Sistemas Operativos

Manejo de Entrada/Salida

Armando Arce, Escuela de Computación, arce@itcr.ac.cr

Tecnológico de Costa Rica

Sistemas de Entrada/Salida

El sistema operativo debe administrar la entrada/salida.

- Esto lo hace enviando comandos a los dispositivos, atendiendo interrupciones y tratando los errores.
- También debe proporcionar una interfaz entre los dispositivos y el resto del sistema que sea sencilla y fácil de utilizar.
- Hasta donde sea posible, la interfaz debe ser la misma para todos los dispositivos (independencia del dispositivo).
- El código de E/S representa una fracción significativamente grande del sistema operativo completo.

Hardware de Entrada/Salida

Por lo general, las unidades de E/S consisten de un componente mecánico y un componente electrónico.

- El componente electrónico es llamado controlador de dispositivo, comúnmente tiene la forma de un chip en la tarjeta principal o como una tarjeta controladora independiente.
- El componente mecánico es el dispositivo en sí.

Hardware de Entrada/Salida

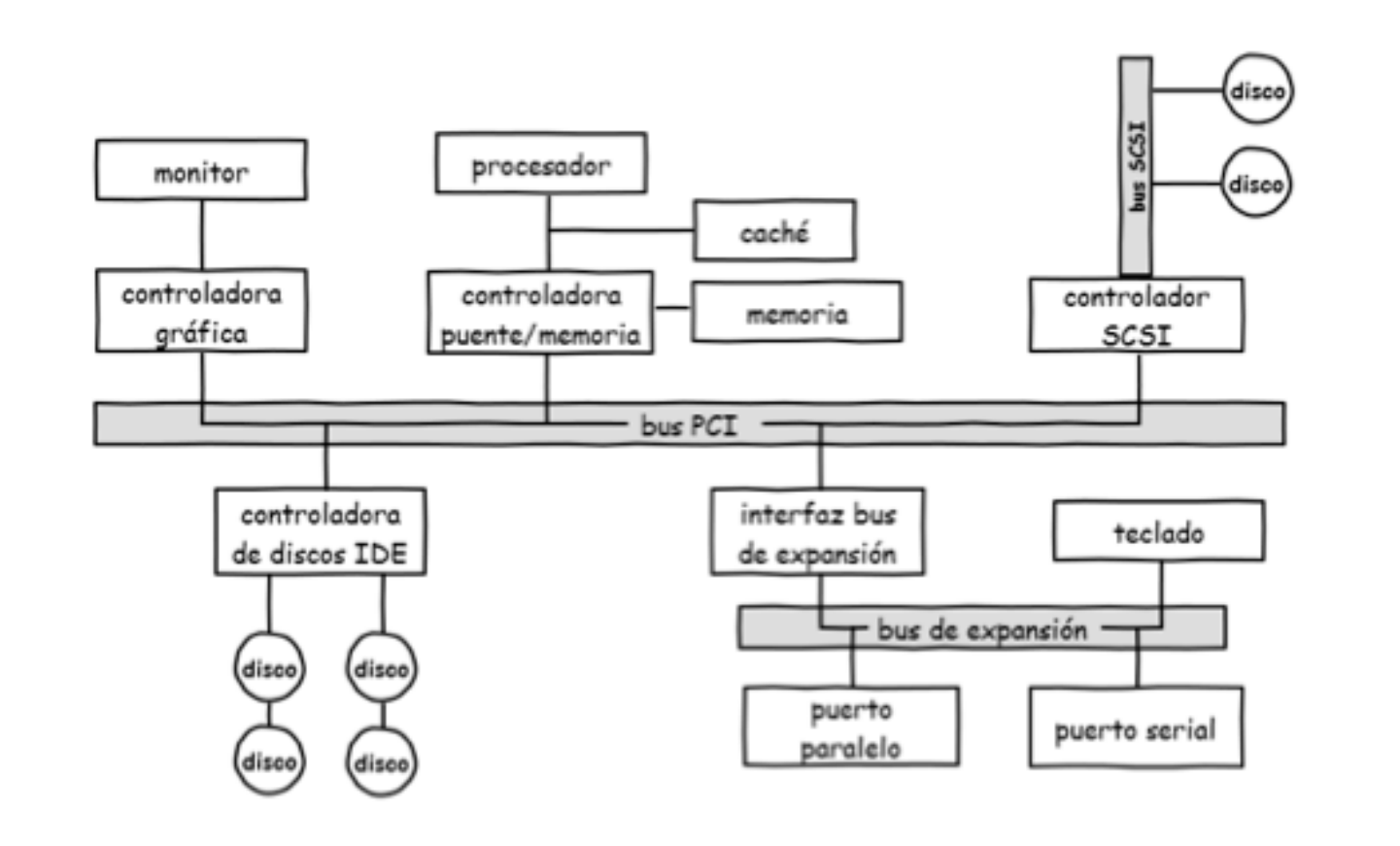

Figure 1:

Hardware de Entrada/Salida

La tarjeta controladora por lo general contiene un conector, en el que se puede conectar un cable que conduce al dispositivo en sí.

- Muchos controladores pueden manejar dos, cuatro o inclusive ocho dispositivos idénticos.
- Entre el controlador y el dispositivo se transfieren flujos de bits, que incluyen códigos de comprobación.
- El trabajo del controlador es convertir el flujo de bits en un bloque de bytes y realizar cualquier corrección de errores.
- Por lo general, primero se ensambla el bloque de bytes, bit por bit, en un búfer dentro del controlador.
- Después de haber verificado su suma de comprobación y de que el bloque se haya declarado libre de errores, puede copiarse a la memoria principal.

Modos de Entrada/Salida

Cada controlador tiene unos registros que se utilizan para comunicarse con la CPU.

- Al escribir en ellos, el sistema operativo puede hacer que el dispositivo envíe o acepte datos, se encienda o se apague, o realice cualquier otra acción.
- Al leer de estos registros, el sistema operativo puede conocer el estado del dispositivo, si está preparando o no para aceptar un nuevo comando.

Modos de Entrada/Salida

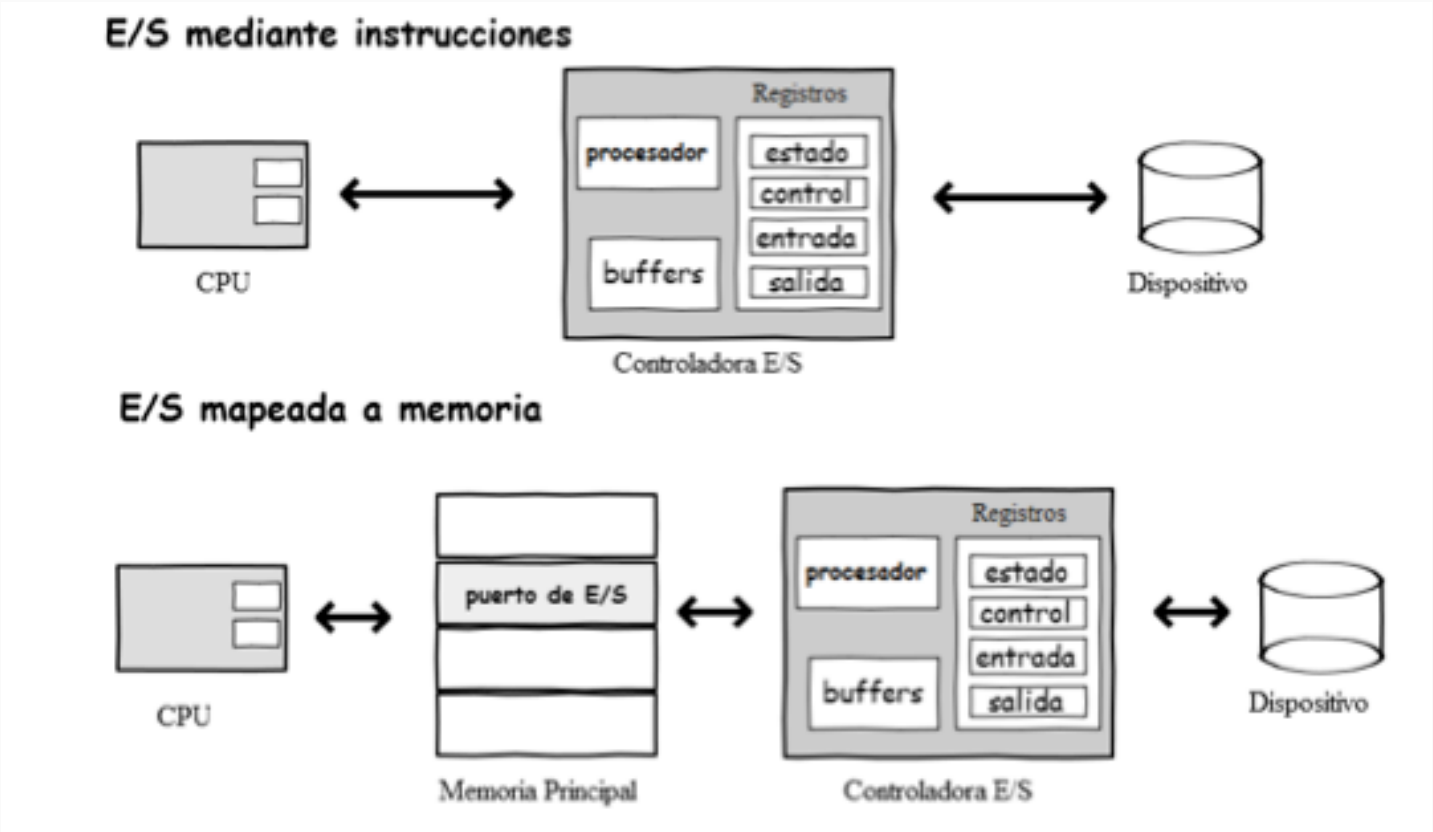

Figure 2:

Modos de Entrada/Salida

Además de los registros de control, muchos dispositivos tienen un búfer de datos que el sistema operativo puede leer y escribir.

- Existen dos alternativas para comunicar el CPU con los registros de control y los búferes de datos de los dispositivos: E/S mediante instrucciones y E/S mapeada memoria.

Entrada/Salida mediante instrucciones

Este mecanismo también se conoce como E/S mediante puertos.

- A cada registro de control se le asigna un puerto de E/S.
- El conjunto de todos los puertos de E/S forma el espacio de puertos de E/S y está protegido de manera que los programas de usuario ordinarios no pueden utilizarlo.
- Mediante el uso de instrucciones especiales de E/S (p.ej. IN y OUT) la CPU puede leer/escribir los registros de control y almacenar/transferir el resultado en/desde registros de la CPU.

Entrada/Salida mapeada a memoria

Consiste en asignar todos los registros de control al espacio de memoria.

- A cada registro de control se le asigna una dirección de memoria única a la cual no hay memoria asignada.
- Por lo general, las direcciones asignadas se encuentran en la parte superior del espacio de direcciones.

Manejo de Entrada/Salida

Hay dos formas fundamentalmente distintas en que se puede llevar a cabo la E/S: programada y controlada por interrupciones.

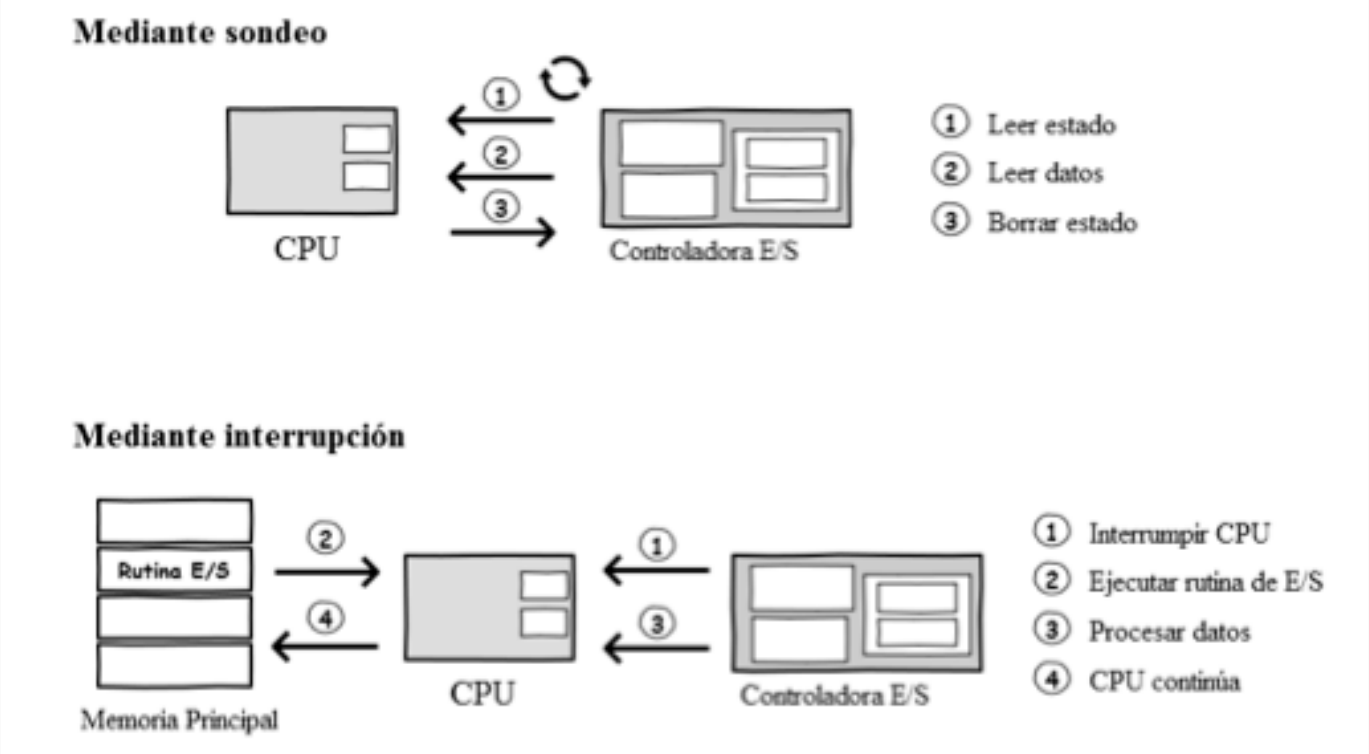

Figure 3:

Programada (Sondeo)

Para coordinar la relación productor-consumidor entre la controladora y el CPU se realiza lo siguiente.

- La controladora indica su estado mediante el bit de ocupado en el registro de estado.
- La controladora activa el bit de ocupado cuando está ocupada trabajando y borra el bit de ocupado cuando está lista para aceptar el siguiente comando.
- El CPU indica sus deseos mediante el bit de comando preparado en el registro de comando: el CPU activa el bit de comando preparado cuando ha disponible un comando para que la controladora lo ejecute.
- El CPU se encuentra siempre en un estado de espera activa o sondeo: ejecuta un ciclo, leyendo una y otra vez el registro de estado hasta que el bit de ocupada se desactiva.

Interrupciones

Cuando la CPU detecta que una controladora ha activado una señal a través de una línea de solicitud de interrupción, la CPU guarda el estado actual y salta a la rutina de tratamiento de interrupciones situada en una dirección fija en memoria.

- La rutina de tratamiento de interrupciones determina la causa de la interrupción, lleva a cabo el procesamiento necesario, realiza la restauración del estado y ejecuta una instrucción especial para volver a situar la CPU en el estado de ejecución anterior a que se produjera la interrupción.

Acceso directo a memoria

La mayoría de la computadoras evitan sobrecargar la CPU con las tareas de E/S, descargando parte de este trabajo en un procesador de propósito especial denominado DMA.

- Para iniciar una transferencia de DMA, el host describe un bloque de comando DMA en la memoria.
- Este bloque contiene un puntero al origen de una transferencia, un puntero de destino de la transferencia y un contador del número de bytes que hay que transferir.
- La CPU escribe la dirección de este bloque de comandos en la controladora DMA y luego continúa realizando otras tareas.
- La controlador DMA será encarga entonces de operar el bus de memoria directamente, colocando direcciones en el bus para realizar las transferencias sin ayuda de la CPU principal.

Acceso directo a memoria

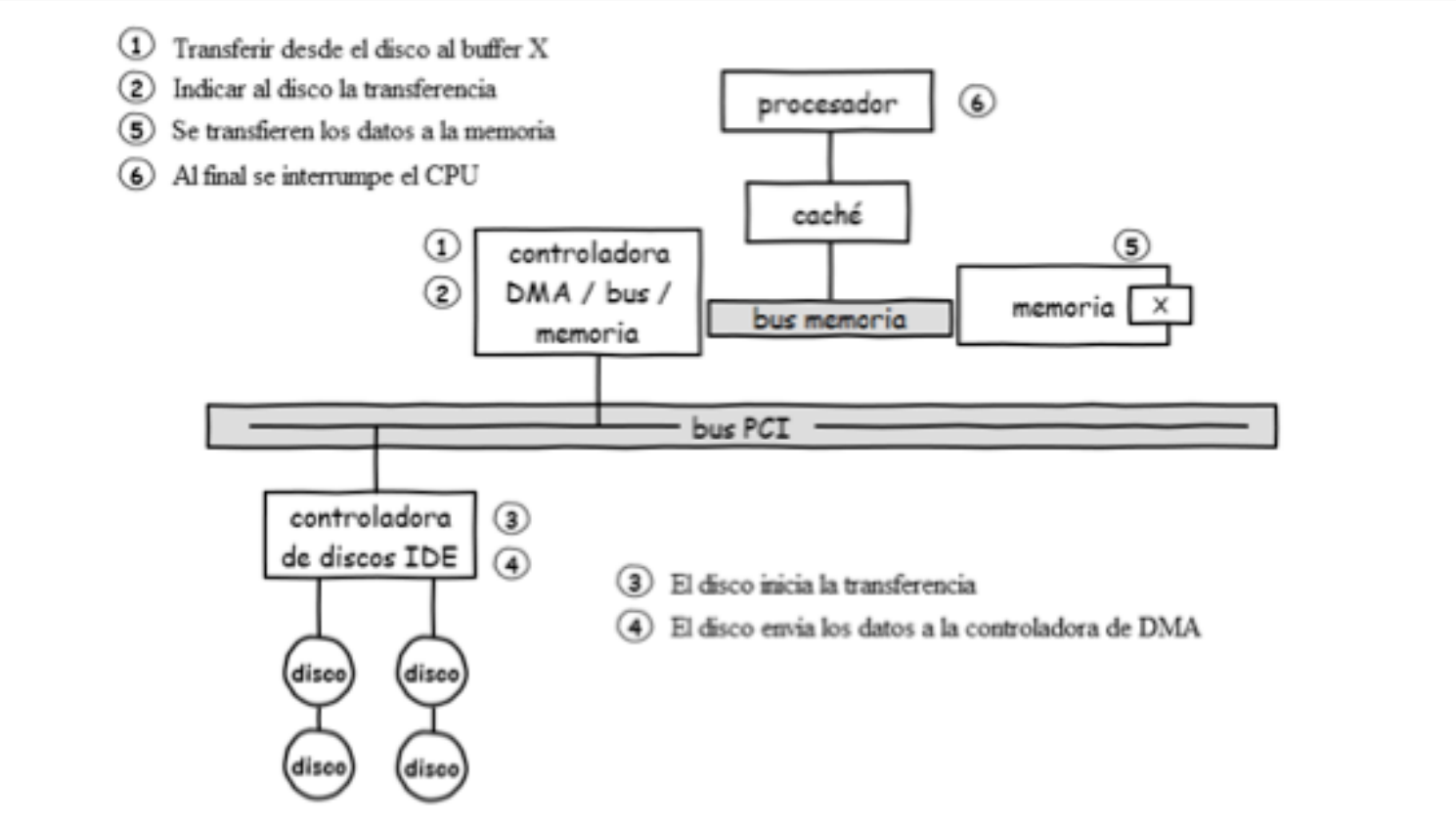

Figure 4:

Acceso directo a memoria

Descargar el trabajo de transferencia de datos en una controladora DMA suele mejorar el rendimiento global del sistema.

- También, el DMA puede realizar una transferencia entre dos dispositivos mapeados en memoria sin la intervención de la CPU y sin usar la memoria principal.

Software de Entrada/Salida

Por lo general el software de E/S se organiza en cuatro capas:

- Software de E/S de capa de usuario
- Software del S.O. independiente del dispositivo
- Manejadores de dispositivo
- Manejadores de interrupciones
- Hardware de E/S

Cada capa tiene una función bien definida que realizar, y una interfaz bien definida para los niveles adyacentes.

Software de Entrada/Salida

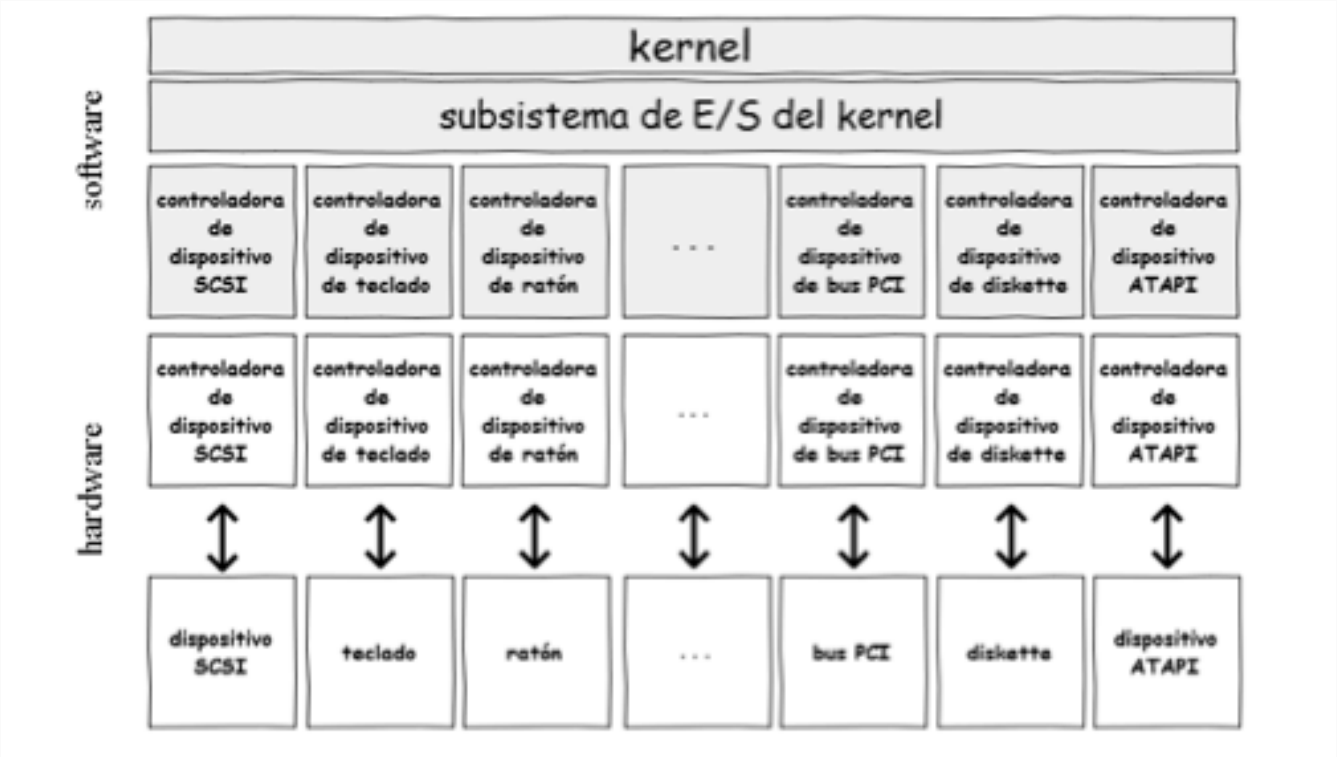

Figure 5:

Manejadores de dispositivo (drivers)

Cada dispositivo de E/S necesita cierto código especifico para controlarlo.

- Este código conocido como el manejador (driver), es escrito por el fabricante y se incluye junto con el mismo.
- Como cada sistema operativo necesita sus propios manejadores, los fabricantes por lo común los proporcionan para varios sistemas operativos.

Manejadores de dispositivo (drivers)

Un manejador de dispositivo tiene varias funciones.

- La más obvia es aceptar peticiones abstractas de lectura y escritura de la capa de software independiente del dispositivo que está por encima de él, y ver que se lleven a cabo.
- Pero también hay otras tantas funciones que deben realizar.
- Por ejemplo, el manejador debe inicializar el dispositivo, si es necesario.
- También puede tener que administrar sus propios requerimientos y eventos del registro.

Software de E/S independiente del dispositivo

Aunque parte del software de E/S es específico para cada dispositivo, otras partes son independientes del dispositivo.

- El límite exacto entre los manejadores y el software independiente del dispositivo depende del sistema y del dispositivo.
- La función básica del software independiente del dispositivo es realizar las funciones de E/S que son comunes para todos los dispositivos y proveer una interfaz uniforme para el software a nivel de usuario.

Interfaz uniforme con los manejadores de dispositivo

Para cada clase de dispositivos, el sistema operativo define un conjunto de funciones que el manejador debe proporcionar.

- Para un disco, estas funciones incluyen naturalmente la lectura y la escritura, pero también encender y apagar la unidad, aplicar formato y otras cosas relacionadas con los discos.

Interfaz uniforme con los manejadores de dispositivo

A menudo, el manejador contiene una tabla con apuntadores a sí mismo para dichas funciones de control.

- Cuando se carga el manejador, el sistema operativo registra la dirección de esta tabla de apuntadores, por lo que cuando necesita llamar a una de las funciones, puede hacer una llamada indirecta a través de esta tabla.
- Esta tabla de apuntadores define la interfaz entre el manejador y el resto del sistema operativo.
- Todos los dispositivos de una clase dada (discos, impresoras, etc.) deben proporcionarla.

Interfaz uniforme con los manejadores de dispositivo

Contar con una interfaz uniforme facilita conectar un nuevo manejador, siempre y cuando sea compatible con la interfaz.

- Esto también significa que los programadores de los manejadores saben lo que se espera de ellos.

Uso de buffers

Una manera de lidiar, con el problema de leer caracteres desde un dispositivo, es hacer que el proceso de usuario realice múltiples llamadas ¨read¨ al sistema y se bloquee en espera de cada caracter.

- Cada caracter que llega produce una interrupción.
- El problema aquí es que el proceso usuario tiene que iniciarse por cada caracter entrante, y esto es ineficiente.

Uso de buffers

Una mejora sería proporcionar un buffer de n caracteres en el espacio del usuario y realizar la lectura de n caracteres.

- La rutina de servicio de la interrupción coloca los caracteres entrantes en este buffer hasta que se llene.
- Después despierta al proceso de usuario.
- Este esquema es más eficiente que el primero, pero tiene el problema que el espacio de direcciones en que se encuentra el buffer se podría paginar y salir de memoria.
- Una solución es que el proceso puede bloquear el buffer en memoria, pero si muchos procesos empiezan a bloquear páginas en memoria esto reducirá la cantidad de memoria disponible.

Uso de buffers

Otra opción es crear el buffer en el espacio de direcciones del kernel y hacer que la rutina de servicio coloque los caracteres allí.

- Cuando el buffer esté lleno se copia al espacio de direcciones del usuario.
- El problema aquí es que mientras se está realizando la copia pueden llegar nuevos caracteres.
- Una opción es utilizar un segundo buffer mientras el primero esté ocupado, luego se copia el segundo buffer y mientras el primero se empieza a llenar.
- A este esquema se le conoce como doble buffer.

Uso de buffers

Otra forma de solucionar este problema es utilizar un buffer circular.

- Conforme ingresan los caracteres se avanza un puntero de entrada, y conforme se copian los caracteres al área de usuario se avanza un puntero de salida.
- Ambos punteros regresan al inicio después de llegar a la parte final.

Dispositivos de bloques y de caracteres

Los dispositivos de E/S se pueden dividir básicamente en dos categorías: dispositivos de bloque y dispositivos de caracter.

- Un dispositivo de bloque almacena información en bloques de tamaño fijo, cada uno con su propia dirección.
- Todas las transferencias se realizan en unidades de uno o más bloques completos (consecutivos).
- La propiedad esencial de un dispositivo de bloque es que es posible leer o escribir cada bloque de manera independiente de los demás. Los discos duros, CD-ROM, y memorias USBs son dispositivos de bloque comunes.

Dispositivos de bloques y de caracteres

El otro tipo de dispositivo de E/S es el dispositivo de carácter.

- Un dispositivo de caracter envía o acepta un flujo de caracteres, sin importar la estructura del bloque.
- No es direccionable y no tiene ninguna operación de búsqueda. Las impresoras, las interfaces de red, y los teclados son considerados dispositivos de caracteres.

Referencia bibliográfica

A.SILBERSCHATZ, P. GALVIN, y G. GAGNE, Operating Systems Concepts, Cap.14, 9a Edición, John Wiley, 2013.