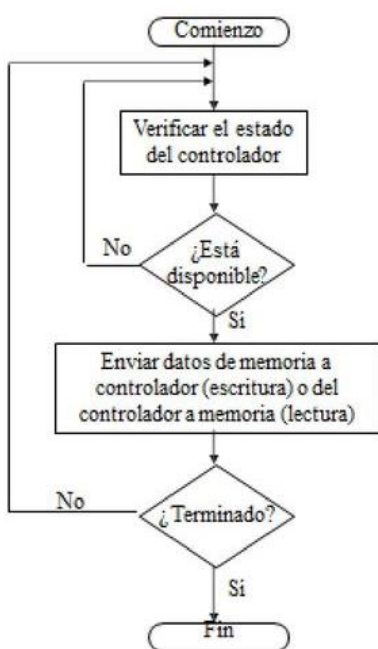
b) E/S por interrupciones

- ‡ El problema de la E/S programada es que el procesador tiene que esperar hasta que el módulo E/S correspondiente esté listo para la recepción o transmisión de más datos.
- ‡ Mientras tanto, el procesador está esperando y debe comprobar repetidamente el estado del módulo de E/S.
- ‡ A causa de esto, el rendimiento global del sistema se degrada.
- ‡ Una alternativa es que **el procesador genera un mandato de E/S y acto seguido continúe realizando algún otro trabajo útil mientras que el módulo de E/S realiza su función y se comunica con el dispositivo externo.**

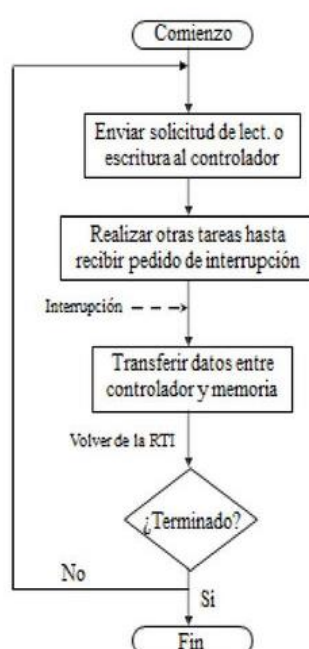
c) Acceso directo a memoria (DMA)

- ‡ La E/S dirigida por interrupciones, aunque es más eficiente que la E/S programada, igualmente requiere de la intervención activa del procesador para transferir datos entre la memoria y un módulo de E/S, ya que cualquier transferencia de datos debe atravesar un camino a través del procesador. Por lo que, ambas formas de E/S tienen dos inconvenientes:
 - 1) La tasa de transferencia de E/S está limitada por la velocidad con la que el procesador puede comprobar el estado de un dispositivo y ofrecerle servicio.
 - 2) El procesador está involucrado en la gestión de una transferencia de E/S, por lo que se deben ejecutar varias instrucciones por cada transferencia de E/S.

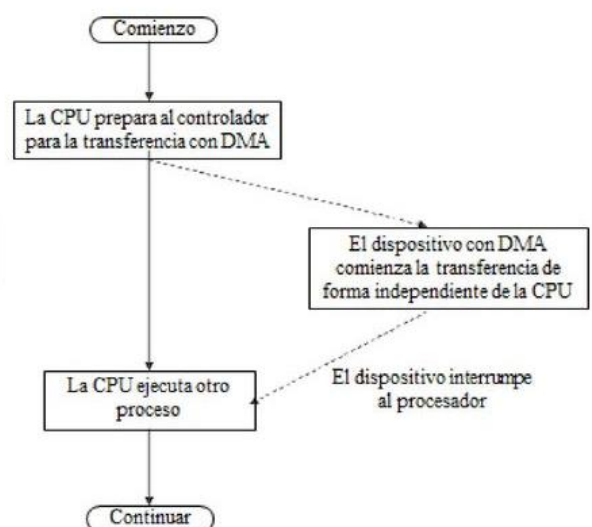
E/S programada



E/S por Interrupciones:



E/S por DMA:



7. Indica las ventajas e inconvenientes que cada una de las técnicas anteriores presenta.

	Ventajas	Inconvenientes
E/S Programada	<ul style="list-style-type: none"> ☺ Hardware mínimo ☺ Acceso a los registros: mismas instrucciones y modos de direccionamiento que para el acceso de memoria ☺ La CPU no diferencia entre el acceso a memoria y los registros de E/S 	<ul style="list-style-type: none"> ☹ Menor espacio direccionable de memoria ☹ Dificultad para controlar el acceso a los registros de E/S por parte de los programas de usuario ☹ Malgasta tiempo de proceso (interroga continuamente al periférico). ☹ Dificultad para atender varios periféricos.
E/S por Interrupciones	<ul style="list-style-type: none"> ☺ Liberan al microprocesador de la pérdida de tiempo causada por esperar a que ocurra cierto evento. ☺ Permiten la ejecución de un programa principal junto con el control simultáneo de varios dispositivos externos. ☺ Proporciona servicio prioritario a dispositivos críticos dentro del sistema. ☺ Facilitan la detección de eventos en “tiempo real”. 	<ul style="list-style-type: none"> ☹ Pueden ser necesarios circuitos externos adicionales para la generación de la señal de interrupción y la identificación del dispositivo. ☹ Dada su naturaleza asíncrona, es muy difícil probar su funcionamiento, y todavía más encontrar posibles fallas. ☹ Pueden requerir instrucciones adicionales además de las estrictamente necesarias para dar servicio al dispositivo. Por ejemplo para preservar registros o identificar al dispositivo
E/S por DMA	<ul style="list-style-type: none"> - DMA modo ráfaga: ☺ La transferencia se realiza de forma rápida - DMA modo Robo de Ciclo: ☺ No se degrada el rendimiento del sistema. 	<ul style="list-style-type: none"> - DMA modo ráfaga: ☹ Durante el tiempo que dura la transferencia, la CPU no puede utilizar el bus con memoria, lo que puede degradar el rendimiento del sistema. - DMA modo Robo de Ciclo: ☹ La transferencia tarda mucho más tiempo en llevarse a cabo