

Generalidades

- El subsistema de entrada y salida es la parte del sistema operativo encargada de la gestión de dispositivos de E/S.
- Los dispositivos de E/S varían ampliamente en sus funciones y velocidad, se requiere de una variedad de métodos para controlarlos.
- Muchos de los dispositivos operan asíncronicamente con respecto al procesador, a velocidades que van desde unas pocas transferencias hasta millones de transferencias de datos por segundo.
- El sistema de E/S atiende requisitos individuales de los dispositivos de entrada y salida.
- Se ocupa que funcionen eficientemente y fiablemente, ocultando los detalles específicos de cada dispositivo.
- Se encarga de mostrar una abstracción uniforme de las operaciones genéricas de E/S a las capas más externas del sistema operativo y a los usuarios.
- Es una interfaz entre los programadores y los dispositivos de E/S.

Hardware de E/S

- Los computadores operan muchos tipos de dispositivos.
 - Dispositivos de almacenamiento (discos, cintas).
 - Dispositivos de transmisión (tarjetas de red, modem).
 - Dispositivos para la interfaz con el ser humano (pantalla, teclado, ratón).
- El dispositivo se comunica con la máquina a través de un punto de conexión denominado puerto.
- Si uno o más dispositivos utilizan un conjunto común de cables, la conexión se denomina bus. Un bus es un conjunto de cables y un protocolo definido rígidamente que especifica un conjunto de mensajes que pueden enviarse por los cables.

Hardware de E/S

- Un **controlador** es un conjunto de componentes electrónicos que pueden operar un puerto, un bus o un dispositivo.
- El CPU le envía comandos a los controladores a través de registros que tienen los controladores, la CPU lee y escribe en ellos patrones de *bits*, mediante instrucciones especiales o una dirección de un puerto de E/S.
- Un puerto de E/S típicamente consta de cuatro registros: denominados registros de status, control, data-in y data-out.

Escrutinio

- El protocolo completo para la interacción entre un anfitrión y el controlador puede ser complejo, pero la noción básica de **secuencia de reconocimiento hanshaking**) es sencilla.
 - 2 *bits* para coordinar la relación productor consumidor entre el controlador y el anfitrión.
 - El *bit* busy en el registro status. El controlador lo activa cuando está ocupado trabajando y lo apaga cuando está listo para aceptar el siguiente comando.
 - El anfitrión activa el *bit* command-ready en el registro command, para indicarle al controlador que desea.

Escrutinio

1. El anfitrión lee repetivamente el *bit* busy hasta que dicho *bit* se apaga.
2. El anfitrión prende el *bit* write en el registro command y escribe un *byte* en el registro data-out.
3. El anfitrión prende el *bit* command-ready.
4. Cuando el controlador se da cuenta que el *bit* command-ready está prendido, prende el *bit* busy.
5. El controlador lee el registro de comando y ve el comando de escritura. Lee el registro data-out para obtener un *byte* y realiza la operación de E/S hacia el dispositivo.
6. El controlador apaga el *bit* command-ready, apaga el *bit* error en el registro de estado para indicar que ha tenido éxito la E/S al dispositivo y apaga el *bit* busy para indicar que ha terminado.

Interrupciones

- El *hardware* de la CPU tiene un cable llamado **línea de solicitud de interrupción** que la CPU revisa luego de ejecutar cada instrucción.
- La mayoría de CPU tiene dos líneas de interrupción.
 - Una es la **interrupción no enmascarable**, que se reserva para eventos como errores de memoria no recuperables.
 - La segunda línea es **mascarable**: puede ser apagada por la CPU antes de la ejecución de secuencias de instrucciones críticas que no deben ser interrumpidas. La interrupción mascarable es utilizada por los controladores de dispositivos para solicitar servicio.
- Las computadoras tiene más dispositivos que interrupciones, para resolver este problema se utiliza la técnica de **encadenamiento de interrupciones**.
- Se implementa también un mecanismo de **niveles de prioridad de interrupciones**.
- Las interrupciones son utilizadas para implementar el mecanismo de **llamadas al sistema**.

Ciclo de E/S activado por interrupciones

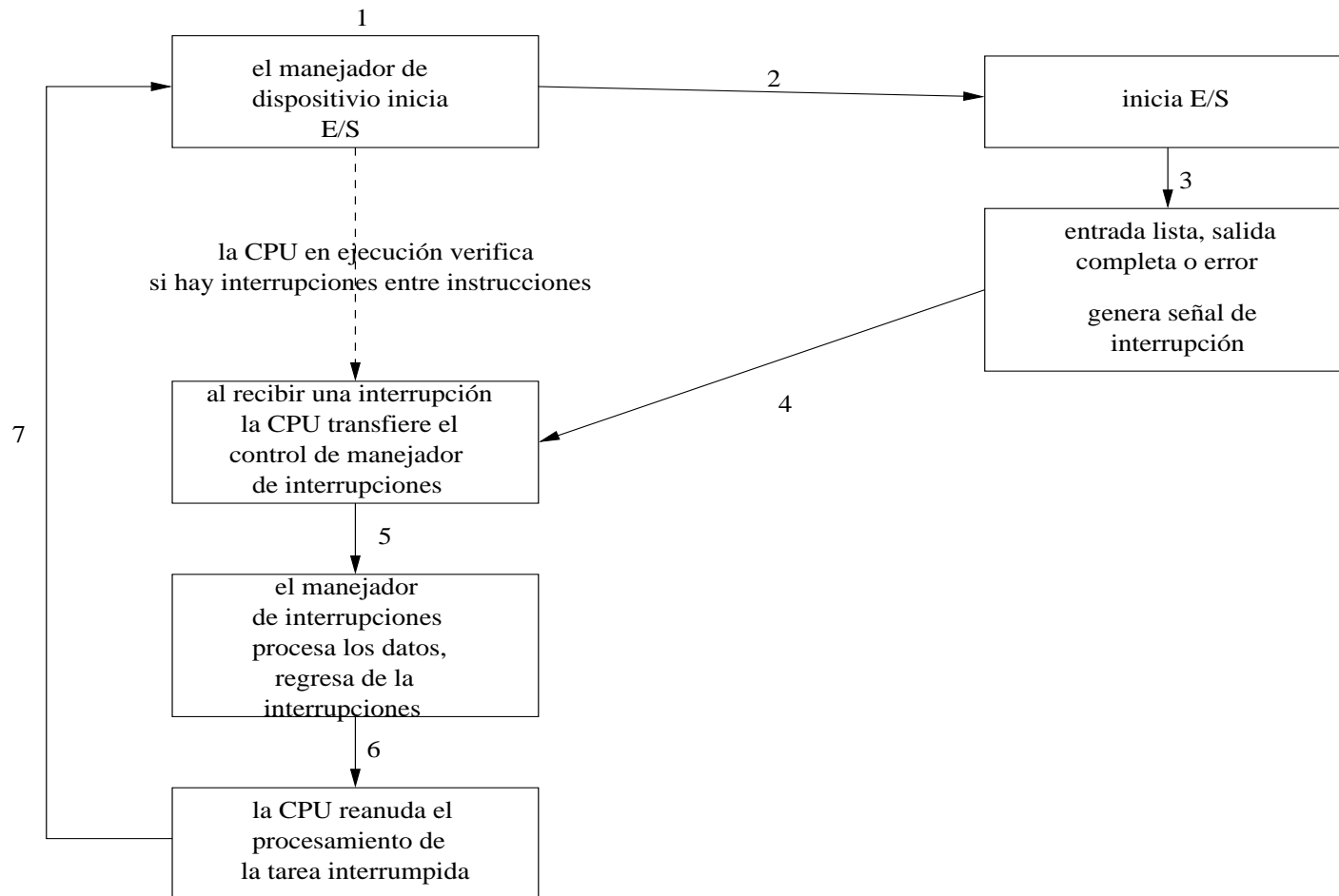


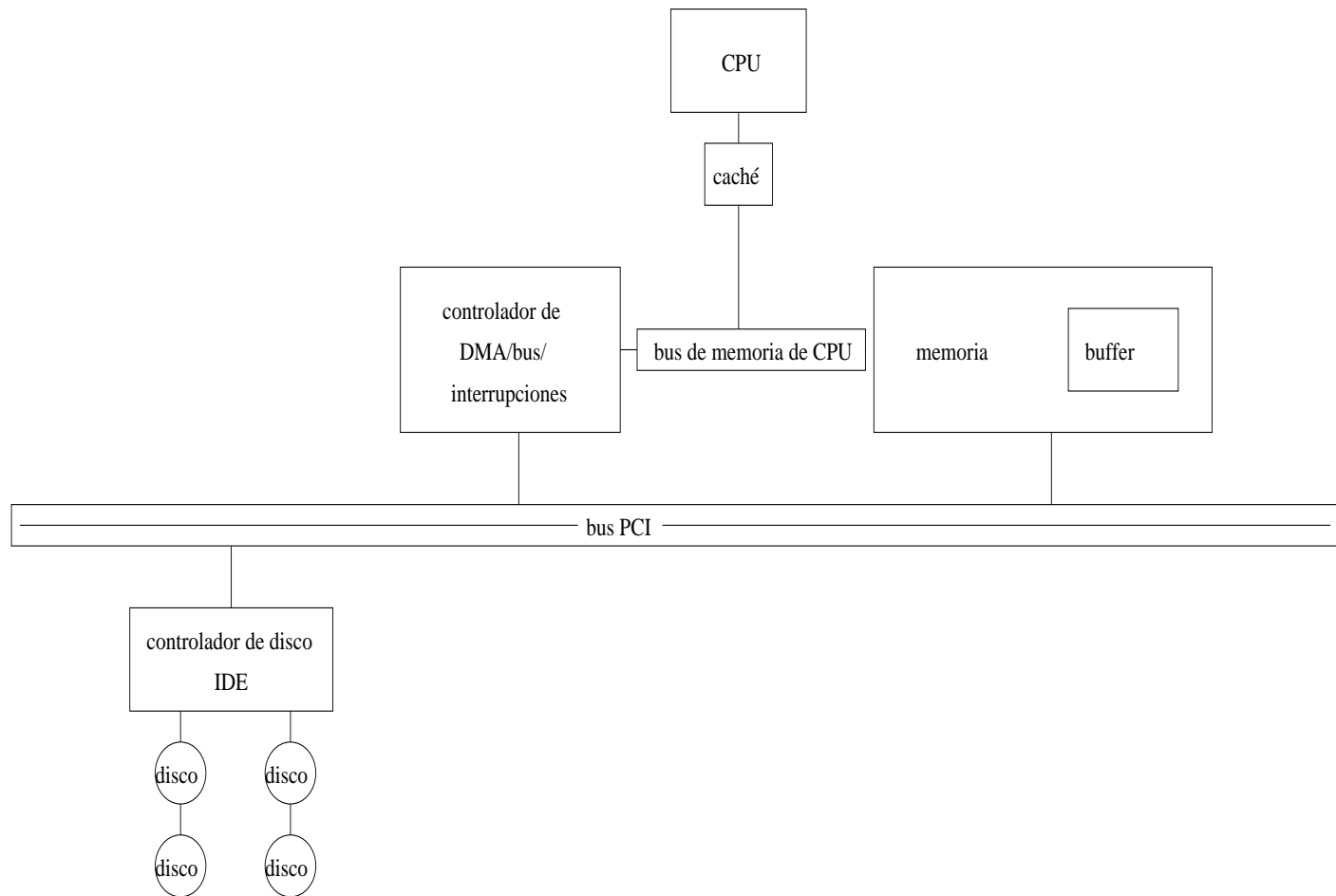
Tabla de vectores de eventos para el Pentium de Intel

número de vector	descripción
0	error de división
1	excepción de depuración
2	interrupción <i>null</i>
3	punto de corte
4	sobreflujo de INTO detectado
5	excepción de rango limitado
6	código de operación inválido
7	dispositivo no disponible
8	doble fallo
9	desbordamiento (reservado) de segmento de coprocesador
10	segmento de estado de tarea inválida
11	segmento ausente
12	fallo de pila
13	protección general
14	fallo de página
15	(reservado por Intel, no utilizar)
16	error de punto flotante
17	verificación de alineación
18	verificación de máquina
19-31	(reservado por Intel, no utilizar)
32-255	interrupciones enmascarables

Acceso directo a memoria

- Ayuda a disminuir la carga de procesador para vigilar el comportamiento de los dispositivos de entrada y salida.
- La técnica DMA trabaja del siguiente modo:
 - Si se solicita una lectura o una grabación, utiliza una línea de control de lectura o escritura entre el procesador y el módulo DMA.
 - Comunica la dirección del dispositivo de E/S involucrado, mediante líneas de datos.
 - Comunica la ubicación de comienzo de la lectura o escritura en la memoria mediante las líneas de datos y queda almacenada en el registro de dirección del módulo DMA.
 - Comunica el número de palabras a leer o escribir nuevamente mediante las líneas de datos y queda almacenada en el registro contador de datos.

Pasos en una transferencia DMA



Pasos en una transferencia DMA

1. Al manejador de dispositivos se le dice que transfiera datos del disco al *buffer* en la dirección X .
2. El manejador de dispositivo le dice al controlador de disco que transfiera C bytes del disco al *buffer* en la dirección X .
3. El controlador de disco inicia la transferencia DMA.
4. El controlador de disco envía cada *byte* al controlador de DMA.
5. El controlador de DMA transfiere *bytes* al *buffer* X , incrementando la dirección de memoria y decrecentando C hasta que $C = 0$.
6. Cuando $C = 0$, el DMA interrumpe la CPU para señalar la terminación de transferencia.

Objetivos de diseño de la E/S

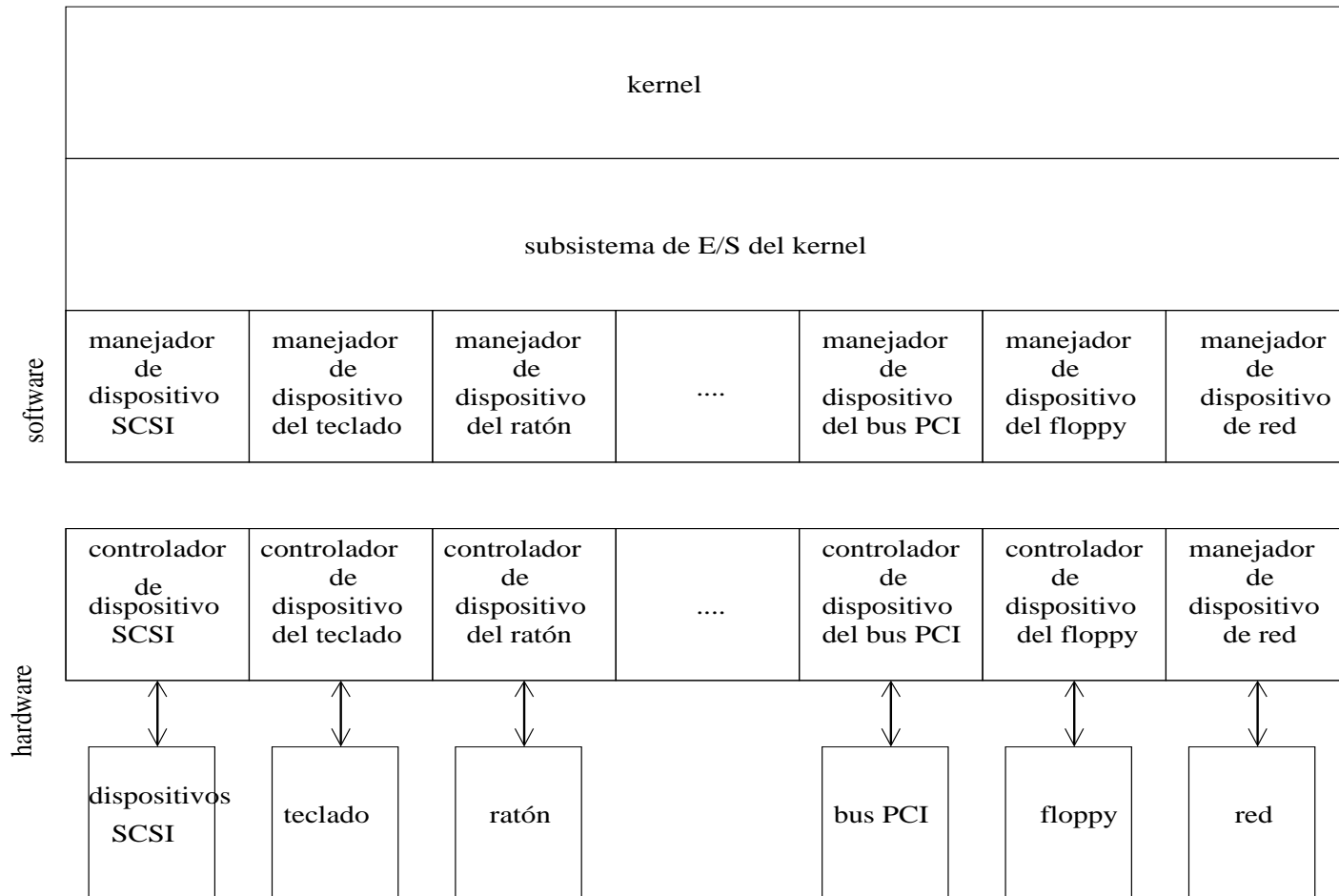
Eficiencia: es importante porque las operaciones de E/S constituyen, a menudo, un cuello de botella.

Generalidad: en interés de la simplicidad y la exención de errores, será deseable gestionar todos los dispositivos de una manera uniforme. Esta afirmación se aplica tanto a la manera en que los procesos contempla los dispositivos de E/S como a la forma en que el sistema operativo gestiona los dispositivos de E/S y las operaciones.

Interfaz de E/S de las aplicaciones

- La enfoque adoptado implica una abstracción, encapsulado y desarrollo de capas de *software*.
- La abstracción sirve para identificar unas cuantas clases generales.
- A cada una de estas clases generales se accede mediante un conjunto de funciones estandarizadas - una **interfaz**.
- Las diferencias reales se encapsulan en módulos del *kernel* denominados **manejadores de dispositivos**, que internamente se diseñan a la medida de cada dispositivo, pero que exportan una de las interfaces estándar.
- El propósito de la capa de manejadores de dispositivo es ocultar al subsistema de E/S del *kernel* las diferencias entre los controladores.
- El subsistemas de E/S es independiente del *hardware*.

Una estructura de E/S del *kernel*



Dimensiones para catalogar dispositivos

Flujo de caracteres o bloques: un **dispositivo de flujo de caracteres** transfiere *bytes* uno por uno, en tanto que un **dispositivo de bloque** transfiere un bloque de *bytes* como una unidad.

Acceso secuencial o aleatorio: un **dispositivo secuencial** transfiere datos en un orden fijo determinado por el dispositivo, en tanto el usuario de un **dispositivo de acceso aleatorio** puede instruir al dispositivo para que busque en cualesquiera de las ubicaciones disponibles de almacenamiento de datos.

Síncronico o asíncronico: un **dispositivo síncronico** es aquel que realiza transferencia de datos con tiempo de respuesta predecibles. Un **dispositivo asíncronico** exhibe tiempos de respuesta irregulares o impredecibles.

Dimensiones para catalogar dispositivos

Compartido o dedicado: un **dispositivo compartido** puede ser usado de manera concurrente por varios procesos o hilos; un **dispositivo dedicado** no puede hacerlo.

Velocidad de operación: las velocidades de los dispositivos van desde unos cuantos *bytes* por segundos hasta unos cuantos gigabytes por segundo.

Lectura-escritura, sólo lectura, o sólo escritura: algunos dispositivos realizan tanto operaciones de entrada como de salida, mientras que otros sólo soportan el movimiento de datos en una dirección.

Dipositivos

- Dispositivos de bloques y caracteres.
 - Bloques diseñados para acceder unidades de disco.
 - Operaciones: `read`, `write`, `seek`.
 - Mapeo de memoria.
 - Caracteres.
 - Operaciones: `get`, `put`.
- Dispositivos de red.
 - Operaciones: `read`, `write`.
 - Modos de comunicación: *half-duplex*, *full-duplex*.
- Relojes y temporizadores (timers).
 - Funciones básicas:
 - Dar la hora actual.
 - Indicar el tiempo transcurrido.
 - Fijar el valor de un temporizador para activar la operación X a la hora T .
 - *Hardware*: se denomina **temporizador de intervalos programable**
- E/S con bloqueo y sin bloqueo.

Subsistema de E/S del kernel

Los *kernels* proporcionan muchos servicios relacionados con la E/S.

- Planificación de E/S.
- Empleo de buffers.
- Empleo de cachés.
- Spooling y reservación de dispositivos.

Planificación de E/S

- Se debe determinar el orden de ejecución de las operaciones de E/S.
- El orden en que las aplicaciones emiten llamadas al sistema rara veces es la mejor opción.
- La planificación puede mejorar el desempeño global del sistema, así como compartir equitativamente el acceso a dispositivos entre los procesos y reducir el tiempo de espera promedio para la terminación de E/S.
- Los diseñadores de los sistemas operativos implementan la planificación manteniendo una cola de solicitudes para cada dispositivo.

Empleo de *buffers*

- Un *buffer* es un área de memoria que almacena datos mientras éstos se transfieren entre dos dispositivos o entre un dispositivo y una aplicación.
- Razones del uso de buffers.
 - Hacer frente a la falta de velocidad entre el productor y el consumidor de los datos.
 - Lograr una adaptación entre dispositivos que tienen diferentes tamaños de transferencia de datos.
 - Soportar la *semántica de copiado*.
- Políticas:
 - *Buffer* sencillos.
 - *Buffer* dobles.
 - Cola de *buffers*.

Empleo de cachés

- Una **caché** es una región de memoria rápida que contiene copias de datos.
- La diferencia entre un *buffer* y una caché es que un *buffer* es que un *buffer* puede contener la única copia existente de un elemento de datos, en tanto que la caché, por definición, sólo contiene en almacenamiento más rápido uan copia de un elemento que reside en alguna otra parte.

Spooling y reservación de dispositivos

- *Spool.*
 - Un **spool** es un *buffer* que contiene la salida para un dispositivo, como una impresora, que no puede aceptar flujos de datos intercalados.
 - La salida de cada aplicación se coloca en *spool* en un archivo en disco distinto.
 - Cuando una aplicación termina, el sistema de *spool* coloca en una cola de el archivo de *spool* correspondiente para su salida a la impresora.
 - El sistema de *spool* copia los archivos que están en la cola de la impresora uno a la vez.
 - Un proceso llamado proceso demonio *daemon*, se encarga de revisar esta última cola y cada vez que haya un archivo lo utiliza para enviar al dispositivo.
- Acceso exclusivo a dispositivos.

Manejo de errores

- Los dispositivos y las transferencias de E/S pueden fallar de muchas formas, ya sea por razones transitorias, o por razones permanentes.
- Los sistemas operativos puede compensar eficazmente las fallas transitorias.
- Cuando una operación de entrada/salida falla, está informa a través de la llamada al sistema que dicho error se produjo (En Unix, 0 ó 1, en Windows BOOL o FALSE).
- Y se devuelve el código de error a través de una variable del proceso (errno) o una llamada la sistema que la retorne (GetLastError).
- En constraste, el *hardware* puede proporcionar más información sobre la falla, pero los sistemas operativos no están diseñados para mostrar dicha información.

Subsistema de E/S

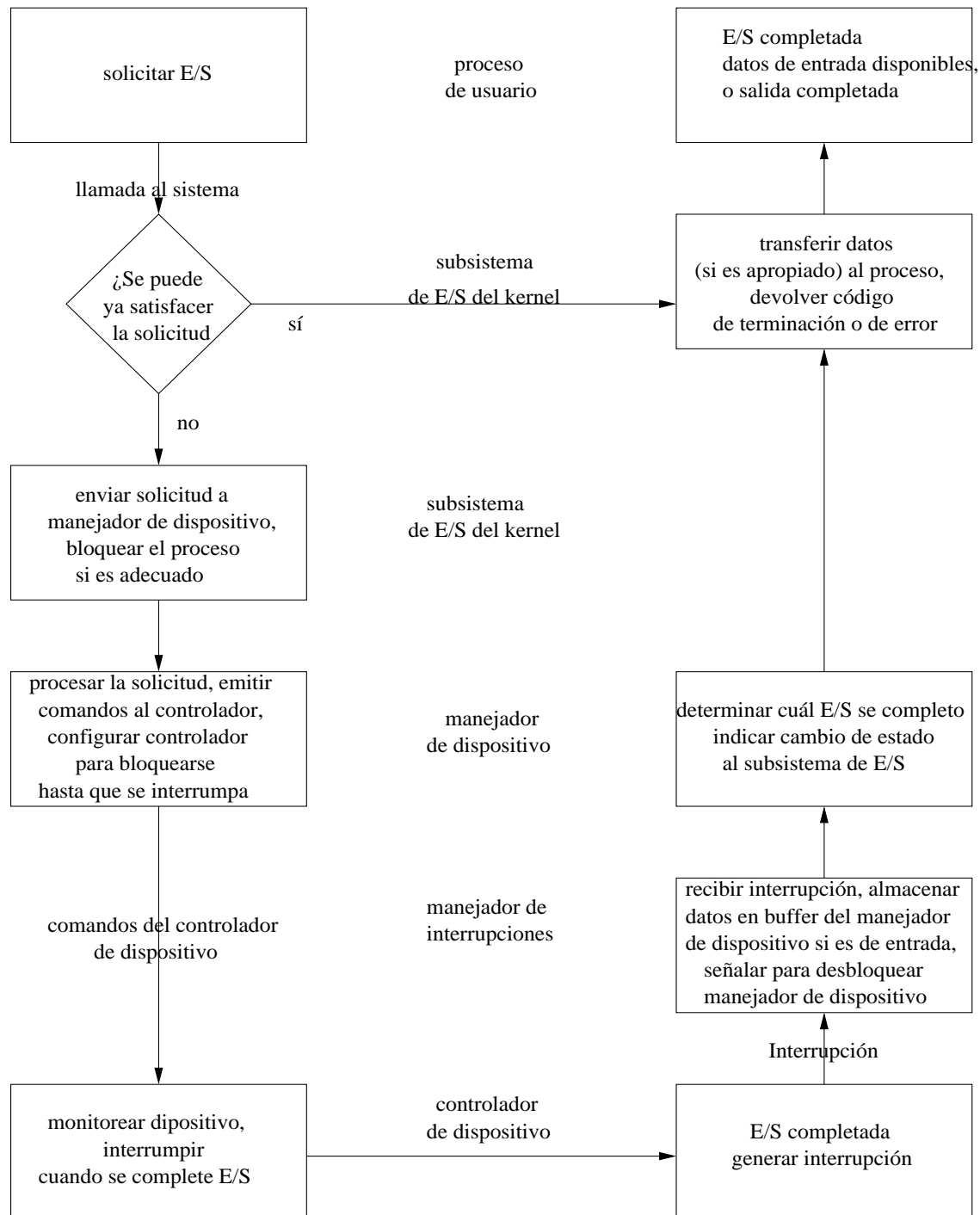
En resumen:

- La administración del espacio de nombres de archivos y dispositivos.
- El control de acceso a archivos y dispositivos.
- El control de operaciones.
- La asignación de espacio del sistema de archivos.
- La asignación de dispositivos.
- El empleo de *buffers*, cachés y *spool*.
- Planificación de E/S.
- El monitoreo del estado del dispositivo, el manejo de errores y la recuperación ante fallos.
- La configuración e inicialización de manejadores de dispositivos.

Manejo de solicitudes de E/S

- Identificación del dispositivo.
 - Sistema de archivos.
 - Sistema de nombres propios para dispositivos.
- Asociación del dispositivo.
- Implementación de los manejadores. (flujos).

Una operación de lectura bloqueante



Planificación de discos

- Planificación FCFS (*First Come First Server*).
- Planificación SSTF (*Short-Seek-Time-First*).
- Planificación SCAN.
- Planificación C-SCAN.
- Planificación LOOK.
- Planificación C-LOOK.

Administración de discos

- Formateo de discos.
- Bloque de arranque.
- Bloques dañados.