

Sistemas y Aplicaciones  
Informáticas

Tema 18. Sistemas Operativos:  
Gestión de Entradas/Salidas.

<b>1. ÁMBITO DE DOCENCIA.</b> .....	<b>3</b>
<b>2. ENTRADA/SALIDA. GESTOR DE E/S. FUNCIONES. OBJETIVOS.</b> .....	<b>3</b>
<b>3. HARDWARE DE ENTRADA/SALIDA</b> .....	<b>3</b>
3.1. DISPOSITIVOS DE E/S. TIPOS. CARACTERÍSTICAS. CATEGORÍAS. ....	3
3.2. CONTROLADORES. INTERFACES Y REGISTROS. TIPOS DE CONEXIÓN. FUNCIONES. ....	4
3.3. CANALES DE E/S. CONEXIÓN. TIPOS DE CANALES. ....	5
<b>4. TÉCNICAS DE GESTIÓN DE E/S.</b> .....	<b>6</b>
4.1. CONTROLADA POR PROGRAMA (SÍNCRONA). FUNCIONAMIENTO. INCONVENIENTES. ....	6
4.2. CONTROLADA POR INTERRUPCIONES (ASÍNCRONA). FUNCIONAMIENTO. TRATAMIENTO. ....	6
4.3. ACCESO DIRECTO A MEMORIA. FUNCIONAMIENTO. TIPOS DE ACCESO AL BUS. ....	7
<b>5. ESTRUCTURA DEL SISTEMA DE E/S.</b> .....	<b>8</b>
5.1. MODELO POR NIVELES. BIBLIOTECA ESTÁNDAR DE E/S. SPOOLING.....	8
5.2. BUFFERS DE E/S. FUNCIONES. TIPOS DE BUFFER. ....	8
<b>6. EJEMPLOS DE SISTEMAS OPERATIVOS</b> .....	<b>9</b>
6.1. GESTIÓN DE ENTRADA/SALIDA EN WINDOWS XP. ....	9
6.2. GESTIÓN DE ENTRADA/SALIDA EN GNU/LINUX.....	10

## 1. **Ámbito de docencia.**

- Sistemas informáticos monousuario y multiusuario (ASI 1).
- Sistemas informáticos multiusuario y en red (DAI 1).
- Sistemas operativos en entornos monousuario y multiusuario (ESI 1).

## 2. **Entrada/Salida. Gestor de E/S. Funciones. Objetivos.**

- Con el término entrada/salida (E/S) se designa cualquier transferencia de información entre los dispositivos de E/S y la memoria principal del sistema. Todas estas transferencias son ordenadas y controladas por la CPU, y permiten la comunicación del sistema con el mundo exterior.
- El gestor de E/S es la parte del sistema operativo que se ocupa de la interacción del ordenador con los dispositivos de E/S. Se encarga de lo siguiente:
  - \* Proporcionar una interfaz uniforme para el acceso a los dispositivos.
  - \* Proporcionar manejadores para los dispositivos concretos.
  - \* Tratar automáticamente los errores lo más cerca posible del dispositivo.
  - \* Utilización de memorias caché y buffers locales de dispositivo.
  - \* Planificar las peticiones de acceso a los dispositivos para permitir su compartición ordenada.
- Los objetivos a cumplir por el gestor de E/S son los siguientes:
  - \* *Independencia de dispositivo.* El sistema de E/S debe proporcionar una abstracción del hardware que permita a los procesos de usuario utilizar los diferentes dispositivos como un conjunto uniforme de recursos, ocultando los detalles de bajo nivel.
  - \* *Eficiencia.* El rendimiento global del sistema depende de la gestión de E/S, debido a que los dispositivos de E/S son lentos en comparación con la memoria principal y la CPU.
  - \* *Tratamiento de distintos tipos de dispositivos.* El gestor de E/S debe ser capaz de manejar dispositivos con diferentes características e integrarlos adecuadamente dentro del sistema.

## 3. **Hardware de entrada/salida.**

### 3.1. **Dispositivos de E/S. Tipos. Características. Categorías.**

- Los dispositivos de E/S son los elementos hardware externos con los que se comunica la CPU a través de la memoria principal. Generalmente constan de un componente mecánico (el dispositivo propiamente dicho) y un componente electrónico (el controlador de dispositivo). El sistema operativo normalmente trabaja con el controlador y no con el dispositivo directamente.
- Se pueden clasificar según su función en dispositivos de:
  - \* *Interfaz de usuario.* Apropiados para interactuar con el usuario: terminales, ratón.
  - \* *Almacenamiento.* Permiten almacenar y recuperar datos: disco duro, lectores CD-ROM.
  - \* *Comunicaciones.* Transmiten información a dispositivos remotos: modems, tarjetas de red.
- Existen una serie de características que establecen las diferencias entre los dispositivos de E/S:
  - \* *Velocidad de transferencia.* Puede haber una diferencia de varios órdenes de magnitud en las velocidades de transmisión de datos, y por tanto el tratamiento debe ser distinto.
  - \* *Tipo de transmisión.* La memoria actúa de manera sincronizada con la CPU mediante el reloj del sistema. Sin embargo, los dispositivos de E/S son variables en sus tiempos de respuesta, actuando de manera asíncrona e impredecible respecto a la CPU.

- \* *Representación de los datos.* Se emplean diferentes esquemas de codificación de datos (serie o paralelo), y diferentes códigos de caracteres y códigos de control de errores.
- \* *Modo de utilización.* Existen dispositivos de uso exclusivo (impresoras) y dispositivos que se pueden compartir (discos). Los primeros presentan problemas como el interbloqueo.
- Los dispositivos de E/S se pueden dividir en dos categorías:
  - \* *Dispositivos de bloque.* Se caracterizan por que la información se transfiere en bloques de tamaño fijo, cuyos valores más comunes van desde 128 hasta 1024 Bytes. Cada bloque tiene su propia dirección, y se puede leer o escribir en ellos de forma independiente de los demás en cualquier momento. Un ejemplo de dispositivo de bloque son los discos.
  - \* *Dispositivos de carácter.* Se caracterizan por que la información se transfiere como un flujo de caracteres, sin estructura de bloques. No existen direcciones ni tampoco se pueden realizar operaciones de búsqueda. Un ejemplo de dispositivo de carácter son los terminales.

### **3.2. Controladores. Interfaces y registros. Tipos de conexión. Funciones.**

- Un controlador de dispositivo es el componente hardware del ordenador responsable del control de uno o más dispositivos externos y del intercambio de datos entre dichos periféricos, la memoria principal y los registros de la CPU. Puede venir implementado en el propio dispositivo, integrado en la placa base o en una tarjeta controladora que se inserta en un zócalo de expansión.
- Los controladores poseen una interfaz interna para su conexión con la CPU y la memoria principal, y una interfaz externa para su conexión con el dispositivo externo:
  - \* *Interfaz interna.* Se realiza a través de un cable y un conector que normalmente cumple alguna norma estándar, como los interfaces de disco IDE y SCSI. Se encarga de convertir el flujo de bits en serie transmitidos desde el dispositivo en un bloque de bytes. Además contiene la lógica específica para cada periférico que controla.
  - \* *Interfaz externa.* Se realiza utilizando las líneas del bus del sistema:
    - **Líneas de control.** Empleadas por la CPU para enviar órdenes al controlador. El módulo de lógica de E/S del controlador interacciona con la CPU a través de estas líneas. También son utilizadas por el controlador para informar a la CPU del estado en que se encuentra y para establecer un diálogo con la CPU por el control del bus.
    - **Líneas de dirección.** Utilizadas por la CPU para referirse a un dispositivo. Están conectadas al módulo de lógica de E/S del controlador, el cual tiene una dirección asociada por cada uno de los periféricos que tiene conectados.
    - **Líneas de datos.** Empleadas para el intercambio de datos entre la CPU y el controlador.
- El controlador propiamente dicho consta de un conjunto de registros dedicados denominado puerto de E/S, que se dividen en las siguientes categorías funcionales:
  - \* *Registros de datos.* Conectados a las líneas de datos del bus del sistema, almacenan los datos hasta que la CPU o el dispositivo destino esté preparado para recibirlos.
  - \* *Registros de control.* Conectados a las líneas de control del bus del sistema, se encargan de transmitir órdenes entre la CPU y los dispositivos de E/S.
  - \* *Registros de estado.* Conectados a las líneas de datos del bus del sistema, se utilizan para indicar a la CPU el estado del dispositivo de E/S.

- La conexión del controlador con la CPU y la memoria se puede realizar de dos maneras:
  - \* *Por mapeo de memoria del sistema.* Existe un único bus entre la CPU, la memoria y el sistema de E/S con líneas de control comunes (Motorola). Los controladores forman parte del espacio normal de direcciones de la memoria y cada dispositivo tiene asignado un rango de direcciones. Son accesibles mediante instrucciones de acceso a memoria.
  - \* *A través de puertos de entrada/salida.* La CPU ve el subsistema de E/S como posiciones de memoria en un mapa de direcciones distinto al de la memoria principal (Intel). Una posición del mapa de E/S se llama puerto, que físicamente es una dirección de memoria dentro del controlador. Los puertos son accesibles mediante instrucciones de acceso a E/S, que activan la señal I/O en el bus de control para indicar que la dirección especificada es de E/S.
- Las funciones que desempeñan los controladores son las siguientes:
  - \* Control y temporización del intercambio de información entre la CPU y los dispositivos.
  - \* Comunicación entre la CPU y el dispositivo externo, que incluye:
    - Decodificación de la orden enviada por la CPU mediante el bus de control.
    - Intercambio de datos entre la CPU y el controlador mediante el bus de datos.
    - Información sobre el estado del controlador (ocupado o preparado).
    - Reconocimiento de la dirección, única para cada periférico que maneja el controlador.
  - \* Almacenamiento temporal de datos para adecuar las diferencias de velocidad entre la interfaz interna (memoria principal y CPU) y la externa (dispositivo).
  - \* Detección y corrección de errores, e información de los mismos a la CPU.

### **3.3. Canales de E/S. Conexión. Tipos de canales.**

- Un canal de E/S es un microprocesador especializado en ejecutar instrucciones de E/S almacenadas en la memoria principal, liberando a la CPU de esta tarea. El canal de E/S controla la transferencia de datos e informa a la CPU al terminar.
- Cuando la CPU debe realizar una operación de E/S, envía una instrucción especial que indica la dirección donde se encuentra el programa a ejecutar, en el que se especifica lo siguiente:
  - \* Los periféricos que intervienen en la operación de E/S.
  - \* La zona de memoria y las prioridades utilizadas en la operación.
  - \* Las acciones a efectuar si se producen ciertas condiciones de error durante la operación.
- El sistema de E/S se comunica con la CPU y con la memoria principal mediante dos buses independientes. La memoria se comunica con la CPU y con el canal por el bus del sistema, mientras que los dispositivos de E/S se comunican con el canal a través de un bus de E/S.
- Hay dos tipos comunes de canales de E/S:
  - \* *Canal selector.* Controla múltiples dispositivos de alta velocidad. En cualquier instante de tiempo está dedicado a la transferencia de datos con uno sólo de estos dispositivos. El canal sustituye a la CPU en la supervisión del funcionamiento de los controladores de E/S.
  - \* *Canal multiplexor.* Puede controlar de forma simultánea operaciones de E/S con múltiples dispositivos. Un multiplexor orientado a la transferencia de bytes transmite caracteres hacia o desde los dispositivos de carácter, mientras que un multiplexor orientado a la transferencia de bloques alterna bloques de datos de este tipo de dispositivos.

## 4. Técnicas de gestión de E/S.

### 4.1. Controlada por programa (síncrona). Funcionamiento. Inconvenientes.

- El funcionamiento se basa en la *transferencia síncrona* de datos entre la CPU y el controlador:
  - \* Cuando la CPU ejecuta una instrucción de E/S en un proceso, envía una orden al controlador de dispositivo correspondiente y espera hasta que finalice la operación de E/S.
  - \* El controlador lleva a cabo la operación requerida, y una vez finalizada activa los bits apropiados en su registro de estado. La CPU debe comprobar periódicamente mediante un bucle el estado del controlador, hasta que esté preparado para la siguiente operación.
  - \* La CPU es la responsable de transferir los datos entre la memoria principal y el controlador de dispositivo a través del bus de datos en las operaciones de E/S.
- Como consecuencia, la CPU debe tener control directo sobre la operación de E/S, para lo cual debe contar con instrucciones de E/S para el control de los dispositivos, la comprobación de su estado y la transferencia de los datos entre la memoria y los periféricos.
- Los inconvenientes de este tipo de tratamiento son los siguientes:
  - \* Degradación del rendimiento del sistema por la pérdida de tiempo en el bucle de espera.
  - \* Atención exclusiva a un periférico durante el bucle de espera en detrimento de los demás.

### 4.2. Controlada por interrupciones (asíncrona). Funcionamiento. Tratamiento.

- El funcionamiento se basa en la *transferencia asíncrona* de datos entre la CPU y el controlador:
  - \* Cuando la CPU ejecuta una instrucción de E/S en un proceso, envía una orden al controlador de dispositivo correspondiente y continúa ejecutando el proceso, si no precisa esperar al final de la operación. En otro caso, se suspende y se continúa con otro proceso.
  - \* El controlador lleva a cabo la operación requerida, y una vez finalizada activa la línea de petición de interrupción del bus de control, que fuerza en la CPU una interrupción en el proceso que está ejecutando. La CPU guarda el contexto de trabajo, realiza la transferencia de datos y reanuda la ejecución del proceso en el punto donde fue interrumpido.
  - \* La CPU es la responsable de transferir los datos entre la memoria principal y el controlador de dispositivo a través del bus de datos en las operaciones de E/S.
- El tratamiento de las interrupciones se realiza de la siguiente manera:
  - \* Después de cada instrucción la CPU verifica la línea de interrupción. Si se encuentra activa indica que se ha producido una interrupción. Las IRQ (Interrupt ReQuest) son líneas que llegan al controlador de interrupciones, un componente hardware que habilita o inhibe líneas de interrupción, y establece prioridades entre las distintas interrupciones habilitadas.
  - \* Una CPU sin controlador de interrupciones integrado suele tener una única línea de interrupción llamada INT, que es activada por el controlador de interrupciones cuando tiene una interrupción que servir. Al activarse esta línea, el procesador completa la ejecución de la instrucción en curso y guarda el estado del proceso en la pila.
  - \* Después la CPU consulta los registros del controlador de interrupciones para averiguar qué interrupción debe atender. A partir del número de IRQ busca en el vector de interrupciones qué rutina debe llamar para atender la petición del dispositivo asociado a dicha IRQ.

- \* El vector de interrupciones es un vector que contiene el valor que apunta a la dirección en memoria de la rutina servidora de interrupción. Esta rutina se encarga de realizar la transferencia de datos solicitada entre la memoria principal y el controlador de dispositivo.
- \* Una vez finalizada la rutina servidora de interrupción, el procesador restaura el estado del proceso interrumpido y vuelve al punto anterior a la interrupción.

#### **4.3. Acceso directo a memoria. Funcionamiento. Tipos de acceso al bus.**

- La E/S por programa y la E/S por interrupciones requieren la participación activa de la CPU para transferir datos entre la memoria y el controlador. El acceso directo a memoria (DMA) es una técnica que permite la transferencia directa de información sin intervención de la CPU.
- Requiere un módulo hardware conectado al bus del sistema, el controlador de DMA, que consta de una serie de registros que le permiten transferir datos desde o hacia memoria:
  - \* *Registro de datos.* Almacena las palabras de datos a enviar o recibir.
  - \* *Registro de dirección.* Se utiliza para almacenar la dirección de la siguiente palabra que se va a enviar o recibir, y se incrementa de forma automática después de cada transferencia.
  - \* *Registro contador de palabras.* Almacena el número de palabras que quedan por enviar o recibir, y se decrementa automáticamente después de cada transferencia.
- El funcionamiento del DMA es el siguiente:
  - \* Cuando la CPU desea leer o escribir un bloque de datos emite una orden al controlador de DMA, indicándole si la operación es de lectura o escritura, la dirección del dispositivo, la posición inicial de memoria de donde hay que leer o donde hay que escribir, y el número de palabras de datos que se tienen que leer o escribir.
  - \* A partir de este momento la CPU continúa realizando otra tarea. Cuando el controlador de DMA está preparado para transmitir o recibir datos, activa la línea de petición de DMA a la CPU. Ésta renuncia al control de los buses de datos y direcciones y activa la línea de reconocimiento de DMA parando su actividad, aunque *no se trata de una interrupción*.
  - \* El controlador de DMA transfiere directamente palabra a palabra el bloque completo de datos entre el dispositivo y la memoria. Cuando la transferencia finaliza, el controlador de DMA envía una señal de interrupción a la CPU. Por tanto, la CPU únicamente participa al comienzo y al final de la transferencia.
- Existen diferentes tipos de acceso al bus por parte del controlador de DMA:
  - \* *Por ráfagas.* Cuando toma el control del bus, no lo libera hasta finalizar la transmisión. Se consigue la mayor velocidad de transferencia, pero se tiene inactiva la CPU.
  - \* *Por robo de ciclos.* Toma el bus durante un ciclo enviando una palabra cada vez. Reduce al máximo la velocidad de transferencia y la interferencia del DMA sobre la CPU.
  - \* *DMA transparente.* Se aprovechan los ciclos en los que la CPU no utiliza el bus. No existe interferencia entre el DMA y la CPU, pero no se obtiene una velocidad muy elevada.
  - \* *Por demanda.* Comienza la transferencia, devolviendo el control a la CPU cuando no tiene más datos disponibles. Cuando dispone de nuevos datos, retoma el control del bus.
  - \* *Dato a dato.* Cada vez que el dispositivo solicita una transferencia, se envía un único dato y se devuelve el control a la CPU. El proceso acaba cuando se ha transferido todo el bloque.

## 5. Estructura del sistema de E/S.

### 5.1. Modelo por niveles. Biblioteca estándar de E/S. Spooling.

- El sistema de E/S utiliza un modelo conceptual por niveles, de manera que:
  - \* Cada nivel realiza un subconjunto de las funciones de E/S que necesita el sistema operativo.
  - \* Cada nivel descansa sobre el siguiente nivel inferior para efectuar operaciones más básicas.
  - \* Cada nivel esconde los detalles a los niveles superiores a los que proporciona servicios.
  - \* Los niveles se definen de forma que los cambios que se realicen en un determinado nivel no requieran modificación alguna en los otros niveles.
- Los niveles del sistema de E/S por orden ascendente respecto al hardware son los siguientes:
  - \* *Manejadores de interrupciones*. Son dependientes del hardware. Se encargan de atender las interrupciones solicitadas por los dispositivos, realizando primero las acciones más urgentes (top half) con el resto de interrupciones deshabilitadas, y retrasando las menos urgentes (bottom half). Para ello salvan los registros del proceso en ejecución, ejecutan las rutinas servidoras de interrupción correspondientes y devuelven el control al proceso interrumpido.
  - \* *Manejadores de dispositivos*. Son dependientes del hardware. Se encargan de escribir las órdenes oportunas en los registros de los controladores de dispositivo, partiendo de las solicitudes del software independiente de dispositivo, y de verificar su correcta ejecución. Si al recibir una solicitud el manejador de dispositivo está ocupado con otra, agregará la nueva solicitud a una cola de solicitudes pendientes.
  - \* *Software independiente de dispositivo*. Sirve de interfase entre el software de usuario y los manejadores de dispositivos. Trata los dispositivos como recursos lógicos mediante identificadores y órdenes simples (abrir, cerrar, leer, escribir). Realiza las funciones de E/S comunes a todos los dispositivos, como la información sobre errores, la asignación y liberación de los dispositivos, la protección de los dispositivos y el uso de sus buffers, su asociación con los manejadores de dispositivo y la asignación del tamaño de los bloques.
  - \* *Software de usuario*. Existen dos tipos de software de E/S a nivel de usuario:
    - **Biblioteca estándar de E/S**. Una parte del software de E/S no está dentro del sistema operativo, se trata de bibliotecas que contienen varios procedimientos que están a disposición de los usuarios para formar parte de sus programas.
    - **Spooling**. Es una técnica que permite utilizar un sistema de almacenamiento auxiliar (disco) para recoger órdenes relacionadas con dispositivos de baja velocidad y uso exclusivo (impresoras) y ejecutarlas a continuación por un demonio o spooler. De esta manera se evita que un proceso mantenga un recurso ocupado durante mucho tiempo.

### 5.2. Buffers de E/S. Funciones. Tipos de buffer.

- Un buffer de E/S es un espacio de memoria principal que se reserva para el almacenamiento intermedio de datos procedentes o con destino a los dispositivos. Sirve para:
  - \* Compensar la diferencia de velocidad de CPU y memoria respecto a los dispositivos de E/S.
  - \* Separar las transferencias de E/S del espacio de direcciones de los procesos, lo que facilita al sistema operativo el intercambio de los procesos suspendidos.



- Existen dos tipos de buffer de E/S:
  - \* *Buffer simple*. Cuando un proceso de usuario realiza una petición de E/S, el sistema operativo le asigna a la operación un buffer en la memoria principal:
    - Para los **dispositivos de bloque**, en las operaciones de entrada se transfiere un bloque desde el controlador de dispositivo al buffer. Cuando finaliza la transferencia, el sistema operativo mueve el bloque desde el buffer al espacio de direcciones del proceso, e inmediatamente solicita otro bloque. En operaciones de salida, el sistema operativo copia el bloque que se desea transferir desde el espacio de direcciones del proceso al buffer, y de éste al controlador de dispositivo.
    - Para los **dispositivos de carácter** se utilizan colas en memoria para almacenar los caracteres con destino o con origen en un dispositivo.
  - \* *Buffer doble*. Se puede realizar una mejora del buffer simple asignando dos buffers a cada operación. De esta forma, el sistema operativo puede transferir datos hacia o desde un buffer mientras que vacía o rellena el otro.

## 6. Ejemplos de sistemas operativos.

### 6.1. Gestión de entrada/salida en Windows XP.

- La estructura básica del sistema de E/S está formada por una serie de rutinas independientes de dispositivo para ciertos aspectos de la E/S y por un conjunto de manejadores de dispositivo para comunicarse con los dispositivos. Estos manejadores de dispositivo están definidos de acuerdo con Windows Driver Model, una norma definida para la compatibilidad entre sistemas Windows.
- El gestor plug and play permite que los dispositivos compatibles sean inmediatamente reconocidos una vez conectados. Para ello, el gestor pide al dispositivo que se identifique, y con esta información carga en memoria el manejador de dispositivo apropiado.
- La entrada/salida de dispositivos se maneja a través de las APIs del subsistema Win32, el cual se relaciona con el hardware a través de la capa de abstracción de hardware (HAL). Esta capa es una interfaz entre el hardware y el resto del sistema operativo que se caracteriza por lo siguiente:
  - \* Está implementada como una biblioteca de enlace dinámico (dll) y es responsable de proteger y aislar el resto del sistema de las especificaciones del hardware.
  - \* Presenta los dispositivos al sistema operativo de manera homogénea a través de un conjunto de funciones bien definidas. Estas funciones son llamadas tanto desde el sistema operativo como desde los propios manejadores de dispositivo, permitiendo a éstos adaptarse a distintas arquitecturas de E/S sin tener que ser modificados en gran medida.
  - \* Si Windows es portado a una nueva arquitectura de procesador el HAL debe ser reescrito, pero el resto del sistema simplemente debe ser recompilado.
  - \* También suministra la interfaz para el multiprocesamiento simétrico, transformando cada procesador físico en un procesador virtual idéntico al resto de procesadores.
  - \* Al HAL sólo pueden acceder componentes del Executive de Windows y nunca los programas en modo usuario. El HAL también intenta ser el único software dentro del sistema que se comunica con el hardware, aunque existe un pequeño número de llamadas de los manejadores de dispositivo y del núcleo que interactúan directamente con el hardware.

## 6.2. Gestión de entrada/salida en GNU/Linux.

- Cada dispositivo de E/S tiene asociado un archivo especial en el directorio /dev gestionado por el sistema de archivos, que se lee y se escribe igual que el resto de archivos. Para acceder a un dispositivo, basta con realizar peticiones de lectura o escritura a su archivo especial asociado.
- Los archivos especiales se dividen en:
  - \* *Archivos de bloque.* Están formados por una secuencia de bloques numerados, y se caracterizan por que es posible acceder a cada bloque por su dirección.
  - \* *Archivos de carácter.* Se utilizan para dispositivos que producen o reciben flujos de caracteres, como teclados, impresoras, ratones, tarjetas de red, etc.
- Cada archivo especial tiene asociado un manejador de dispositivo, el cual tiene un número principal que sirve para identificarlo. Si un manejador de dispositivo puede manipular varios dispositivos, cada dispositivo tiene un número secundario que le identifica. Por tanto, cada pareja de números principal y secundario identifica a un determinado dispositivo.
- La E/S se implementa mediante una colección de manejadores de dispositivo, uno por cada tipo de dispositivo. Su función consiste en aislar al resto del sistema de las peculiaridades del hardware. Cuando un proceso de usuario accede a un archivo especial, el sistema determina los números principal y secundario, que son utilizados como parámetros para acceder a las rutinas de manejo del dispositivo correspondiente.
- Los manejadores de dispositivo pueden estar incluidos estáticamente dentro del núcleo o incluidos en módulos que pueden ser cargados dentro del núcleo en tiempo de ejecución.
- La entrada/salida puede realizarse de dos maneras:
  - \* *Utilizando DMA.* Es el método más rápido, pero no permite que los procesos suspendidos por la operación de E/S puedan ser intercambiados. Además, el dispositivo se paraliza junto al proceso mientras dure la transferencia, quedando inasequible para otros procesos.
  - \* *Utilizando los buffers del sistema.* Se pueden usar dos clases de buffers:
    - **Caché de disco.** Adecuada para dispositivos de bloque. Se trata de un conjunto fijo de bloques recientemente leídos que se guardan en memoria para aumentar el rendimiento. Si un bloque no está en la caché de disco, se lee del disco hacia la caché y de ahí a la memoria principal, y viceversa en el caso de escritura. Para cargar un bloque en una caché totalmente ocupada se plantea una situación muy parecida a la paginación, y se resuelve con algoritmos similares. La transferencia de datos entre la caché de disco y el espacio de usuario del proceso se produce mediante DMA. Ante la posibilidad de un fallo del sistema, es necesario que los bloques críticos que hayan sido modificados permanezcan el menor tiempo posible en la caché y se transfieran al disco, ya sea a intervalos regulares o justo después de modificarse la caché.
    - **Cola de caracteres.** Adecuada para dispositivos de carácter. El dispositivo de E/S escribe en una cola de caracteres de la que lee el proceso, o también, el proceso escribe y el dispositivo lee de ella. De esta manera, las colas de caracteres sólo pueden ser leídas una vez. A medida que se lee cada carácter, éste es destruido. Este mecanismo es distinto al de la caché de buffers, donde se puede leer varias veces.