
Gestión de Entrada/Salida

Transparencias basadas en el libro de referencia:

Sistemas operativos. Una visión aplicada.

J. Carretero, F.García, P. de Miguel, F. Pérez. McGraw Hill 2001

Curso 2005-2006

Contenido

- Introducción
- Controladores de dispositivos
- Software de E/S
- Almacenamiento secundario

Introducción

La CPU es el principal elemento del computador

- procesa datos e instrucciones

Requiere colaboración de los **dispositivos de E/S**:

- *Periféricos*: Permiten la comunicación entre los usuarios y la computadora.
 - Entrada: teclado, ratón, ...
 - Salida: impresora, pantalla,...
- *Dispositivos de almacenamiento*. Proporcionan almacenamiento no volátil de datos y memoria.
 - Almacenamiento secundario: discos y disquetes
 - Almacenamiento terciario: cintas y sistemas de archivo
- *Dispositivos de comunicaciones*. Conectan el ordenador con otros ordenadores a través de una red
 - tarjetas de red, modems

Problema: Son bastante lentos

Funciones del sistema de E/S

El sistema de E/S se ocupa de facilitar el manejo de los dispositivos de E/S

- Oculta características físicas y de funcionamiento.

Funciones:

- Envío de comandos a los dispositivos, recibir sus interrupciones y ocuparse de sus errores.
- Ofrecer una interfaz entre los dispositivos y el resto del sistema simple y fácil de usar
- Optimizar la E/S del sistema
- Proporcionar dispositivos virtuales que permitan conectar cualquier tipo de dispositivo físico
- Permitir la conexión de nuevos dispositivos de E/S:

Componentes de los dispositivos de E/S

Los dispositivos de E/S constan de :

- Un elemento electrónico (**controlador** o unidad de E/S)
 - Transfiere información entre la memoria principal y los periféricos.
 - Conectado al bus y al dispositivo.
 - Pueden controlar múltiples dispositivos (p.e. el de disco)
 - Pueden incluir CPU y bus
- Un elemento mecánico (el dispositivo)
 - . Se conectan a la CPU a través de los controladores

El SO normalmente trata con el controlador, no con el dispositivo.

- Estandarización: Usar un mismo controlador para distintos dispositivos de distintos fabricantes (ejemplo: SCSI o IDE)

Contenido

- Introducción
- Controladores de dispositivos
- Software de E/S
- Almacenamiento secundario

Controladores de dispositivos

Constituye la interfaz del dispositivo con el bus de la computadora

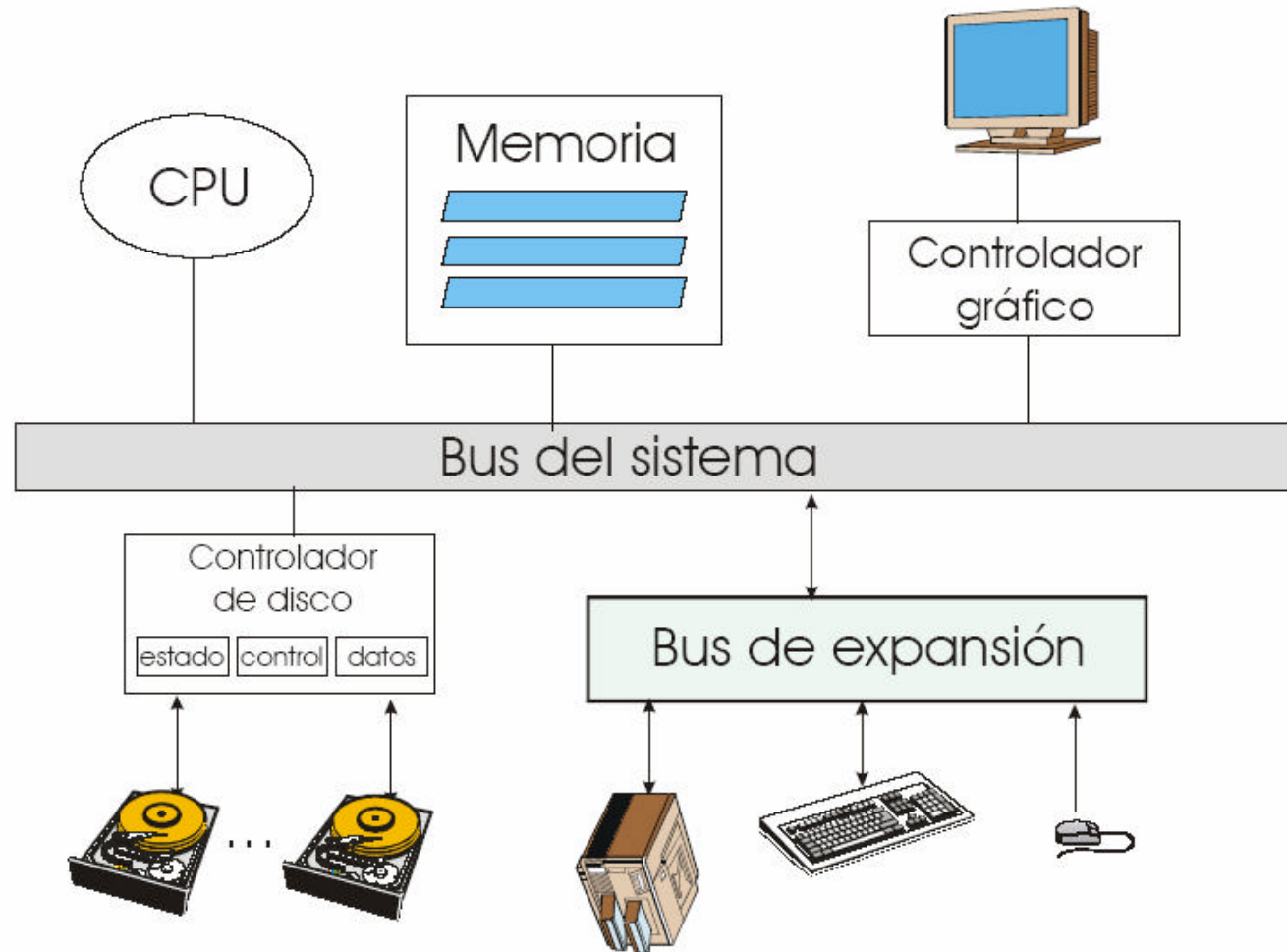
La comunicación entre la CPU y el controlador se realiza a través de los **registros** del controlador.

- Registro de datos: Almacena los datos de entrada o salida
- Registro de estado: Indica si la orden se ha ejecutado, si ha habido errores, ..
- Registro de control: Indica al controlador las ordenes a realizar

Estos registros:

- Forman parte del espacio normal de direcciones de memoria o,
- Tienen un espacio de direcciones especial

Controladores de dispositivos



Controladores de dispositivos: Clasificación

- Dirección de E/S
 - Dispositivos conectados por puertos
 - Dispositivos proyectados en memoria
- Unidad de transferencia
 - Dispositivos de bloques
 - Dispositivos de caracteres
- Interacción computadora-controlador
 - Entrada/salida programada
 - Entrada/salida por interrupciones
 - Acceso directo a memoria (DMA)

Dispositivos conectados por puertos o proyectados en memoria

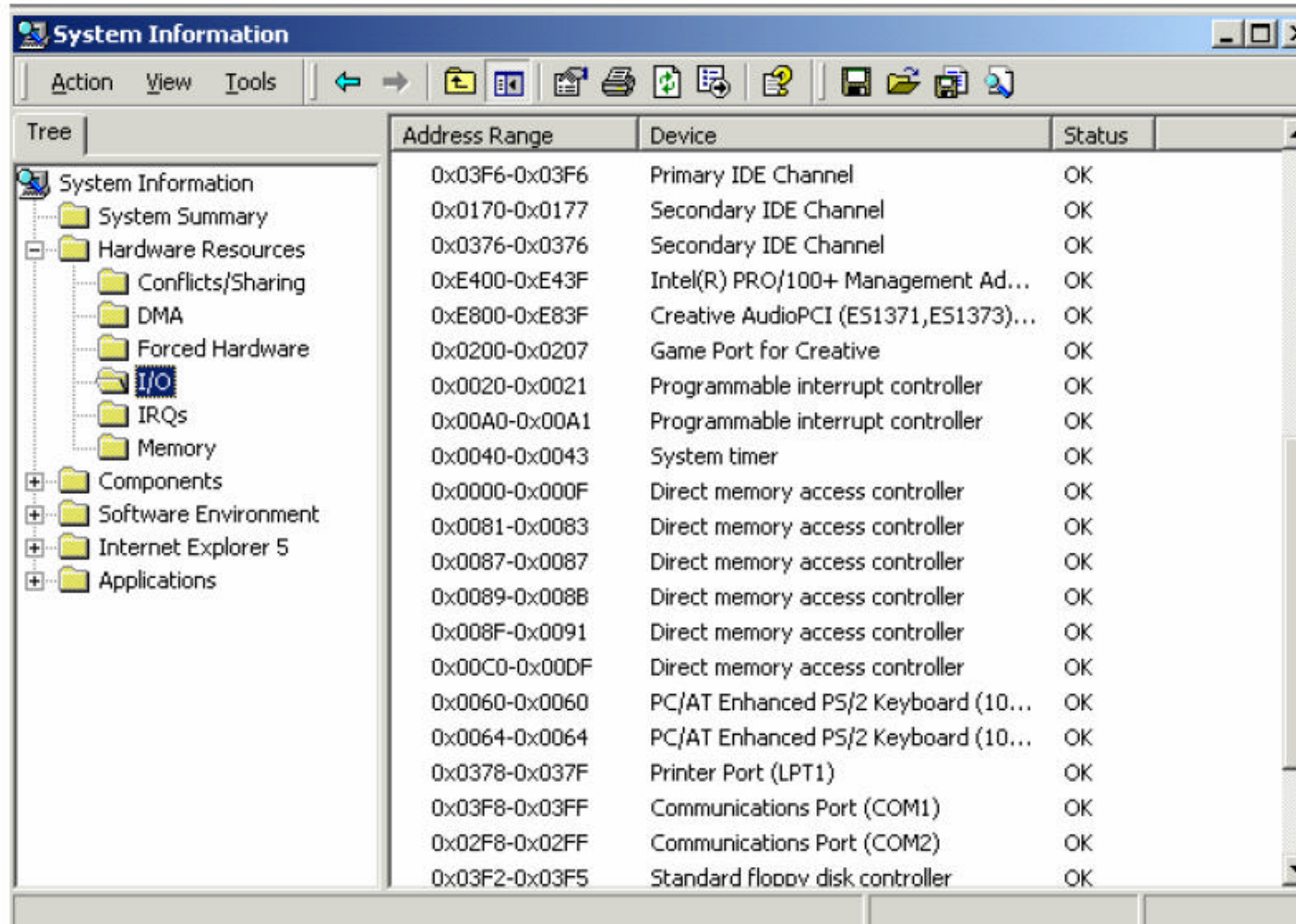
Dispositivos por puertos (arquitecturas de Intel)

- El controlador tiene asignado:
 - un puerto de E/S
 - una interrupción
 - un vector de interrupción.
- Operaciones específicas *portin* o *portout* en las que se indica
 - Dirección de puerto
 - Registro a manipular.
- Problema:
 - Es necesario conocer las direcciones de E/S
 - Programar las instrucciones especiales de E/S

Dispositivos proyectados en memoria (arquitecturas de Motorola)

- Tienen asignado un rango específico de direcciones en memoria
- Las operaciones se realizan mediante instrucciones máquina de manejo de memoria (No hay operaciones específicas)

Direcciones de E/S de algunos controladores



Dispositivos de bloques y de caracteres

Dispositivos de bloques:

- Acceso a nivel de bloque, secuencial o aleatorio
- Mandatos: leer, escribir, buscar, ..
- E/S directa o a través del servidor de ficheros
- Acceso posible a través de ficheros proyectados en memoria
- Ejemplos: discos y cintas

Dispositivos de carácter:

- Acceso a nivel de carácter, secuencial o aleatorio
- Mandatos: obtener (get) o colocar (put) un carácter.
- Bibliotecas para optimizar y dar forma a este tipo de accesos: edición de líneas, ventanas virtuales, ...
- Ejemplo: terminales e impresoras

Interacción computador-controlador: Tipos de entrada/salida

1. Entrada/salida programada.

- La CPU accede a los registros desde programa
- Consulta periódica para detectar si el dispositivo está listo

2. Entrada/salida por interrupciones

- El dispositivo avisa a la CPU cuando está listo
- La E/S se hace mediante una rutina de servicio de interrupción

3. Acceso directo a memoria.

- El controlador accede directamente a memoria
- Cuando la operación finaliza avisa a la CPU

E/S programada

1. La CPU verifica el estado del controlador leyendo el registro de estado
2. Mientras el controlador no está listo el proceso queda en un bucle
3. Cuando el controlador está listo, la CPU realiza la transferencia leyendo o escribiendo datos en el registro del controlador

Ventaja: No hay gasto de tiempo de gestión de interrupción

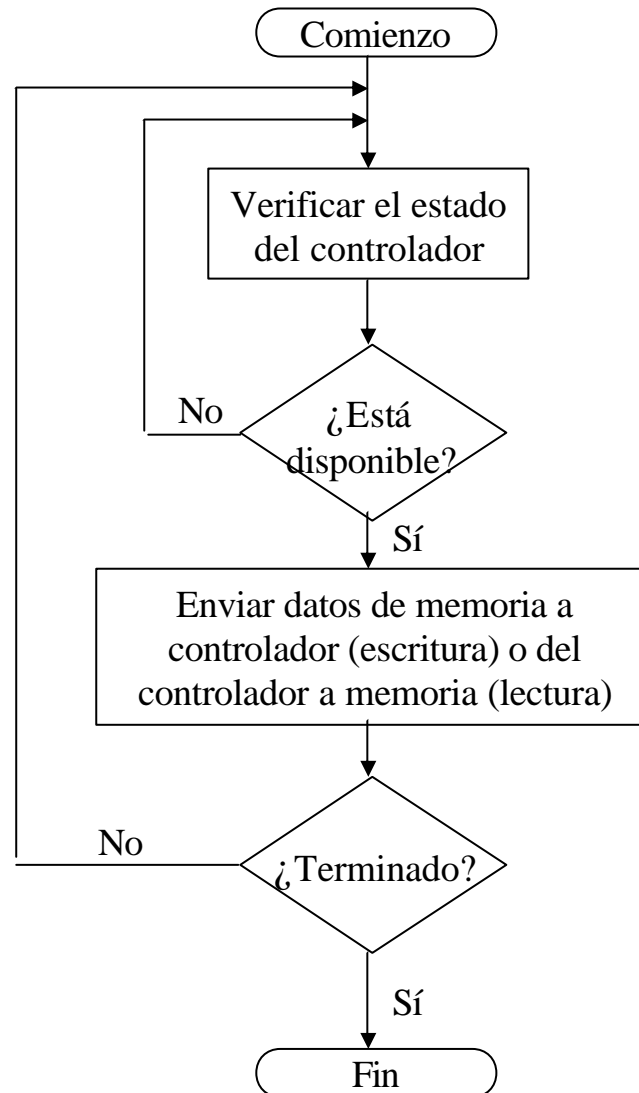
- Idóneo en sistemas de tiempo real en los que la velocidad de E/S es rápida

Desventaja: Grandes tiempos de espera.

Aplicación:

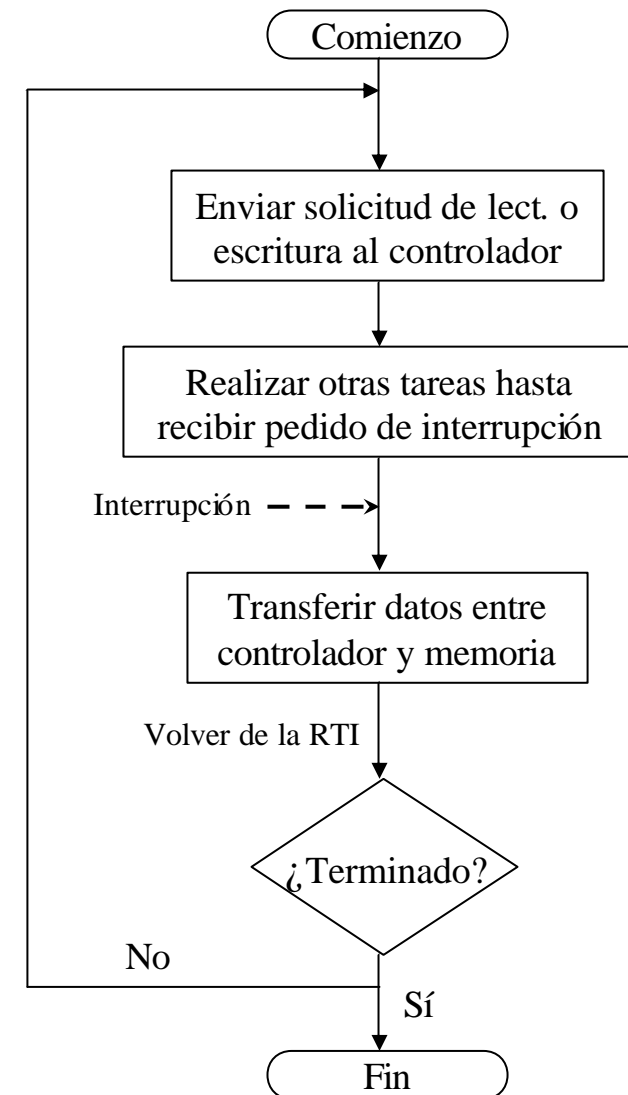
- Hardware sin interrupciones
- Programas de control de canales de E/S

E/S programada



E/S por interrupciones

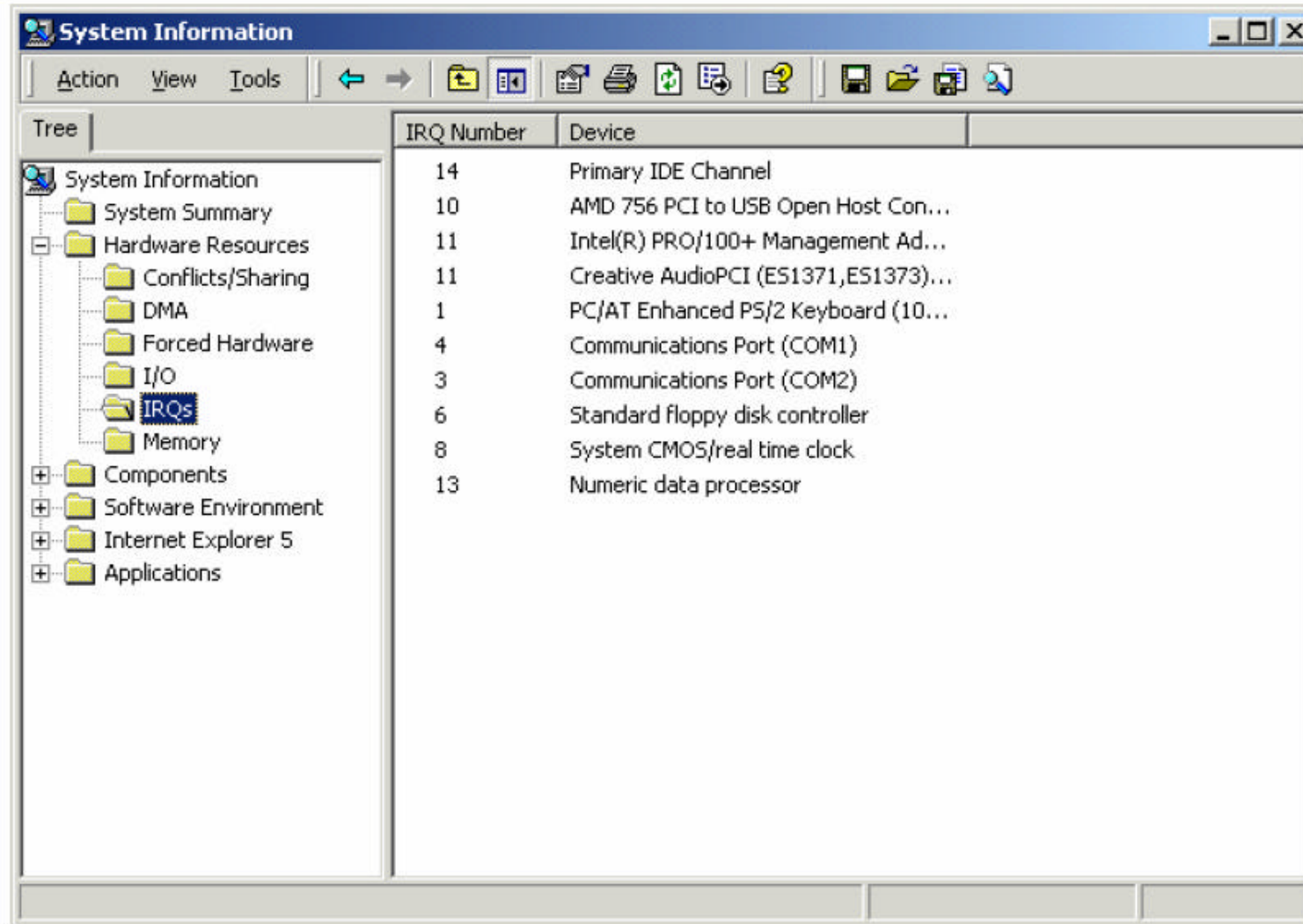
1. El procesador envía la orden de E/S al controlador del dispositivo y, en lugar de esperar, se dedica a otras tareas
2. Cuando el dispositivo está listo para intercambiar datos se activa una línea de pedido de interrupción
3. El procesador ejecuta la rutina de tratamiento de interrupción y, en ella, la transferencia de datos.



E/S por interrupciones (2)

- La gran mayoría de CPU's incluyen interrupciones vectorizadas
 - cada una está asignada a un dispositivo (o a un rango de ellos).
- Cada señal de interrupción tienen una determinada **prioridad**.
 - Si se activan varias señales simultáneamente, se tratará la de mayor prioridad
- La CPU incluye, también, un mecanismo de **inhibición**
 - Las señales inhibidas (enmascaradas) no son atendidas hasta que pasen a estar desinhibidas.

Interrupciones asociadas a algunos controladores



Rutina de tratamiento de interrupción

Las rutinas de tratamiento de interrupción suelen tener dos partes:

- Genérica: Proporcionada por el SO
- Particular: La proporciona el fabricante o el SO (dispositivo estándar)

La parte **genérica** permite:

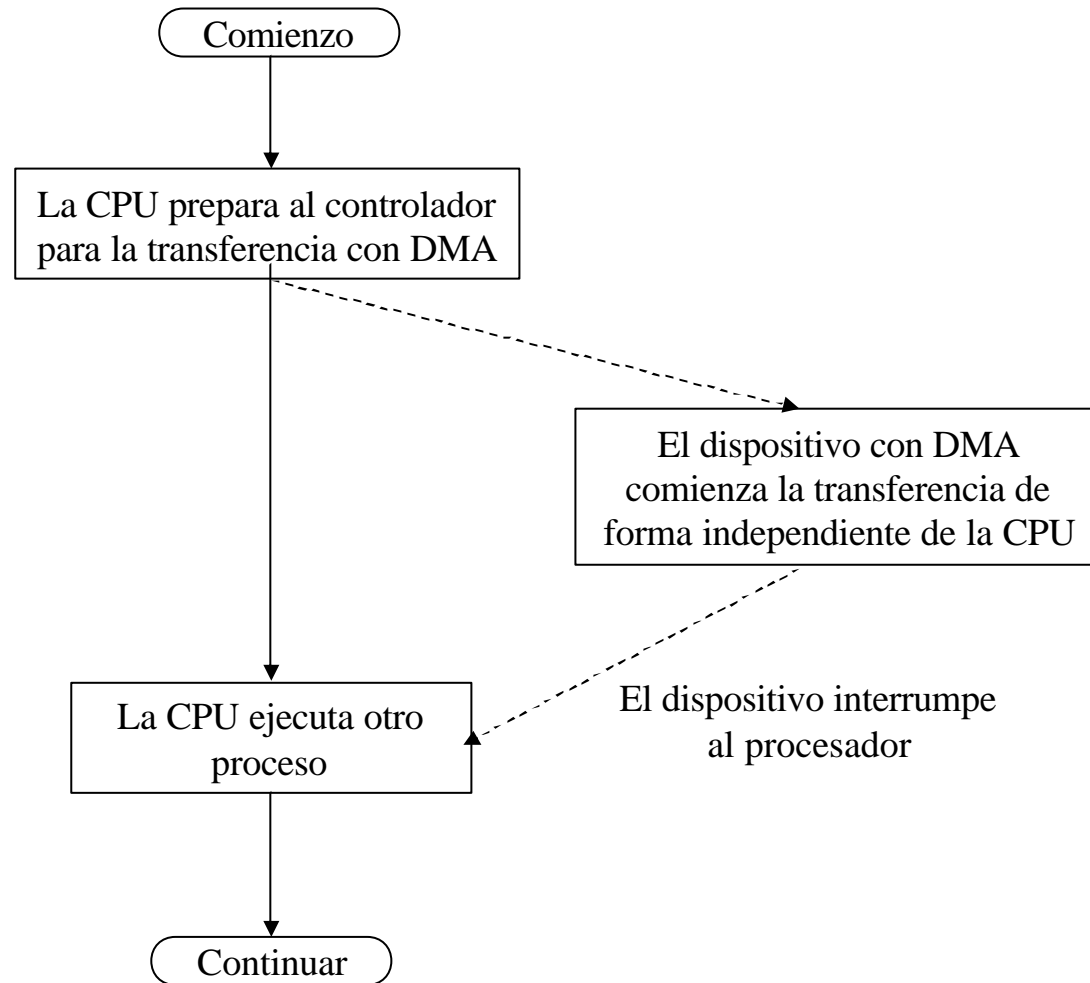
1. Capturar la interrupción
2. Salvaguardar el estado del procesador
3. Activar la rutina de manejo de la interrupción
4. Indicar al planificador que debe poner lista para ejecutar la rutina particular
5. Desactivar la interrupción
6. Restaurar el estado del procesador
7. Ceder el control (RETI)

La parte **particular** indica al planificador que encole la rutina particular (se activará cuando le llegue su turno de planificación).

Acceso directo a memoria (DMA)

- La transferencia la realiza el controlador
- Cuando la transferencia finaliza, se genera una interrupción
- Para realizar correctamente la operación, es necesario que la CPU indique al controlador:
 - Tipo de operación: lectura o escritura
 - Periférico involucrado en la operación
 - La dirección inicial de memoria desde la que se va a leer o a la que va a escribir
 - El número de palabras a leer o escribir

Acceso directo a memoria (DMA)



Pasos en una operación de lectura con DMA

1. Programación de la operación de E/S. Se indica al controlador:
 - Tipo de operación (en este caso lectura)
 - Datos a transferir
 - Dirección de memoria sobre la que se efectuará la operación.
2. El controlador contesta aceptando la petición de E/S
3. El controlador le ordena al dispositivo que lea una cierta cantidad de datos desde una posición determinada del dispositivo a su memoria interna
4. Cuando los datos están listos:
 - El controlador los copia a la posición de memoria que tiene en sus registros
 - Incrementa dicha posición de memoria y
 - Decrementa el contador de datos pendientes de transferir.
5. Los pasos 3 y 4 se repiten hasta que no quedan más datos por leer
6. Cuando el registro de contador está a cero, el controlador interrumpe a la CPU para indicar que la operación DMA ha terminado.

Contenido

- Introducción
- Controladores de dispositivos
- **Software de E/S**
- Almacenamiento secundario

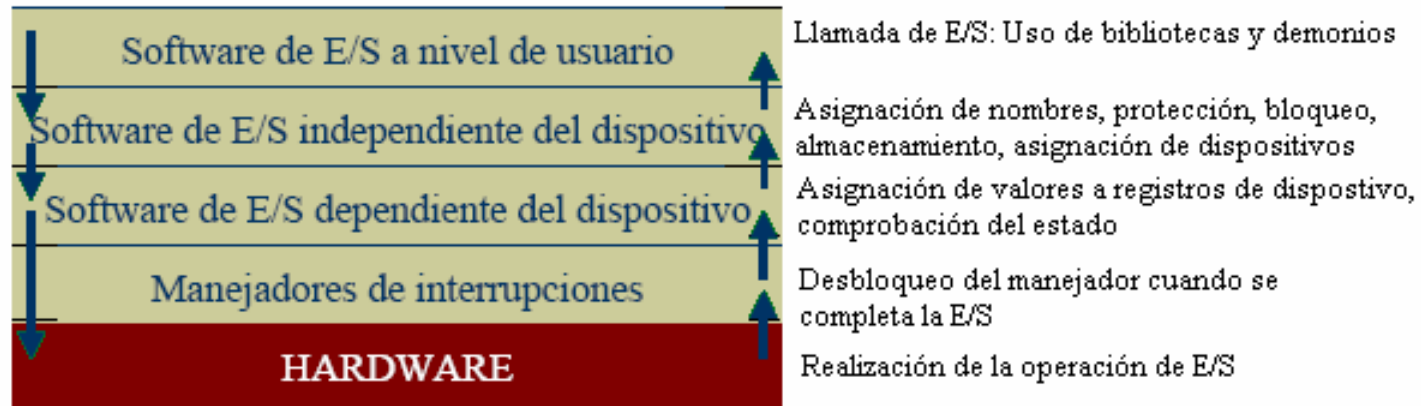
Software de E/S

El SO estructura el software de E/S en cuatro niveles, cada uno de las cuales tiene una función y una interfaz bien definidas.

- Manejadores de interrupciones
- Manejadores de dispositivos o drivers
- SW de E/S independiente del dispositivo
- SW de las aplicaciones de usuario.

Petición
de E/S

Respuesta
de E/S



Manejadores de interrupción

Se encargan de tratar las interrupciones generadas por los controladores

Ante la llegada de una interrupción es necesario:

- Interrumpir la ejecución de un proceso
- Salvar los registros
- Identificar la fuente de la interrupción y comunicar el evento al manejador del dispositivo.
- Ordenar la ejecución de la RTI
- Restaurar la ejecución de un proceso (que no tiene por qué ser el interrumpido)

Manejadores de dispositivos: Componentes

Cada dispositivo de E/S tiene un manejador asociado que incluye:

- Código independiente del dispositivo: Proporciona al nivel superior del SO una interfaz de alto nivel
- Código dependiente del dispositivo: Necesario para programar el controlador del dispositivo.
- Lista de peticiones pendientes por dispositivo:
 - Si existe alguna petición en la lista, selecciona una (FIFO, prioridad, etc.) y ordena su ejecución.
 - Si no hay peticiones, queda a la espera de que llegue alguna.

Manejadores de dispositivos: Funciones

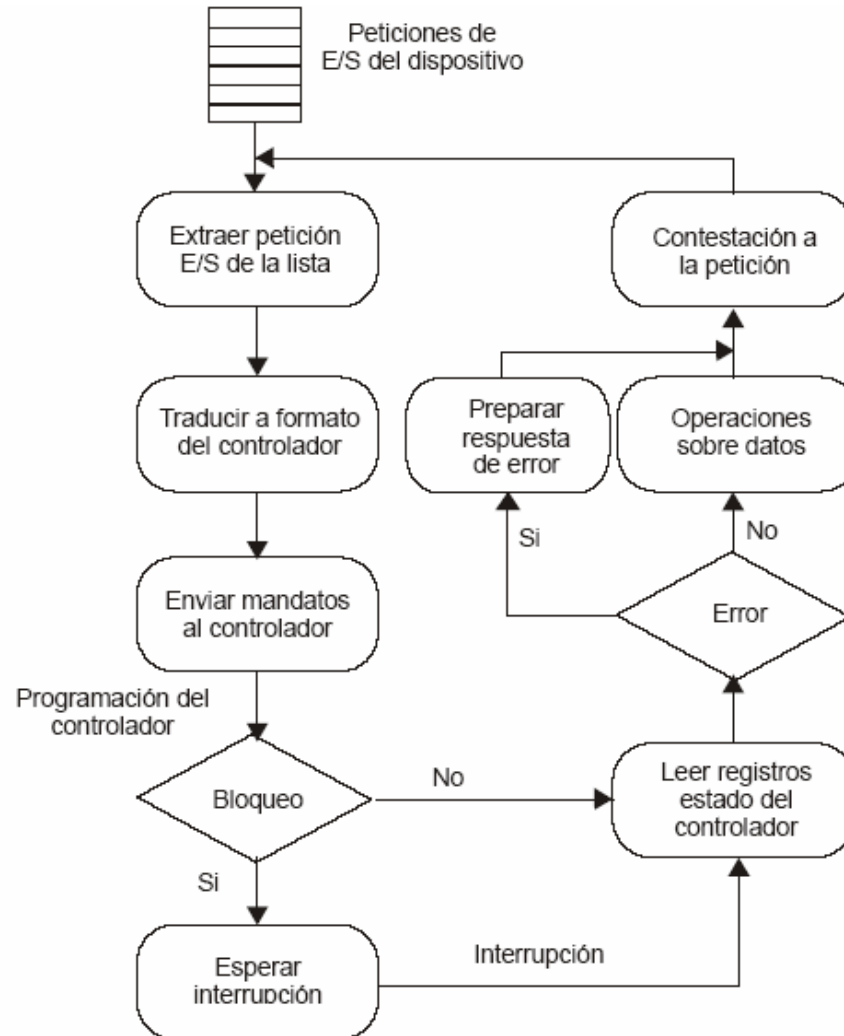
Funciones:

- Recibir peticiones abstractas de las rutinas independientes de los dispositivos
- Comprobar que se realizan.

Pasos:

- Traducir dichas peticiones a términos inteligibles por el controlador.
- Enviar al controlador la secuencia de ordenes adecuada
- Detectar el fin de la operación
- Comprobar si ha habido errores
- Comunicar al nivel superior el fin de operación y el posible error.

Operaciones de un manejador



Software independiente del dispositivo. Funciones

1. Asignar nombres simbólicos a los dispositivos de E/S
2. Evitar el acceso a usuarios in permiso
3. Ocultar las diferencias en el tamaño de bloque.
4. Proporcionar y gestionar almacenamiento temporal en memoria para:
 - Evitar acceso directo a los dispositivos
 - Ocultar las diferencias de velocidad
 - Facilitar la compartición.
5. Planificar la E/S de los distintos componentes.
6. Informar y gestionar aquellos errores que el manejador de dispositivo no puede solucionar (p.e. error permanente de lectura)

Software a nivel usuario. Componentes

El Software a nivel de usuario, o interfaz de aplicaciones, está formado por:

- Bibliotecas que se enlazan con los programas de usuario
 - Son el medio para realizar las llamadas al S.O. (fread o fwrite en C, Kernel132.dll en Windows)
- Los demonios del sistema, como los de red o los spooler de las impresoras.
 - Cuando una aplicación quiere acceder al puerto telnet, llama al demonio de red *inetd* y le pide el servicio.
 - Cuando se imprime un archivo, se envía a un proceso *spooler* que lo copia en unos determinados directorios y, posteriormente, lo imprime.

Software a nivel usuario

El software a nivel de usuario define el modelo de E/S que ven los usuarios.

Requiere tomar decisiones sobre:

- Nombres independientes de dispositivo
- E/S bloqueante y no bloqueante
- Control de acceso a dispositivos compartidos y dedicados
- Indicadores de error
- Uso de estándares

E/S bloqueante y no bloqueante

- Bloqueante: Procesos suspendidos hasta que la E/S termina
 - Cuando la E/S termina, se desbloquea el proceso.
 - Fácil de usar y comprender
 - Insuficiente para algunos requisitos
 - Ejemplo: *read*, *write* de POSIX

- No bloqueante: Permiten a la aplicación seguir su ejecución
 - Interfaz de usuario más complejo y espacio para copia de datos
 - Idóneo en sistemas que emiten peticiones y reciben respuesta después de un tiempo
 - Ejemplo: *aioread* y *aiowrite* de POSIX
 - Aplicación: Programas que leen datos de varios archivos y desean adelantar la lectura.

Control de acceso a dispositivos

Resolver conflictos de acceso

- Mandatos externos (como el lpr para la impresora) o programas especiales (demonios)
- Llamadas al sistema que permiten bloquear (lock) y desbloquear (unlock) el acceso a un dispositivo o a parte de él.

Controlar quién accede:

- El usuario debe tener derechos de acceso.

Contenido

- Introducción
- Controladores de dispositivos
- Software de E/S
- Almacenamiento secundario

Almacenamiento secundario. Estructura

- **Función:** Almacenar los programas y datos en dispositivos rápidos, de forma que sean fácilmente accesibles a las aplicaciones a través del sistema de archivos.
- **Componentes:**
 - **Discos.**
 - **Manejadores de disco.** Controlan las operaciones que se hacen sobre los discos.

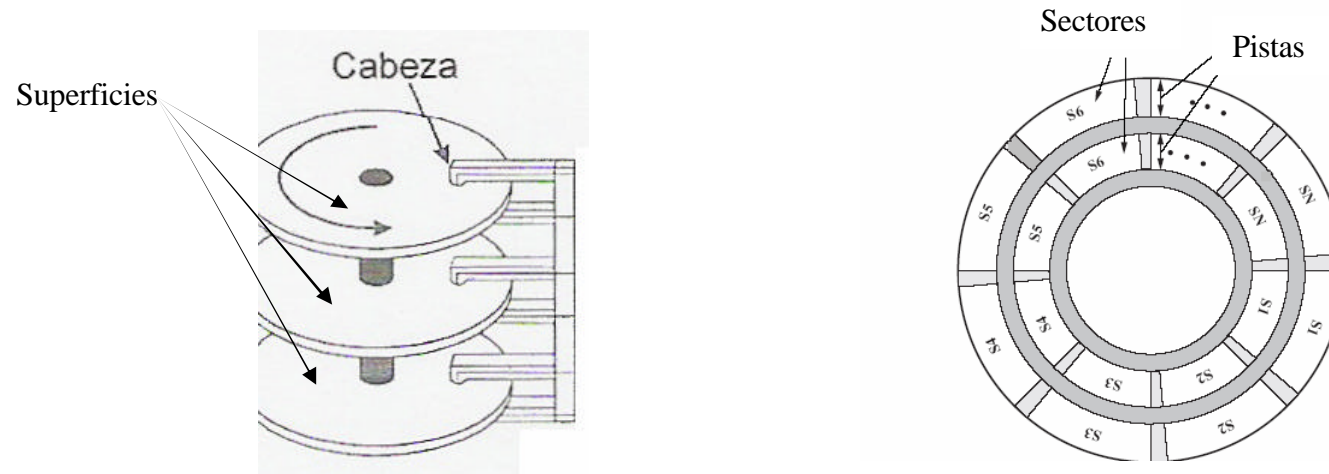
Discos

- Funciones:
 - Almacenamiento masivo y no volátil de datos
 - Plataforma para la memoria virtual.

- Clasificación:
 - Atendiendo a la interfaz de su controlador:
 - Dispositivos SCSI (Small Computer System Interface)
 - Dispositivos IDE (Integrated Drive Electronics)

 - Atendiendo a su tecnología de fabricación:
 - Discos duros
 - Discos ópticos
 - Discos extraíbles

Estructura física de los discos



- Capacidad del disco:

$$\text{Capacidad} = \text{superficies} * \text{pistas} * \text{sectores} * \text{tamaño sector}$$

- Parámetros fundamentales:

- Tiempo de búsqueda: Tiempo para posicionar la cabeza en la pista deseada
- Tiempo de latencia: Tiempo medio que tardan en llegar los datos debajo de las cabezas, una vez posicionadas en la pista

$$T_{\text{acceso}} = T_{\text{búsqueda}} + T_{\text{latencia}} + T_{\text{transferencia}}$$

Estructura lógica de los discos Formateo lógico

1. **Creación de un sistema de archivos** dentro de cada partición. (*format* de Windows o *mkfs* de UNIX)
2. **Construcción del bloque de carga.** Incluye el programa de arranque
3. **Creación de una lista de bloques defectuosos.**
 - Bloques que no son asignados por el sistema de E/S
 - Se marcan como ocupados y no se liberan nunca.
4. **Creación de una lista de bloques de repuesto**
 - Se usan para suplantar un bloque defectuoso.

El manejador de disco

Las funciones principales del manejador de disco son:

1. Proceso de la petición de E/S de bloques
2. Traducción del formato lógico a mandatos del controlador.
3. Inserción de la petición en la cola del dispositivo, llevando a cabo la política de planificación de disco pertinente (FIFO, SSTF, SCAN, C-SCAN, etc.)
4. Envío de los mandatos al controlador, programando el DMA.
5. Bloqueo en espera de la interrupción de E/S
6. Comprobación del estado de la operación cuando llega la interrupción
7. Gestión de errores, si existen, y resolverlos si es posible
8. Indicación del estado de terminación al nivel superior del sistema de E/S

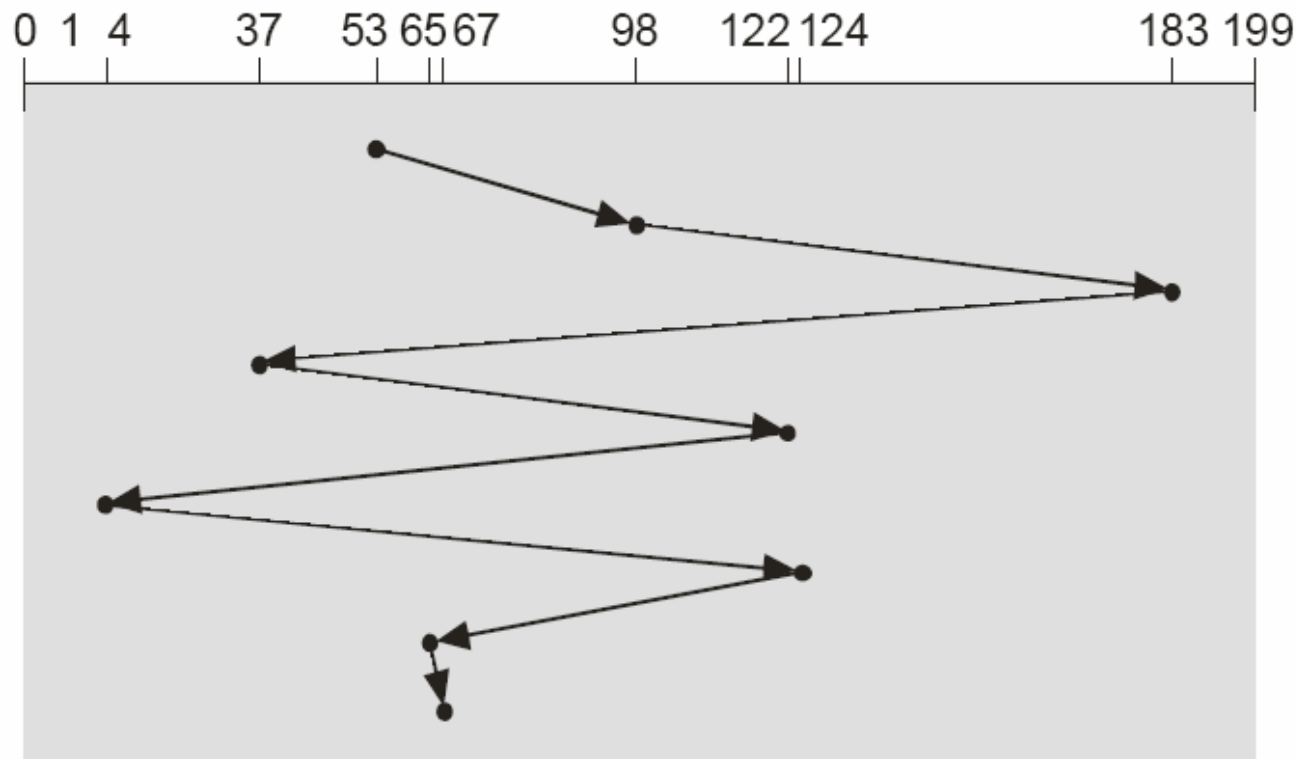
Se analizará la planificación y la gestión de errores.

Planificación de discos

- Fundamental para optimizar el acceso a disco
- Criterios
 - Optimizar el tiempo de búsqueda
 - Dar servicio determinista
- Algoritmos:
 - FCFS
 - SSTF
 - SCAN
 - C-SCAN

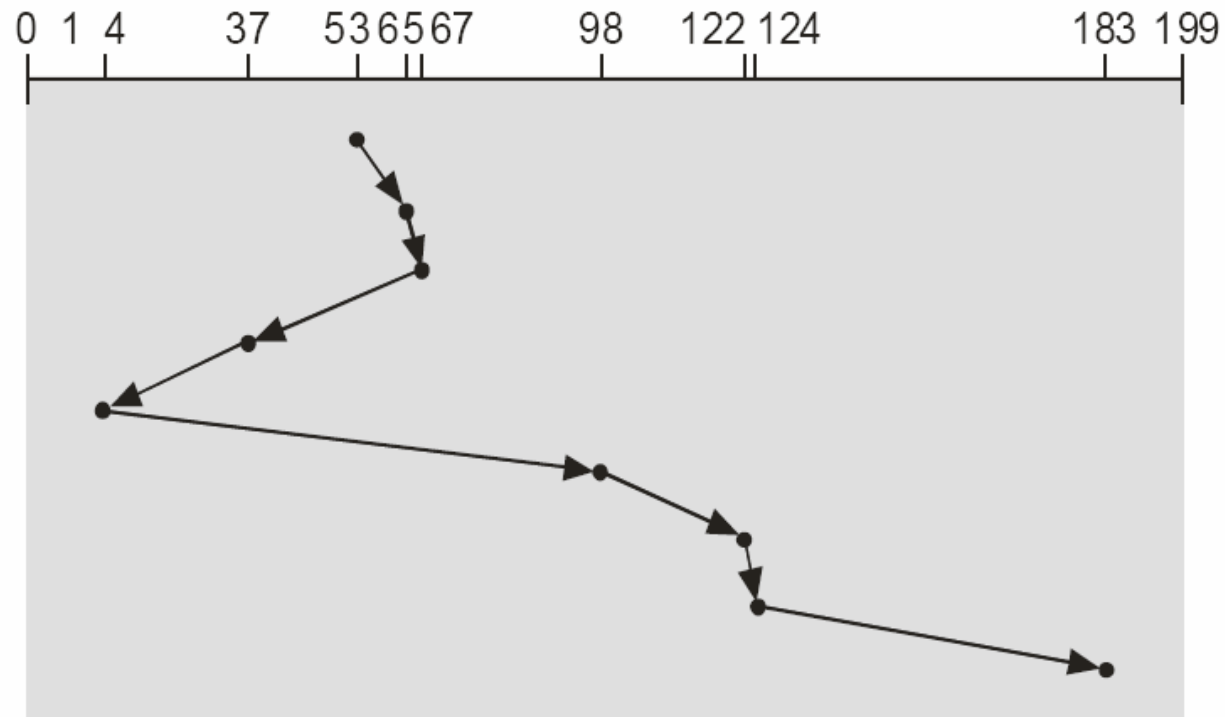
Planificación FCFS

Ejemplo: Cabeza situada en el cilindro 53 y solicitudes de E/S a bloques situados en los cilindros 98, 183, 37, 122, 14, 124, 65, 67



Planificación SSTF

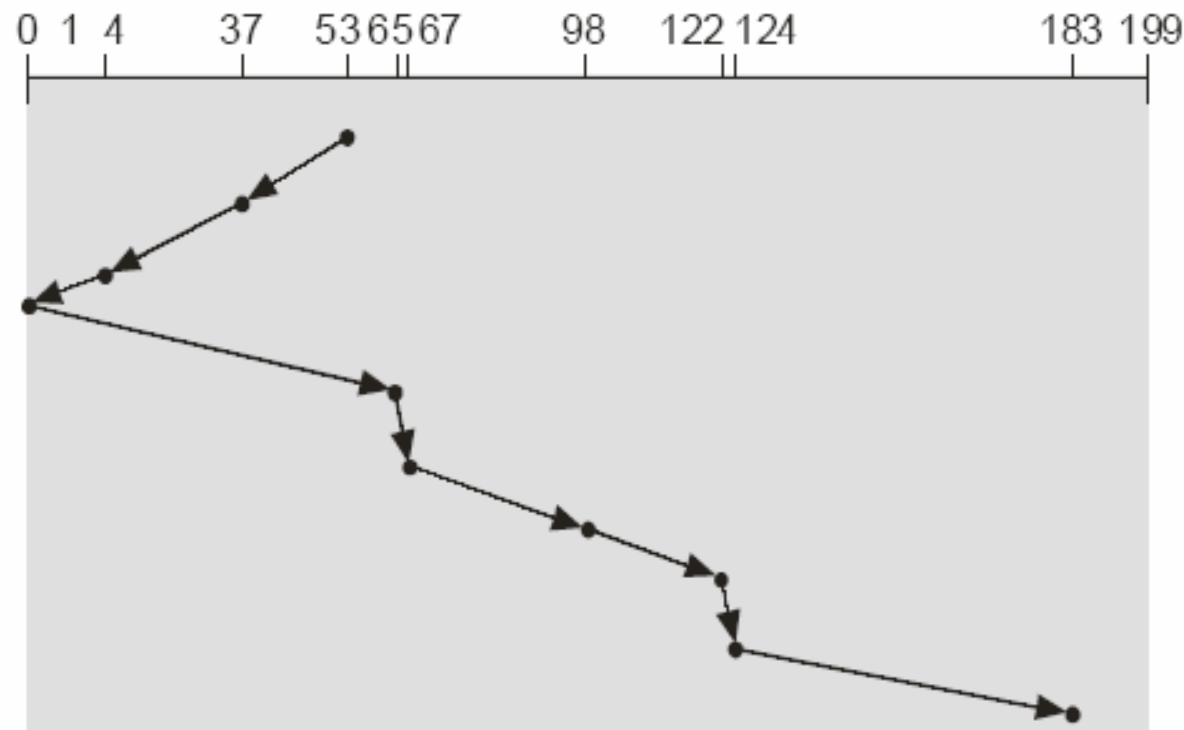
Se planifica primero la solicitud que tiene el menor tiempo de búsqueda:



Cola de peticiones 98, 183, 37, 122, 14, 124, 65, 67

Planificación SCAN

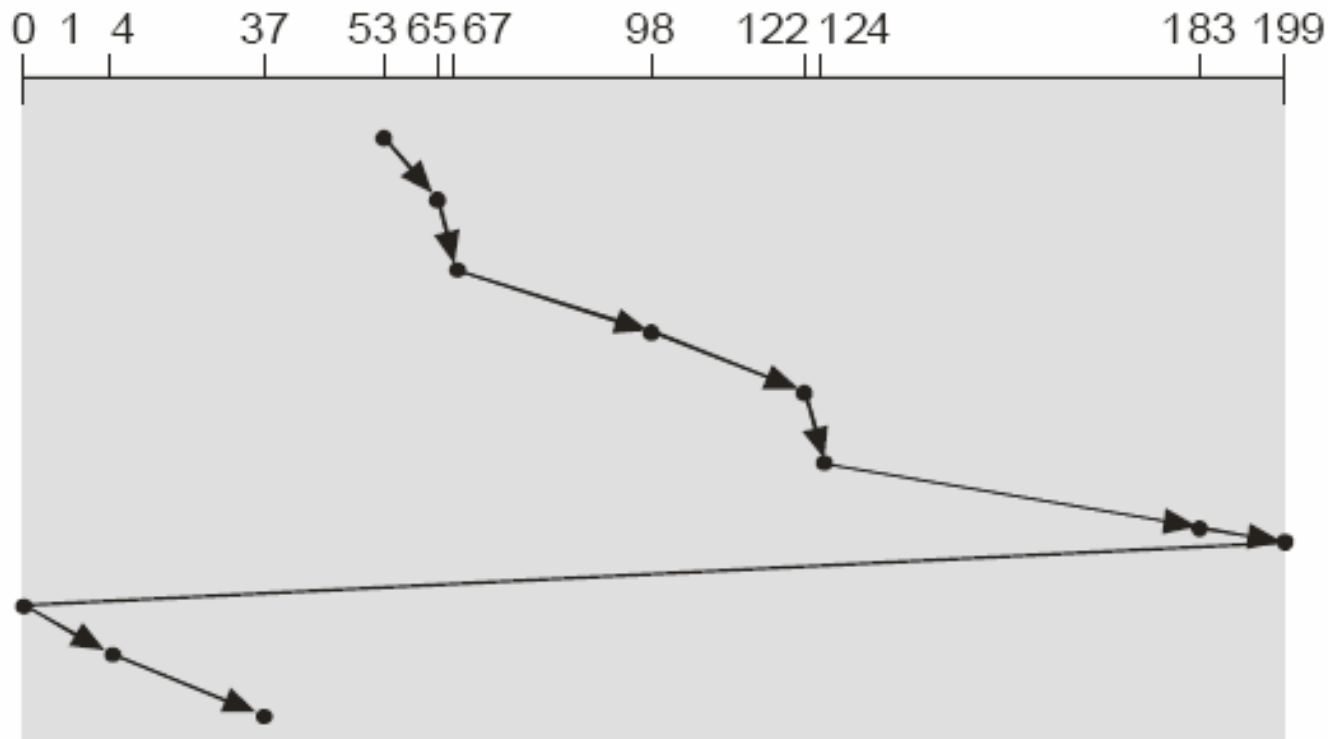
El movimiento del brazo empieza siempre en un extremo del disco y continúa hasta el otro. Allí se cambia el sentido y se vuelve al otro extremo.



Cola de peticiones 98, 183, 37, 122, 14, 124, 65, 67

Planificación C-SCAN

Las cabezas se mueven de un extremo a otro del disco, atendiendo las solicitudes en el camino, pero cuando la cabeza llega al extremo final, regresa de inmediato al principio sin atender solicitudes.



Cola de peticiones 98, 183, 37, 122, 14, 124, 65, 67

Selección de un Algoritmo de Planificación

- SSTF es frecuente y parece el más natural
- SCAN Y C-SCAN tiene mejor rendimiento para sistemas que usan mucho el disco
- El rendimiento depende del número y el tipo de peticiones
- Las peticiones al disco pueden depender de la política de asignación de espacio a los ficheros
- El algoritmo de planificación de disco debe ser escrito como un módulo separado, para que sea fácil de reemplazar
- El algoritmo CSCAN es el más usado actualmente.

Gestión de errores

Los errores pueden provenir de:

- Las aplicaciones: Petición para un dispositivo o sector que no existe
- Del controlador: Errores al aceptar peticiones o parada del controlador
- De los dispositivos. Fallos transitorios o permanentes de lectura o escritura y fallos en la búsqueda de pistas.
- Errores transitorios: Se resuelven en el manejador.
 - Debidos a partículas de polvo, pequeñas variaciones eléctricas, fallos de calibración de cabezas,...
 - Se detectan mediante el ECC (Error Correcting Code)
 - Se resuelven repitiendo la operación de E/S
 - Si no se resuelven se transforman en error permanente.
- Errores permanentes: Se comunican al nivel superior
 - Errores de aplicación: poco que hacer
 - Errores de controlador: tratar de reiniciar el controlador
 - Errores de superficie del dispositivo: sustituir el bloque por uno de repuesto

Fiabilidad y tolerancia a fallos. Técnicas

- Códigos correctores de error, como los existentes en las cabeceras y colas de los sectores
- Técnicas de almacenamiento estable
- Redundancia: Dispositivos RAID (Redundant Array of Inexpensive Disk)
- Redundancia hardware: Conectar un disco a dos controladores (Ej. Windows NT)

Almacenamiento estable

- Uso de discos espejo.
- Escritura. Supone escribir en ambos discos con éxito
 - Se escribe en un disco y luego en el otro
 - Si ambas escrituras tienen éxito, éxito en la operación
 - Si una de las dos falla, error parcial.
 - Si ambas fallan, fallo total del almacenamiento estable.
- Lectura.
 - Para leer, basta con que uno de los dispositivos esté disponible
 - Para verificar el estado de los datos globales,
 - Leer ambos dispositivos y comparar los datos.
 - En caso de error, se elige uno de ellos
- Desventajas:
 - Bajo rendimiento de las operaciones de escritura
 - Mal aprovechamiento del espacio
- Ventajas:
 - Es barata y sencilla de implementar.
 - Alto rendimiento en las operaciones de lectura.

Dispositivos RAID

- Un único controlador y varios discos
 - Discos para almacenar la información
 - Discos para almacenar información de paridad
- Niveles:
 - RAID 1. Discos espejo
 - RAID 2. Distribuye los datos de acuerdo a una unidad de definida por el sistema o la aplicación.
 - RAID 3. Reparte los datos a nivel de bit por todos los discos.
 - Se pueden añadir bits con códigos correctores de error.
 - Exige que las cabezas estén sincronizadas.
 - RAID 4. Reparto de bloques y cálculo de paridad para cada franja de bloques que se almacena en un disco fijo.
 - Ejemplo: Cuatro discos para datos y un quinto para paridad
 - RAID 5. Reparto de bloques y paridad por todos los discos de forma cíclica.

Redundancia con RAID 5

