

# Tema. Gestión de Dispositivos de Entrada/Salida



## Introducción

La Función principal de un S.O. es controlar todos los dispositivos de E/S de la computadora. El Subsistema de E/S se encarga de:

- Emitir órdenes a los dispositivos
- Captar las interrupciones
- Manipular errores

Además de proporcionar una interfaz entre dispositivos y resto del sistema.

Los Sistemas Operativos controlan los dispositivos de E/S por tres razones:

- a) La interfaz de hardware para la mayoría de los periféricos es relativamente cruda (necesita software complejo para controlarlos y utilizarlos).
- b) Los periféricos son recursos compartidos (el S.O. debe protegerlos para que los accesos sean correctos y seguros).
- c) El Sistema Operativo proporciona una interfaz consistente, uniforme y flexible para todos los periféricos (permite a los usuarios referenciar a los dispositivos por el nombre y realizar operaciones de alto nivel sin necesidad de conocer la configuración de la máquina).

## Principios del hardware de E/S

### Diferencias entre dispositivos de E/S. Clasificación

Existen gran cantidad de periféricos de E/S que difieren en aspectos de:

- Velocidad de transferencia de información (por ejemplo, en un disco magnético está en torno a los 10 millones de c.p.s., mientras que en un teclado ronda los 50 c.p.s.).
- Unidad de transferencia (puede ser carácter, palabra, byte, registro, bloque, ...).
- Representación de los datos (distintas formas de codificación para diferentes soportes de E/S).
- Operaciones permitidas (distintos periféricos, distintos tipos de operaciones. Por ejemplo, una cinta magnética se puede rebobinar, pero el papel de una impresora no).
- Condiciones de error (distintos periféricos tienen distintas causas de error. Por ejemplo, error de paridad, tarjeta arrugada, error de checksum, ...).

Los periféricos de E/S se pueden clasificar en dos categorías:

- a) **Dispositivos de bloques:** Almacenan la información en bloques de tamaño fijo, cada uno con una dirección propia (permite leer, escribir o buscar un bloque sin dependencia de los demás). Por ejemplo: Disco magnético.
- b) **Dispositivos de caracteres:** Aceptan o entregan un flujo de caracteres sin considerar estructuras de bloques. No son direccionables y por tanto, no permiten operaciones de búsqueda. Por ejemplo: Impresora, terminales, cintas de papel, interfaz de redes, ...

Algunos dispositivos no encajan bien en esta clasificación. Por ejemplo: El reloj del sistema, que emite interrupciones cada cierto intervalo de tiempo.

## Controladores de dispositivos

Las unidades de E/S constan de dos partes: una mecánica y otra electrónica (controlador de dispositivo o adaptador). El S.O. casi siempre trata con los controladores y no con los dispositivos.

La comunicación entre la CPU y los controladores se realiza:

En muchos minis y micros	mediante un sistema de bus simple
En grandes computadoras	mediante buses múltiples y procesadores especializados en la E/S, llamados <i>Canales de E/S</i>

Cada controlador tiene un conjunto de registros (conocidos como *Puerto de E/S*) que se usan para la comunicación con la CPU. Básicamente:

- Registros de datos (búfer de E/S)
- Registros de órdenes
- Registros de estado

Ejemplos de puertos son los puertos paralelos, los puertos serie, los puertos USB, ...

Las direcciones de los puertos de E/S pueden:

- estar incluidas en las direcciones de memoria, ó
- estar en un espacio de direcciones aparte, llamado *Espacio de E/S*.

El S.O. realiza la E/S escribiendo comandos en los registros de los controladores.

La E/S se realiza generalmente guiada por interrupciones, según los siguientes pasos:

- El S.O. solicita una operación a un controlador.
- El controlador inicia la operación (la CPU puede asignarse a otro proceso).
- La operación del controlador finaliza.
- El controlador envía una interrupción para informar del resultado.
- El S.O. comprueba los resultados de la operación y obtiene los resultados y el estado del dispositivo leyendo uno o más bytes de información de los registros del controlador.

## Acceso Directo a Memoria (DMA, Direct Memory Access)

Muchos controladores, especialmente los de dispositivos de bloque, utilizan el Acceso Directo a Memoria (DMA). Para comprobar la utilidad del DMA, veamos cómo se produce la lectura de disco sin DMA:

- El S.O. escribe las órdenes apropiadas en los registros internos del controlador.
- El controlador lee bit a bit la información del disco y la deposita en el búfer interno del controlador.
- Se comprueba la suma de verificación (Checksum).
- El controlador genera una interrupción.
- El S.O. inicia su ejecución leyendo el contenido del búfer del controlador.

El problema que se presenta es que este último paso consume mucho tiempo de CPU.

La DMA libera a la CPU de este trabajo de bajo nivel. Al solicitar la operación, la CPU proporciona al controlador, además de la dirección del bloque en el disco, otros dos datos:

- La dirección de memoria donde se va a colocar la información.
- El número de bytes a transferir.

La transferencia la efectúa completamente el controlador. Las operaciones que realiza son las siguientes (lectura de disco con DMA):

- El controlador lee la información y la deposita en el búfer interno del controlador.
- Verifica el Checksum.
- Copia la información a partir de la dirección de la M.P. dada y tantos bytes como se le indicaron.
- El controlador genera una interrupción.

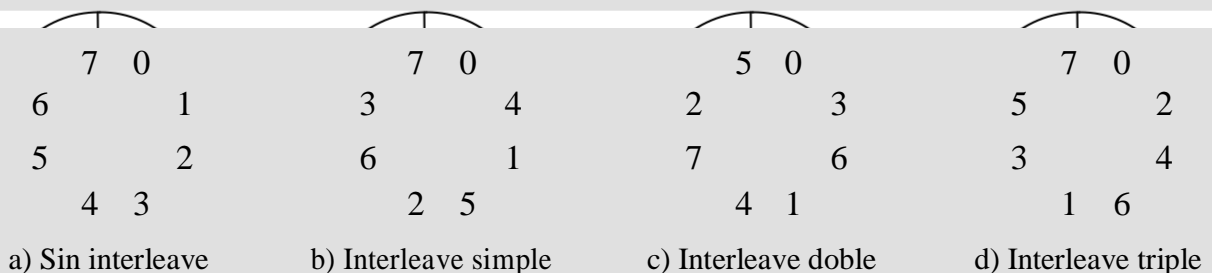
El S.O. no tiene que copiar el bloque en memoria puesto que ya lo ha hecho el controlador.

#### ***Necesidad del búfer interno del controlador:***

Mientras los datos se transfieren del controlador a la memoria, el siguiente sector estará pasando bajo la cabeza de lectura/grabación del disco y los bits estarán llegando al controlador. Los controladores sencillos no pueden copiar y realizar una operación de E/S al mismo tiempo, por lo que cuando tiene lugar una transferencia a memoria, se pierde el sector que pasa bajo la cabeza de lectura/grabación del disco.

El salto de bloques (sectores) para dar tiempo al controlador a transferir los datos a la memoria se denomina “interleave” o intercalación. Al dar formato a un disco, los bloques se numeran teniendo en cuenta el factor de intercalación, consiguiendo la mayor velocidad posible en el hardware disponible.

A continuación se muestra el orden de numeración de los sectores de un disco considerando distintos factores de interleave:



## **Principios del software de E/S**

La idea básica es organizar el software de E/S en una serie de capas de forma que podamos ocultar las peculiaridades y complejidades del hardware a los usuarios.

## Objetivos del software de E/S

- **Independencia de los dispositivos:** Debe ser posible escribir programas que se puedan utilizar con archivos (en disquete o en disco duro) o sobre otros dispositivos sin tener que modificar los programas para cada caso.
- **Uniformidad en los nombres de dispositivo:** El dispositivo se debe poder identificar mediante un número o una cadena de caracteres y no debe depender del dispositivo. Los programas harán referencia al dispositivo mediante su identificador asociado, siendo el S.O. el encargado de detectar el dispositivo real al que se está haciendo referencia.
- **Distinguir el tipo de transferencias:** síncronas (bloqueado) o asíncronas (dirigido por interrupciones). La mayoría de la E/S física es asíncrona, es decir, la CPU solicita la operación y hace cualquier otra cosa hasta que recibe la interrupción de fin de la operación solicitada. Los programas de usuario se bloquean hasta que el S.O. recibe las operaciones asociadas a la interrupción.
- **Manipulación de errores:** Los errores se deben gestionar tan cerca del hardware como sea posible. Si el controlador detecta un problema en la lectura, intentará corregirlo (por ejemplo, haciendo otra lectura); si no puede, lo hará el driver del dispositivo (por ejemplo, vuelve a leer el bloque). Sólo si las capas inferiores no pueden solucionar el error, se informará a las capas superiores (usuario).
- **Tratamiento uniforme de los periféricos:** El S.O. debe manipular todos los tipos de dispositivos evitando los problemas que pueden presentar. Por ejemplo, existen dispositivos compartidos (como el disco duro) y dedicados (como la impresora), y el S.O. debe tratarlos como tales sin que el usuario tenga constancia de estas distinciones.

Estos objetivos se pueden lograr estructurando el software de E/S en 4 capas:

1. Manipulador de interrupciones
2. Drivers de dispositivos
3. Software de E/S independiente de los dispositivos
4. Software a nivel de usuario

**NOTA:** Las características de los periféricos se codifican en los llamados descriptores de periféricos, que deben contener la siguiente información:

- Identificador del periférico
- Instrucciones que reconoce
- Punteros a tablas de traducción de caracteres
- Estado actual: ocupado, libre, averiado, ...
- Puntero al descriptor del proceso que lo está utilizando

## Manipulador de interrupciones

Las interrupciones deben ocultarse en la medida de lo posible. La forma de conseguirlo es bloqueando (con WAIT o RECEIVE) el proceso que emite el comando de E/S hasta que reciba la interrupción de fin de operación. Cuando se genera esta interrupción, se debe desbloquear (con SIGNAL o SEND) el proceso.

Cada vez que se inicia una operación de E/S, también se bloquean los drivers hasta que se termina la operación.

## Drivers de dispositivos

Un “driver” es un proceso o conjunto de instrucciones que controlan un dispositivo.

Contiene el código dependiente del dispositivo, es decir, conoce todas las peculiaridades del dispositivo.

Cada driver manipula un tipo de dispositivo o, a lo sumo, una clase de dispositivos ligeramente diferentes (como es el caso de los terminales).

Los drivers utilizan los registros del controlador de dispositivos para ejecutar un comando o comprobar que se ha ejecutado apropiadamente.

La función del driver es aceptar demandas del software independiente de los dispositivos y comprobar si la demanda se está ejecutando.

## Software de E/S independiente de los dispositivos

Parte del software de E/S es independiente del dispositivo. La frontera exacta entre los drivers y el software de E/S independiente del dispositivo depende del sistema.

Funciones del software de E/S independiente del dispositivo:

- Realizar funciones de E/S comunes a todos los dispositivos y proporcionar una interfaz uniforme al software de usuario. Estas funciones son:
  - Asignar nombres a los dispositivos y asociarlos con el driver adecuado.
  - Establecer mecanismos de protección en el acceso a los dispositivos (permiso de acceso apropiado para cada dispositivo). Esto evita que los usuarios no autorizados tengan acceso a ciertos dispositivos.
  - Suministrar un tamaño de bloque independiente del dispositivo. Es decir, ocultar la posibilidad de que diferentes discos tengan distintos tamaños de sector o las diferencias entre las unidades de transferencia.
  - “Buffering” (almacenamiento intermedio) para dispositivos de bloques y de caracteres.

- Dispositivos de bloques: El hardware trabaja a nivel de bloques (unidad de transferencia), pero los procesos de los usuarios pueden hacerlo en otras unidades (por ejemplo, archivos).
  - Dispositivos de caracteres: Los usuarios pueden escribir datos en el sistema más rápido de lo que puede gestionar el hardware, por lo que se necesita un almacenamiento intermedio.
- 
- Asignación de almacenamiento a archivos sobre dispositivos de bloque. Para ello se necesita una lista de bloques libres en el dispositivo, pero este algoritmo no debe ser gestionado por el usuario.
  - Asignación y liberación de dispositivos dedicados. Ejemplo: una impresora sólo puede ser usada por un usuario en cada momento, por lo que el S.O. debe aceptar o rechazar las peticiones de uso.
  - Generar informes de errores. Los errores están fuertemente ligados al dispositivo, por lo que los drivers deben ser capaces de resolverlos. Si no es así, debe informar al software de E/S independiente del dispositivo, que tratará el error sin depender del dispositivo, con lo que sólo informará al usuario del error o parará el sistema.

## Software de E/S del espacio de usuario

Una pequeña porción del software de E/S está fuera del S.O. y consiste en librerías que se enlazan con los programas del usuario. Las llamadas al sistema (incluyendo las llamadas al sistema de E/S) se suelen hacer dentro de procedimientos de librería.

No todo el software de E/S a nivel de usuario consta de procedimientos de librería. Otra parte importante es el sistema de *Spooling* (manejo por cola de impresión). El *Spooling* es un modo de tratar dispositivos dedicados en sistemas de multiprogramación (por ejemplo, spooling de impresora, spooling de red, ...).

### ***Ejemplo de sistema de Spooling: La impresora***

Si un proceso abre el archivo especial de caracteres para la impresora y después no hace nada, ningún otro proceso podrá imprimir nada.

El sistema de spooling crea un proceso especial (llamado “spooler” o “daemon”) y un directorio especial (llamado “directorio de spooling”). Cuando un proceso quiere imprimir un archivo, primero genera el archivo de impresión completo y lo coloca en el directorio de spooling. El spooler es el único proceso que tiene permiso para usar el archivo especial de la impresora para imprimir los archivos del directorio.

Otro ejemplo (muy similar al anterior) sería el sistema de spooling para transferencia de archivos en red. En este caso, el proceso sería el spooler de red y el directorio sería el directorio de spooling de red.

***Resumen de las capas del software de E/S y sus funciones:***

1. Procesos de usuario:
  - Realizan la llamada de E/S
  - Dan formato a la E/S
  - Gestionan la cola de impresión
2. Software de E/S independiente del dispositivo:
  - Asigna nombres, protección, bloques, búferes y drivers
3. Driver de dispositivos:
  - Coloca órdenes en los registros del dispositivo
  - Comprueba el estado del dispositivo
4. Manipulador de interrupciones:
  - “Despierta” al driver cuando se ha completado la operación de E/S
5. Hardware:
  - Realiza la operación de E/S