

Tema 3

Sistema de Entrada/Salida

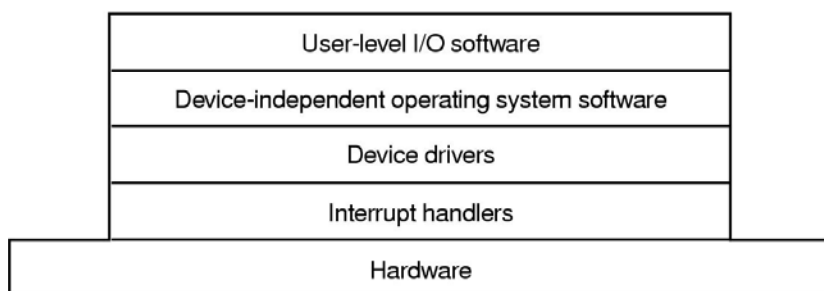
SISTEMA DE ENTRADA/SALIDA

- El corazón de una computadora lo constituye la UCP, pero no serviría de nada sin la memoria principal y los dispositivos de E/S:
 - Dispositivos periféricos, que le permitan interactuar con el usuario.
 - Generalmente están fuera de la computadora y se conectan a ella mediante cables.
 - Son los teclados, ratones, micrófonos, cámaras y cualquier otro dispositivo de E/S que se le ocurra conectar a una computadora.
 - Dispositivos de almacenamiento secundario (discos) y terciario (cintas y sistemas de archivo).
 - Dispositivos de comunicaciones.

SISTEMA DE ENTRADA/SALIDA

- El acceso a los dispositivos es:
 - Incómodo
 - Detalles físicos de los dispositivos
 - Dependiente de las direcciones físicas
 - No seguro
 - Si el usuario accede a nivel físico no tiene restricciones
- Objetivos:
 - Suministrar una visión lógica de los dispositivos
 - Ofrecer primitivas de acceso cómodas e independientes de los detalles físicos
 - Mecanismos de protección

Niveles del software de E/S



Niveles del software e un sistema de E/S

Entrada/Salida

- **Caracterización de los dispositivos de E/S**
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- Almacenamiento terciario
- El reloj
- El terminal
- La red
- Servicios de entrada/salida

Clasificación de dispositivos E/S

- **Periféricos.** Se llama así a los dispositivos que permiten la comunicación entre los usuarios y la computadora. Dentro de este grupo se incluyen todos los dispositivos que sirven para proporcionar interfaz con el usuario, tanto para entrada (ratón, teclado, etc.) como para salida (impresoras, pantalla, etc.).
- **Dispositivos de almacenamiento.** Se usan para proporcionar almacenamiento no volátil de datos y memoria. Su función primordial es abastecer de datos y almacenamiento a los programas que se ejecutan en la UCP. Según su capacidad e inmediatez se pueden dividir en almacenamiento secundario (discos y disquetes) y terciario (cintas y sistemas de archivo).
- **Dispositivos de comunicaciones.** Permiten conectar a la computadora con otras computadoras a través de una red. Los dos tipos de dispositivos más importantes de esta clase son los MODEM y las tarjetas de interfaz a la red.

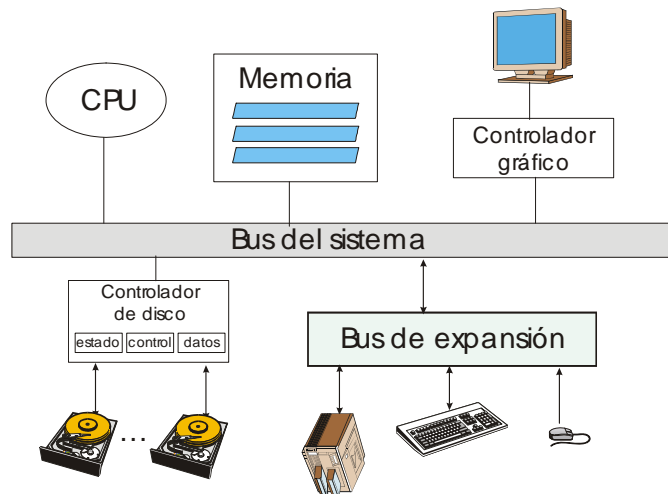
Velocidades de dispositivos típicos

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec

Dispositivos de Bloque y de Carácter

- **Dispositivos de bloque:**
 - 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
 - Ejemplo: discos y cintas
- **Dispositivos de carácter:**
 - Acceso a nivel de carácter, secuencial o aleatorio
 - Mandatos: get, put,
 - Bibliotecas para optimizar y dar forma a este tipo de accesos: edición de líneas, ventanas virtuales,
 - Ejemplo: terminales e impresoras

Conexión de dispositivos de E/S



Controladores de dispositivos I

- El dispositivo de E/S tiene dos componentes:
 - componente mecánico
 - componente electrónico
 - El componente mecánico se conectan a la CPU a través del controlador
- ↓
- El componente electrónico es el controlador del dispositivo
 - Puede controlar varios dispositivos de E/S
 - Tareas del controlador
 - convertir bit en serie (bit stream) a bloques de bytes
 - Realizar corrección de errores si es necesario
 - Dejar la información en MP

Controladores de dispositivos II

- Los controladores son muy variados, casi como los dispositivos de E/S.
 - Muchos de ellos pueden controlar múltiples dispositivos.
 - Otros, incluyen su propia UCP y bus para controlar la E/S por programa y evitar interrupciones en la UCP de la computadora.
- En los últimos años ha existido un esfuerzo importante de estandarización de los dispositivos, lo que permite usar un mismo controlador para dispositivos de distintos fabricantes (ejemplo: SCSI, *Small Computer System Interface*, o IDE, *Integrated Drive Electronics*).
- El controlador es la interfaz del dispositivo con el bus de la computadora y es el componente que se ve desde la UCP.
 - Su programación se lleva a cabo mediante una interfaz de muy bajo nivel que proporciona acceso a una serie de registros del controlador.
 - Las características del controlador son muy importantes, ya que definen el aspecto del periférico para el sistema operativo.
 - Fundamental: Dirección de E/S, Unidad de transferencia e Interacción computadora-controlador.

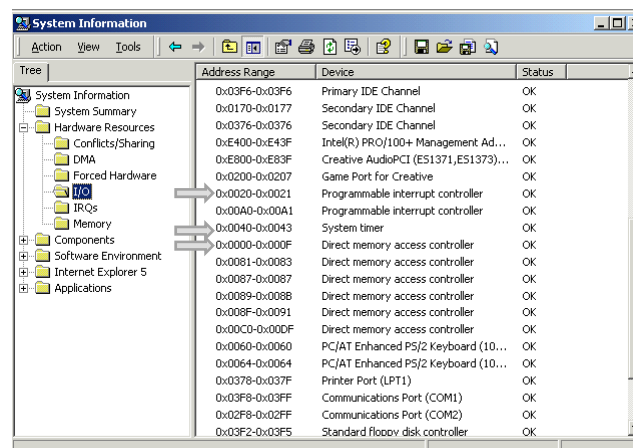
Conexión por puerto o memoria

- Para realizar una operación de E/S, la UCP tiene que escribir en los registros del controlador.
- Cada registro del controlador. una dirección de E/S o de memoria asignada.
- El modelo de **dispositivos por puertos** es clásico en las arquitecturas de Intel.
 - Cuando se instala un dispositivo, a su controlador se le asigna un puerto de E/S, una interrupción hardware y un vector de interrupción.
 - Una operación de E/S la UCP ejecuta operaciones `portin` o `portout`
- El modelo **proyectado en memoria** asigna a cada registro de E/S una dirección de MP.
 - No hay instrucciones específicas de E/S, se usan las instrucciones de acceso a memoria.
 - Mapa único de direcciones de memoria.

Ubicación de puertos de E/S en PCs (parcial) I

I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)

Ubicación de puertos de E/S en PCs (parcial) II

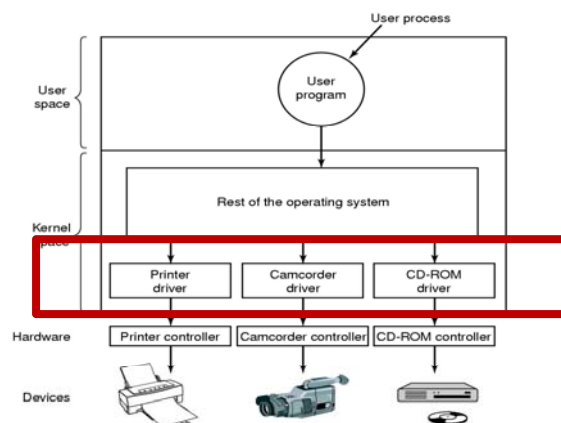


Address Range	Device	Status
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK
0xE400-0xE43F	Intel(R) PRO/100+ Management Ad...	OK
0xE800-0xE83F	Creative AudioPCI (ES1371, ES1373)...	OK
0x0200-0x0207	Game Port for Creative	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x0040-0x0043	System timer	OK
0x0000-0x000F	Direct memory access controller	OK
0x0081-0x0083	Direct memory access controller	OK
0x0087-0x0087	Direct memory access controller	OK
0x0089-0x008B	Direct memory access controller	OK
0x008F-0x0091	Direct memory access controller	OK
0x00CD-0x00DF	Direct memory access controller	OK
0x0060-0x0060	PC/AT Enhanced PS/2 Keyboard (10...	OK
0x0064-0x0064	PC/AT Enhanced PS/2 Keyboard (10...	OK
0x0378-0x037F	Printer Port (LPT1)	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x02F8-0x02FF	Communications Port (COM2)	OK
0x03F2-0x03F5	Standard Floppy disk controller	OK

Entrada/Salida

- Caracterización de los dispositivos de E/S
- **Drivers: software de E/S dependiente del dispositivo**
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- Almacenamiento terciario
- El reloj
- El terminal
- La red
- Servicios de entrada/salida

Drivers de dispositivos



- Posiciones lógicas de los drivers de dispositivo
- La communications entre drivers y controladores es mediante el bus

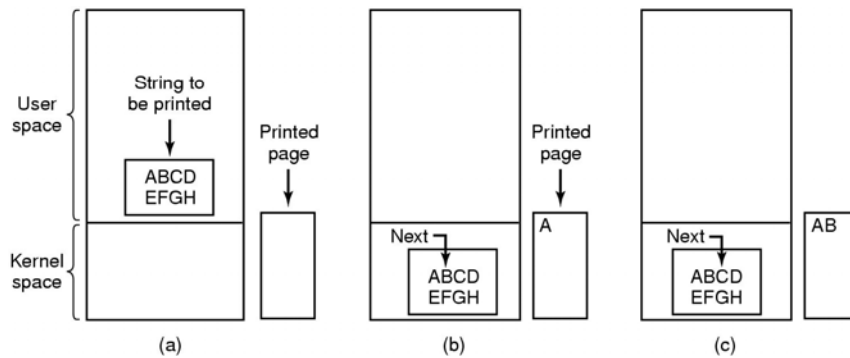
Tipos de E/S

- Programada
 - Espera activa
- Por interrupciones
 - Asociada a la multiprogramación
- Por DMA
 - Para dispositivos de bloques

E/S Programada I

- Se determina el estado de dispositivo:
 - ready
 - busy
 - Error
- Si está **busy** se espera a que esté **ready** (espera activa)
- Se realiza la operación de E/S

E/S Programada II



Pasos para imprimir un string

19

E/S Programada III

```

copy_from_user(buffer, p, count);          /* p is the kernel bufer */
for (i = 0; i < count; i++) {               /* loop on every character */
    while (*printer_status_reg != READY);    /* loop until ready */
    *printer_data_register = p[i];           /* output one character */
}
return_to_user();

```

Escribiendo un string en la impresora mediante E/S programada

20

E/S por interrupciones I

- La CPU tiene una/s **línea/s de petición de interrupción** conectada a los dispositivos de E/S
- El **manejador de interrupción** recibe la interrupción
- **Enmascarable** para ignorar or postponer algunas interrupciones
- El vector de Interrupciones elige el manejador de interrupción correspondiente
 - Basado en la prioridad de la interrupción
 - Algunas son **no-enmascarables**
- El mecanismo de interrupciones también se usa para excepciones

E/S por interrupciones II

```
copy_from_user(buffer, p, count);
enable_interrupts();
while (*printer_status_reg != READY) ;
*printer_data_register = p[0];
scheduler();
```

(a)

```
if (count == 0) {
    unblock_user();
} else {
    *printer_data_register = p[i];
    count = count - 1;
    i = i + 1;
}
acknowledge_interrupt();
return_from_interrupt();
```

(b)

- **Escribiendo un string en la impresora mediante E/S por interrupciones**
 - El código se ejecuta cuando se invoca la llamada al sistema de la impresora
 - Rutina de servicio de interrupción

22

Manejadores de interrupciones

- ❑ El driver inicia una operación de E/S y se bloquea hasta que una interrupción notifica su finalización
- ❑ La función de interrupción hace la tarea
 - Cuando termina desbloquea el driver que la inició
- ❑ Las rutinas de interrupción suelen tener dos partes:
 - una genérica y
 - otra particular para el dispositivo

23

Rutina de interrupción

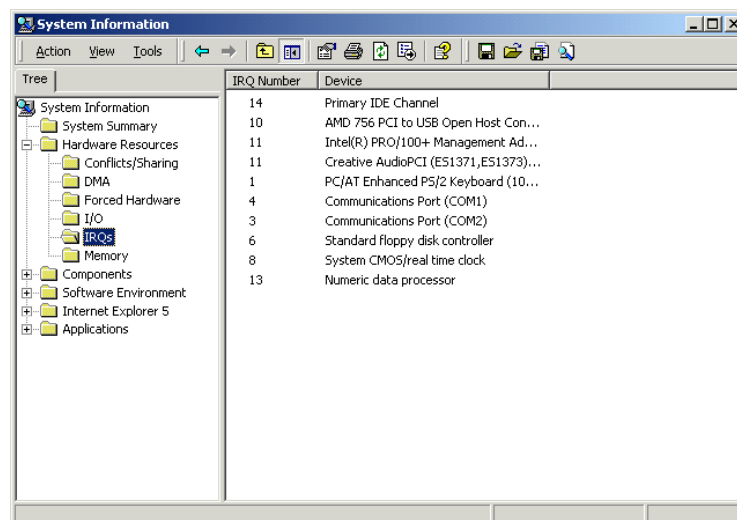
la parte genérica hace lo siguiente:

1. Captura de la interrupción.
2. Salvaguarda de estado del procesador.
3. Activación de la rutina particular de servicio de la interrupción.
4. Planificación como lista para ejecutar la rutina particular.
5. Desactivación de la interrupción.
6. Restauración del estado del procesador.
7. Cesión de control (RETI).

Tabla de excepciones del procesados Pentium

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19-31	(Intel reserved, do not use)
32-255	maskable interrupts

Interrupciones que usan algunos controladores

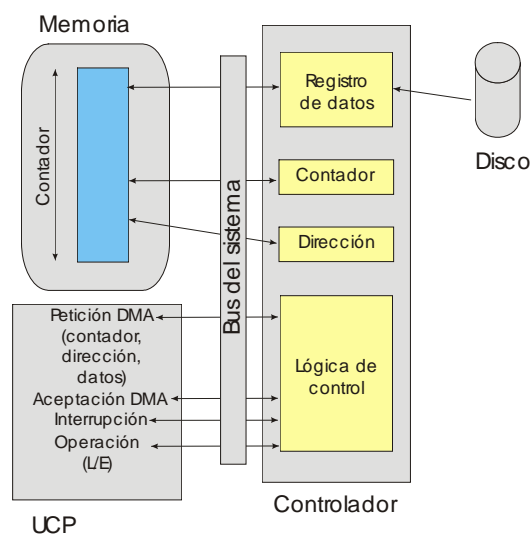


IRQ Number	Device
14	Primary IDE Channel
10	AMD 756 PCI to USB Open Host Con...
11	Intel(R) PRO/100+ Management Ad...
11	Creative AudioPCI (ES1371, ES1373)...
1	PC/AT Enhanced PS/2 Keyboard (10...
4	Communications Port (COM1)
3	Communications Port (COM2)
6	Standard floppy disk controller
8	System CMOS/real time clock
13	Numeric data processor

E/S mediante DMA I

- Se usa para evitar la E/S programada de grandes bloques de datos
- Necesita un controlador con DMA
- Evita el uso de la CPU transfiriendo los datos directamente entre los dispositivos de E/S y la memoria
 - del usuario
 - del núcleo
- Básico para aprovechar la CPU en un sistema multiprogramado, ya que libera tiempo de la CPU que se puede usar para ejecutar otros programas.

E/S mediante DMA II



E/S mediante DMA III

```
copy_from_user(buffer, p, count);
set_up_DMA_controller();
scheduler();
```

(a)

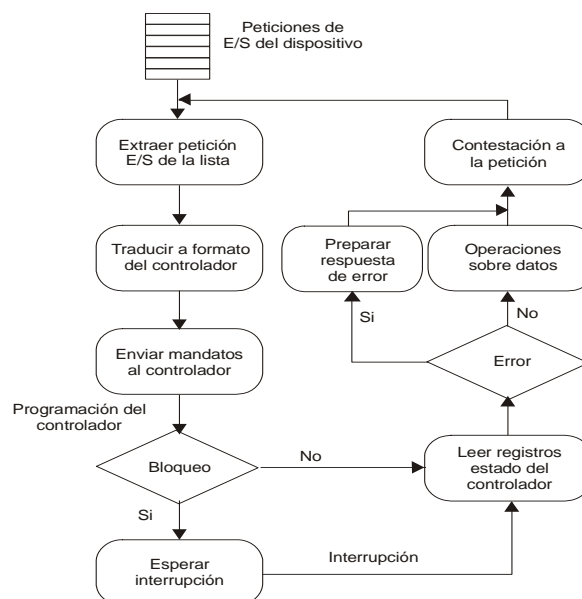
```
acknowledge_interrupt();
unblock_user();
return_from_interrupt();
```

(b)

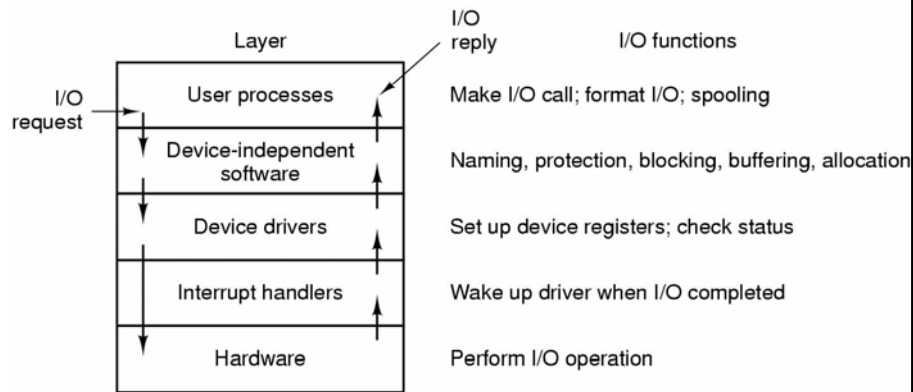
- **Imprime un string usando DMA**
 - Utilizado para programar E/S para grandes movimientos de datos
 - Requiere un controlador DMA
 - Transfiere los datos directamente entre los dispositivos de E/S a MP

29

Operaciones en un manejador (driver)



Flujo del software de E/S



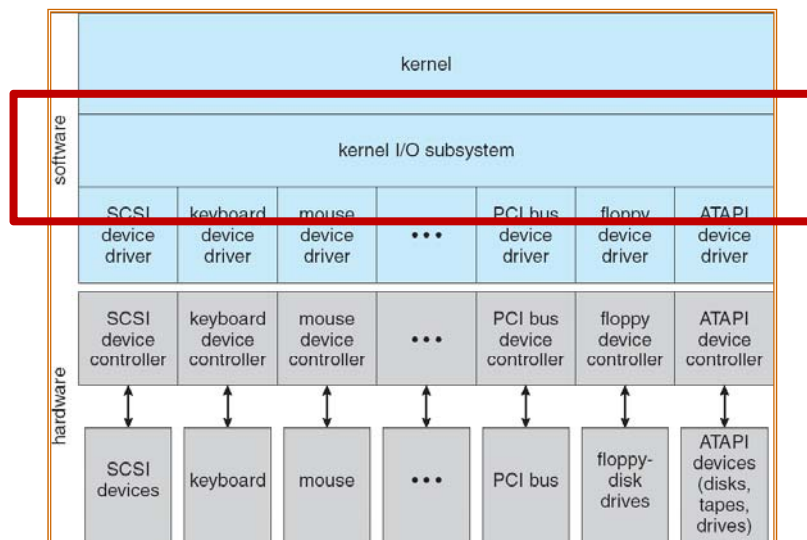
Niveles del sistema de E/S y principales funciones de cada nivel

31

Entrada/Salida

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- **Software de E/S independiente del dispositivo**
- Almacenamiento secundario
- Almacenamiento terciario
- El reloj
- El terminal
- La red
- Servicios de entrada/salida

SW de E/S independiente de dispositivo



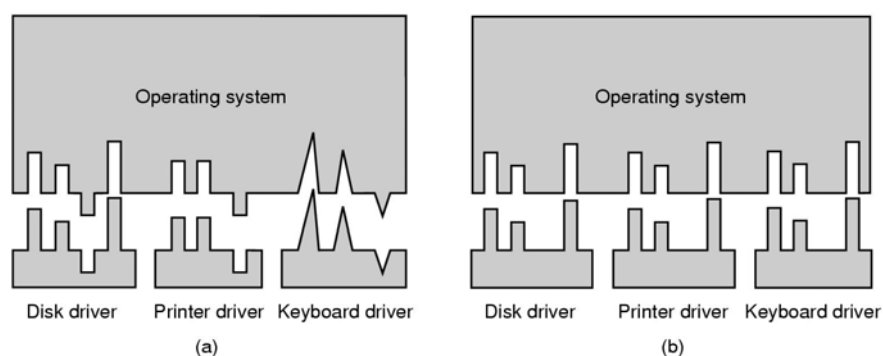
Funciones del sw de E/S independiente del dispositivo I

- **Interfaz uniforme para los dispositivo**
 - Tamaño de acceso: bloques, flujo de caracteres
 - Proporcionar un tamaño de bloque independiente del dispositivo
 - Almacenamiento intermedio (buffering): almacena datos en memoria mientras se transfieren entre dos dispositivos
 - evita accesos repetidos
 - oculta diferencias de velocidad
 - copia única (semántica de compartición)
 - Gestión de dispositivo
 - nombrado [/dev/cdrom → inodo → (mayor, menor)]
 - protección (rwx)
 - asignación y liberación de dispositivos
 - acceso en serie (si no se puede compartir)

Funciones del sw de E/S independiente del dispositivo II

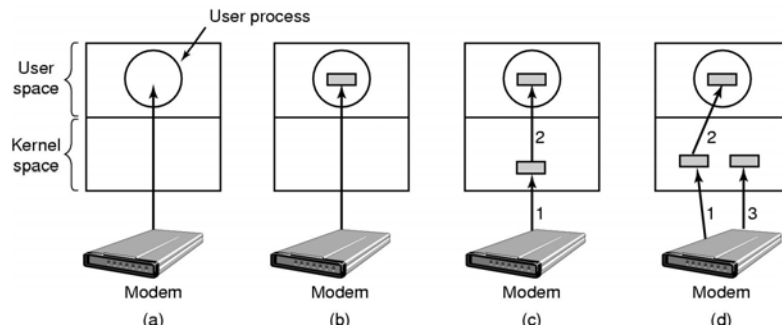
- **Planificación de E/S (Scheduling)** – encola las peticiones de E/S en colas; una por dispositivo; genérica (ej., FIFO) y de manejador (ej., SCAN)
- **Gestión de errores**
 - Transitorios: resueltos por el driver (reparación)
 - Permanentes: notificados al usuario y registrados
- **Buffering Caching** – mantiene una copia en memoria muy rápida
 - Performance
- **Spooling** – retiene la salida de un dispositivo
 - Si el dispositivo únicamente puede servir una petición a la vez
 - ej., la impresora
- **Device reservation** – proporciona acceso exclusivo al servicio
 - Llamadas al sistema para la asignación y liberación
 - Problema con interbloqueos

Independencia del dispositivo I



(a) Sin interfaz de E/S estandar
(b) Con interfaz de E/S estandar

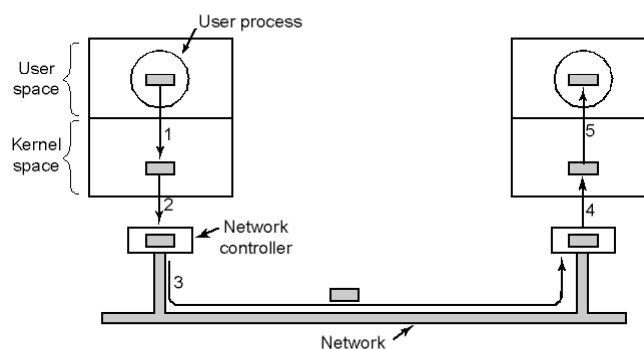
Independencia del dispositivo II



- (a) Sin buffer
 (b) Con buffer en espacio de usuario
 (c) Con buffer en el kernel y copia en espacio de usuario
 (d) Double buffer en el kernel

37

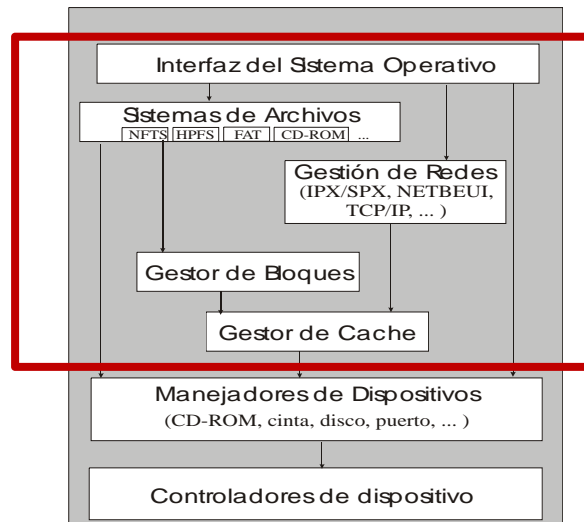
Independencia del dispositivo



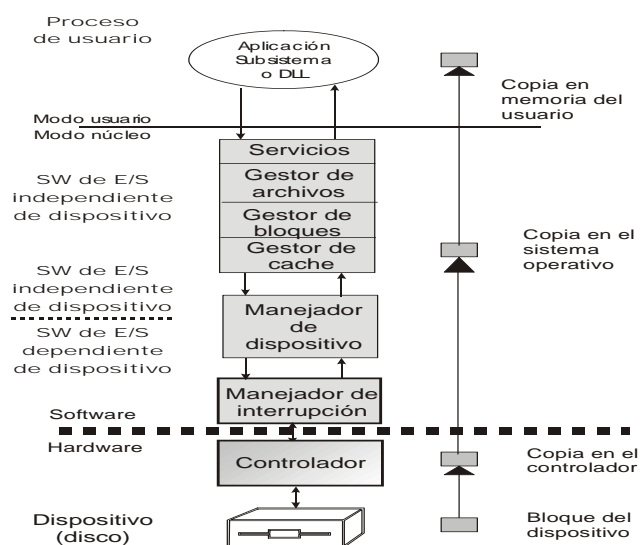
Una transferencia por red puede requerir varias copias

38

Capas del sistema de E/S



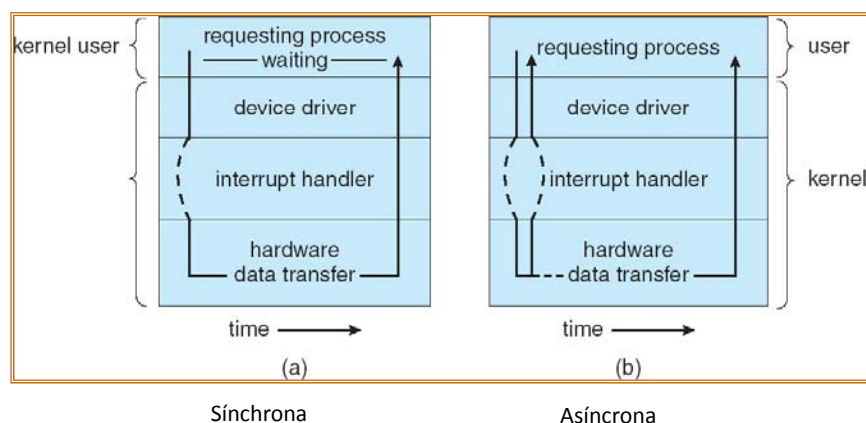
Flujo de una operación de E/S



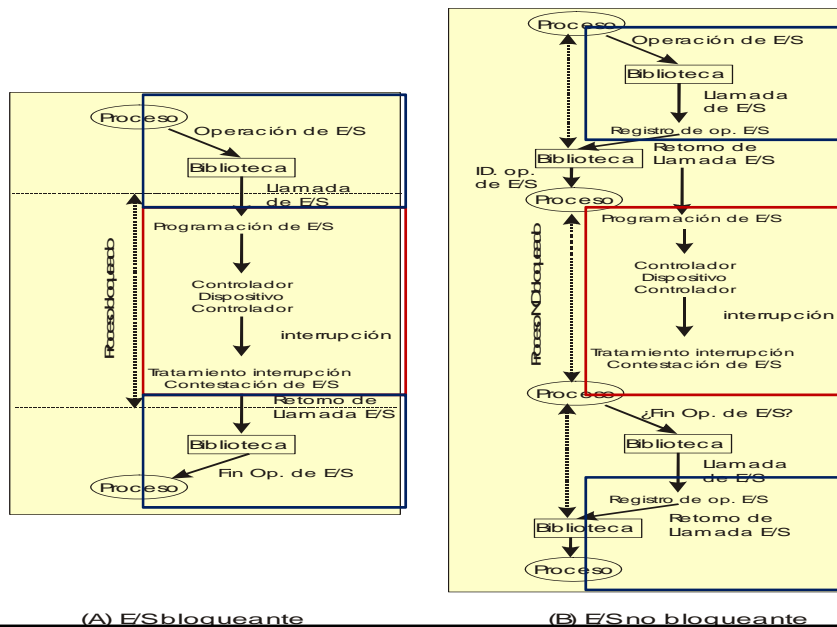
E/S Bloqueante y No-bloqueante I

- **Bloqueante:** procesos suspendidos hasta que la E/S termina
 - Fácil de usar y comprender
 - Insuficiente para algunos requisitos
 - Ejemplo: **read**, **write**, ... de POSIX
- **No bloqueante:** la llamada de E/S vuelve inmediatamente
 - Interfaz de usuario más complejo y espacio para copia de datos
 - Vuelve rápidamente con el estado de la operación
 - Se puede implementar con threads
- **Asíncronas:** los procesos ejecutan mientras la E/S se lleva a cabo
 - Difícil de usar por operaciones pendientes
 - El sistema de E/S envía en una señal el fin de operación pendiente
 - Interfaz para comprobar estado y esperar (**aiowait**)

E/S Bloqueante y No-bloqueante II



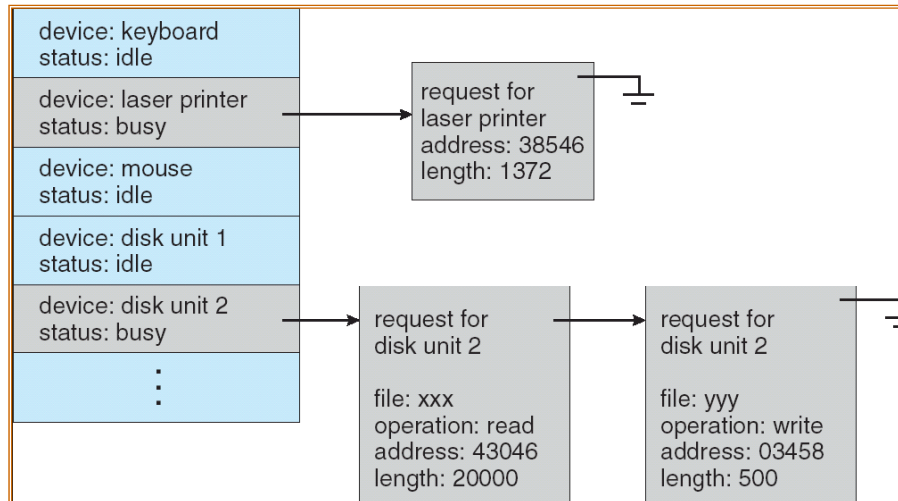
E/S Bloqueante y No-bloqueante III



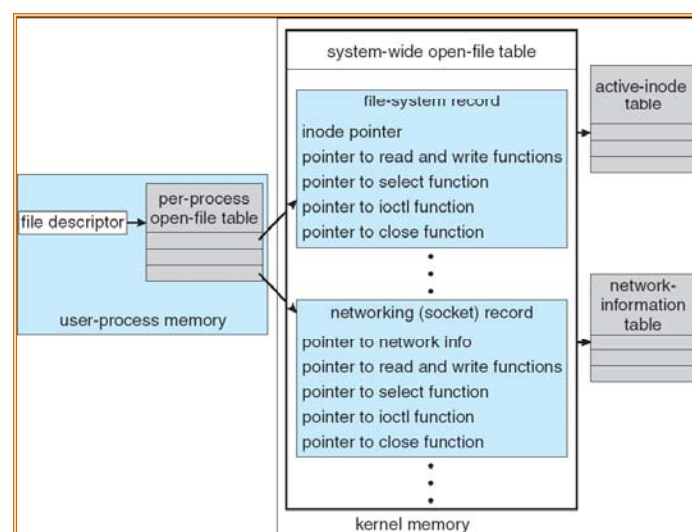
Estructuras de datos del Kernel de E/S

- El kernel guarda información de estado de los dispositivos de E/S, incluyendo tablas de ficheros abiertos, conexiones de red, estado de los dispositivos de caracteres.
- Muchas estructuras de datos complejas para buffers, ubicación de memoria, bloques "sucios".

Tabla de estado de dispositivos



Estructura del kernel de E/S en UNIX



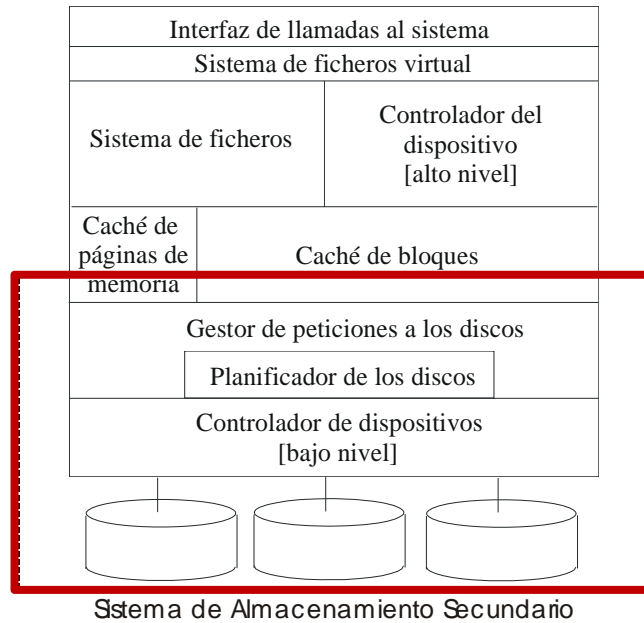
Entrada/Salida

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- **Almacenamiento secundario**
- Almacenamiento terciario
- El reloj
- El terminal
- La red
- Servicios de entrada/salida

Estructura

- El sistema de almacenamiento secundario se usa para guardar los programas y datos en dispositivos rápidos, de forma que sean fácilmente accesibles a las aplicaciones a través del sistema de archivos. Esa es la razón por la que en la jerarquía de E/S los dispositivos de almacenamiento secundario se colocan justo debajo de la memoria RAM.
- Hay dos elementos principales involucrados en este sistema:
 - Discos. El almacenamiento secundario se lleva a cabo casi exclusivamente sobre discos, por lo que es interesante conocer su estructura y cómo se gestionan.
 - Manejadores de disco. Controlan todas las operaciones que se hacen sobre los discos, entre las que son especialmente importantes las cuestiones de planificación de peticiones a disco.

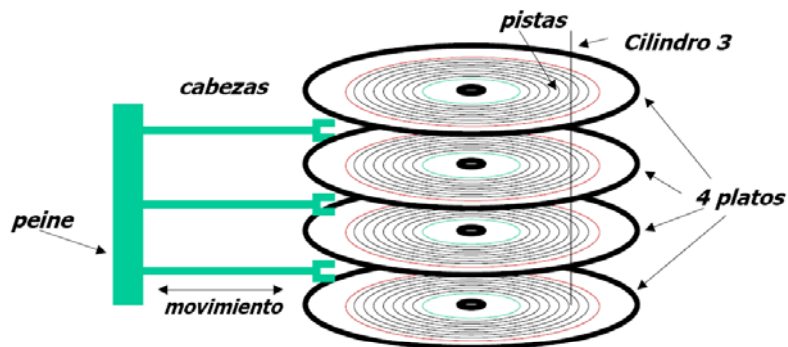
Estructura del sistema de E/S en LINUX



Discos

- Dispositivos básicos para llevar a cabo almacenamiento masivo y no volátil de datos. Además se usan como plataforma para el sistema de intercambio que usa el gestor de memoria virtual. Son dispositivos electromecánicos (HARD DISK) u optomecánicos (CD-ROM y DVD), que se acceden a nivel de bloque lógico por el sistema de archivos y que, actualmente, se agrupan en:
 - dos tipos básicos, atendiendo a la interfaz de su controlador:
 - Dispositivos SCSI (*Small Computer System Interface*).
 - Dispositivos IDE (*Integrated Drive Electronics*).
 - tres tipos básicos atendiendo a su tecnología de fabricación:
 - Discos duros (*Winchester*).
 - Discos ópticos.
 - Discos extraíbles.
- Independientemente del tipo al que pertenezcan, las estructuras física y lógica de todos los discos son muy similares, como se muestra a continuación.

Organización del disco duro



Pista: cada corona circular que se encuentra sobre una superficie.

Sector: divisiones físicas de tamaño fijo en que se considera dividida una pista. Típicamente contienen 512B

Cilindro: conjunto de pistas, una por cada superficie, al que se accede con una posición fija del peine de cabezas. Son como **los anillos** que marcan la edad de un árbol.

Organización del disco duro III



Los **sectores** son como los **pedazos de una pizza**.

Una **pista** es **uno de los anillos** que solo se encuentran en **un plato del HD**.

En la imagen podemos apreciar un HD viejo, con **20 pistas y 16 sectores**.

Características de los discos I

- Un disco duro es un dispositivo de gran capacidad compuesto por varias superficies magnetizadas y cuyas cabezas lectoras funcionan por efecto electromagnético.
- En el ámbito organizativo, las superficies del disco están divididas en cilindros, con una pista para cada cabeza y un cierto número de sectores por pista. El tamaño del sector es 512 bytes.
- Capacidad del disco:
 - **$Capacidad = cilindros * pistas * sectores * tamaño\ sector$**
- Los parámetros de la estructura física del disco son muy importantes para el manejador del mismo.
- Las operaciones de E/S se calculan y optimizan mediante dos parámetros fundamentales:
 - **tiempo de búsqueda** (lo que se tarda a ir de una pista a otra) y
 - **tiempo de latencia** (el tiempo medio que se tardan a llegar los datos debajo de las cabezas, una vez posicionadas en la pista)
 - **$T_{acceso} = n * T_{búsqueda} + T_{latencia} + T_{transferencia}$**

Características de los discos III

- **Densidad de cada pista.** En los discos modernos las pistas tienen distinta densidad, por lo que el número de sectores varía de unas pistas a otras.
- **Intercalado de sectores.** El controlador debe emplear tiempo en copiar los datos leídos a memoria principal. Durante este tiempo no puede estar transfiriendo datos del disco, que se sigue moviendo a velocidad constante, lo que significa que si quiere leer el sector siguiente deberá esperar a que las cabezas lectoras den una vuelta y lleguen de nuevo a ese bloque.
- **Almacenamiento intermedio en el controlador.** Se puede optar por leer la pista entera y mantenerla en memoria del controlador.
- **Controladores inteligentes.** Cuando controlan varios dispositivos, permiten efectuar operaciones de búsqueda de forma solapada.

Características de los discos II

Ejemplo de disco duro



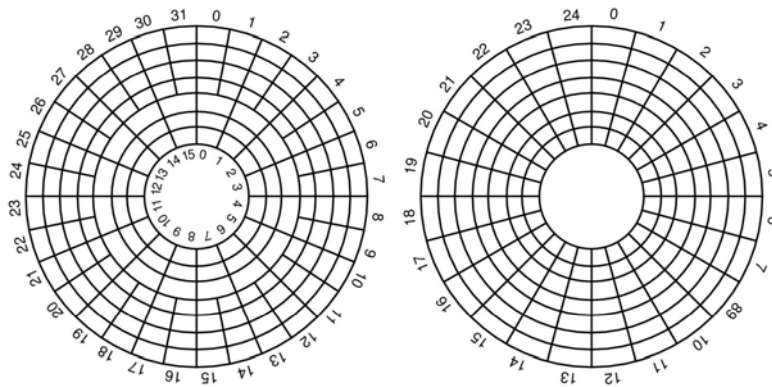
SEAGATE Barracuda ATA II ST330630A

Parámetro	Valor
Capacidad:	10.2 GB
Cilindros:	1023
Cabezas:	256
Sectores:	83
Velocidad:	7200 RPM
T Búsqueda:	8,5 mseg.
Latencia:	4,16 mseg
Memoria:	2 Mbytes

Estructura de disco I

- Los discos se direccionan como un gran array 1-dimensional de *bloques lógicos*, donde el bloque lógico es la unidad de transferencia más pequeña.
- Los bloques lógicos del array 1-dimensional se corresponden con sectores secuenciales en el disco
 - El sector 0 es el primer sector de la primera pista del cilindro más externo.
 - En el sector 0 se guarda la *tabla de particiones*.
 - Particiones activas o de sistema: permiten arrancar desde un sistema operativo.
- La correspondencia se realiza en orden alrededor de la pista, el resto de pistas y el resto de cilindros desde el más externo al más interno. El manejador de disco no sabe nada de la organización de los ficheros, sólo de particiones y bloques.

Estructura de disco II



- Geometría física de un disco con 2 zonas (176 sectores)
- Una posible geometría virtual para este disco (175 sectores)

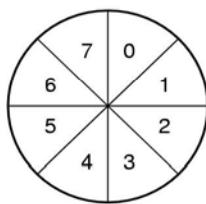
Gestión de disco

- **Formateo físico o de bajo nivel** — Divide al disco en sectores que el controlador de disco puede leer y escribir.
- **Formateo lógico o “construir el sistema de archivo”**, lleva a cabo otras tres tareas fundamentales:
 1. Creación del **bloque de carga o bloque Boot que inicializa el sistema**. incluye el programa de arranque del computador, programa que se activa cuando se conecta la corriente eléctrica y se produce un RESET.
 2. Creación de una lista de bloques defectuosos.
 3. Creación de una lista de bloques de repuesto.

Formateo de disco I

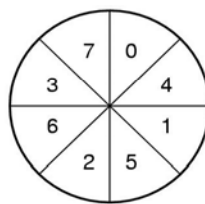


Un sector de disco



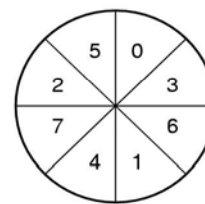
(a)

a) No intercalados



(b)

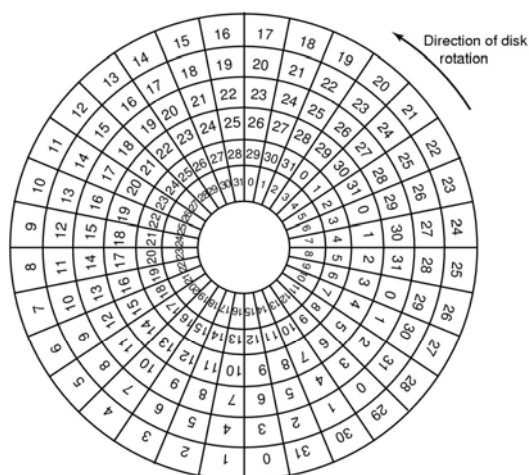
b) Intercalado sencillo



(c)

c) Intercalado doble

Formateo de disco II



Decalaje de cilindros

Creación de la estructura lógica I

- El **bloque de carga** incluye el programa que se carga en RAM (núcleo del sistema operativo) cuando se enciende el ordenador.
 - El **programa cragador Bootstrap se almacena en ROM**, incluye solo los aspectos más básicos de iniciación del hardware del sistema y un bucle de lectura que carga el sistema operativo en una dirección de la memoria.

– Lee palabras consecutivas de la entrada y las coloca de forma consecutiva en MP a partir de una dirección determinada.

Address	Content	Purpose.
000		17001 Read a word from the input into the following memory location.
001	17002	Execute what was just read above.
002	12000	Go back and do it again.

- Cuando ha terminado salta a la dirección de memoria donde puso
- La **lista de bloques defectuosos**, incluye los bloques en mal estado.
 - Se marcan siempre como ocupados y no se liberan nunca.
 - Un bloque es defectuoso porque alguno de los sectores que lo componen son defectuosos. Se sabe por la paridad del sector introducida durante el formato físico.
- **Lista de bloques de repuesto** para el caso en que algún bloque del dispositivo falle durante el tiempo de vida del mismo.

Creación de la estructura lógica II

- Para facilitar el uso de una partición de distintas formas , la utilidad **fdisk** o **format** permite crear dos imágenes asociadas a cada partición:
 - Dispositivo de bloques. Todos los accesos al mismo pasan por el sistema de archivos y la cache de bloques.
 - Dispositivo de caracteres. Permite acceder a bloques de la partición directamente.
 - Las peticiones deben ser forzosamente múltiplo del tamaño de bloque
 - Deben estar alineadas al principio de un bloque.

Tabla de particiones de un disco

```
[root@cuervo /root]# fdisk /dev/hda

The number of cylinders for this disk is set to 1583.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., LILO)
 2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1583 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1           31       248976   82  Linux swap
/dev/hda2             32        1583     12466440   85  Linux extended
/dev/hda5             32           35        32098+   83  Linux
/dev/hda6             36           557     4192933+   83  Linux
/dev/hda7            558        1583     8241313+   83  Linux

Command (m for help):
```

Manejadores de disco I

Funciones principales:

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

Manejadores de disco II

- El paso 1 se lleva a cabo en un manejador para la clase de dispositivo disco independiente del tipo de disco en particular, denominado **manejador genérico**.
- El paso 2 se lleva a cabo en un manejador dependiente del dispositivo, denominado **manejador particular**.
 - El tipo de dispositivo y el dispositivo en particular se distinguen porque en la petición de E/S viene información que lo indica.
En UNIX y LINUX *major number* y *minor number*
- El paso 3 depende del diseño del manejador:
 - Cola global para cada tipo de dispositivo
 - Cola para dispositivo particular
 - Ambas.

Planificación de disco I

- El SO es responsable del uso eficiente del Hw— en un disco, esto es el tiempo de acceso mínimo y el mayor ancho de banda.
- Mejorar el tiempo de acceso:
 - Minimizar el **tiempo de posicionamiento o búsqueda**
-> distance recorrida en el posicionamiento
 - Minimizar la **latencia de rotación** (tiempo de espera hasta que el sector deseado está debajo de la cabeza)
- Objetivo: minimizar el tiempo de búsqueda, que es directamente proporcional a la distancia de búsqueda
- El **tiempo de transferencia** tiene que ver con la velocidad de rotación del disco:
 - Ancho de banda: bytes transferidos / velocidad de transferencia
- El ancho de banda es el número total de bytes transferidos dividido entre el tiempo total entre la primera petición del servicio y la última transferencia.

Planificación de disco II

- Criterios de planificación:
 - Optimizar el tiempo de búsqueda (**SSOO**)
 - Dar servicio determinista (**multimedia y SSTR**)
- Existen diferentes algoritmos de planificación de peticiones de E/S a disco.

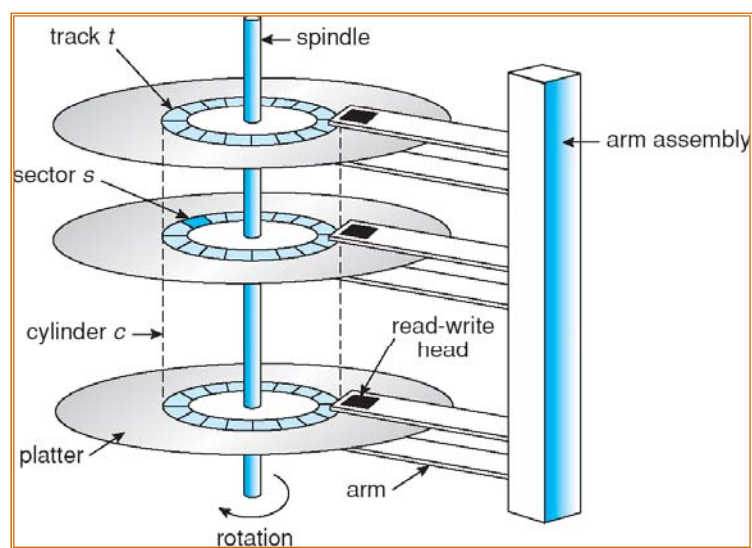
FCFS, SSTF, SCAN, C-SCAN, C-LOOK

- Por ejemplo para una cola de peticiones de 0 a 199:

98, 183, 37, 122, 14, 124, 65, 67

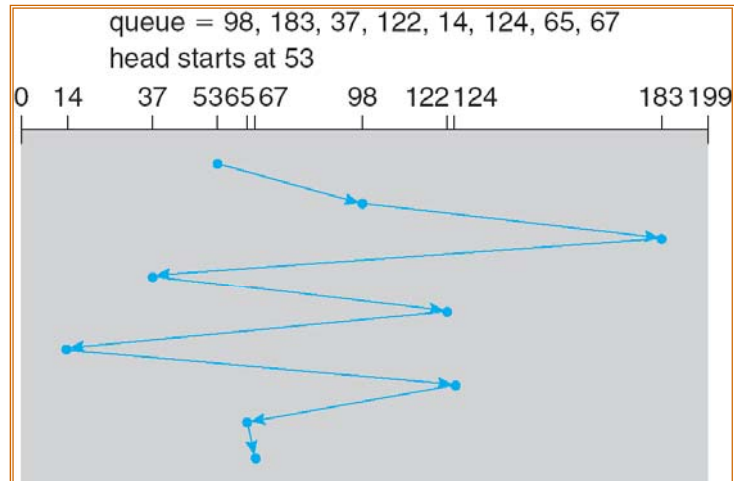
Head pointer 53

Mecanismo de movimiento de brazo de disco



FCFS (FIFO)

Vemos que la cabeza I/O atraviesa un total de **640** cilindros

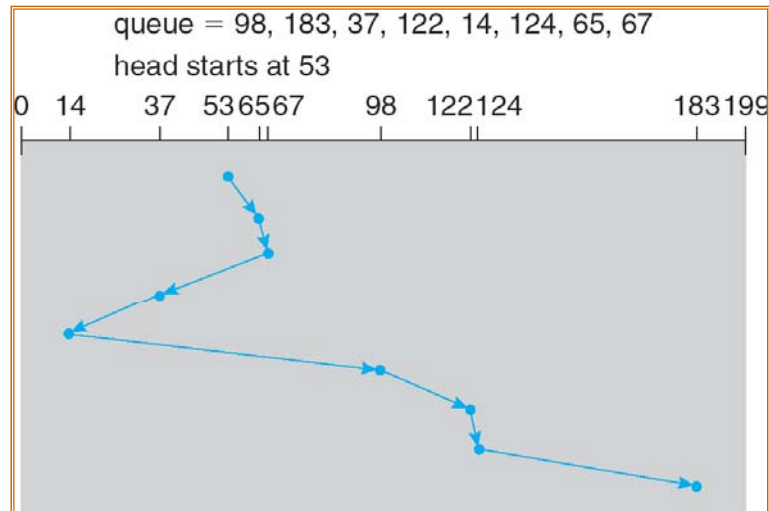


Shortest Seek First (SSTF) I

- Selecciona la petición con mínima distancia desde el posicionamiento actual.
- SSTF es una variante de planificación SJF (*Shortest Job First*); puede causar inanición (*starvation*) en algunas peticiones.
- Primero las peticiones que minimizan el movimiento de cabezas desde la posición actual.
- Idea: maximizar el ancho de banda del disco.

Shortest Seek First (SSTF) II

Vemos que la cabeza I/O atraviesa un total de **236** cilindros

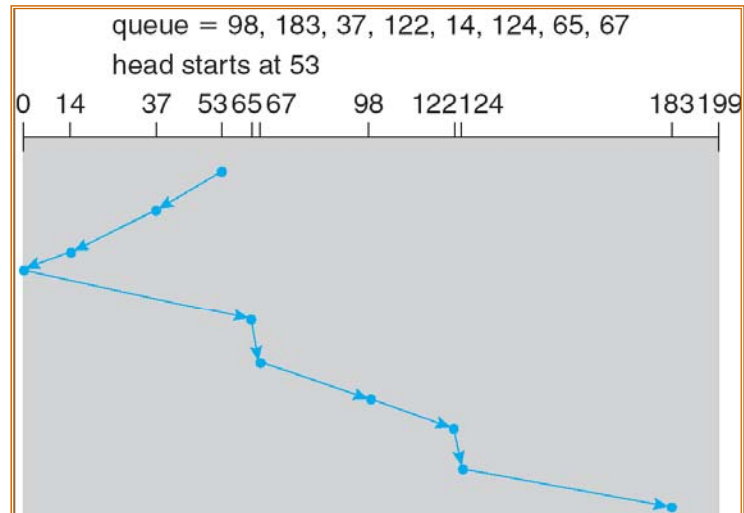


Ascensor (SCAN) I

- El brazo de disco se mueve hacia un extremo sirviendo peticiones en su recorrido; el sentido se invierte y continúa el servicio de peticiones hasta alcanzar el otro extremo.
- Idea: evitar desplazamientos atrás y adelante.
- Ventaja: combate el problema de inanición y homogeneiza las desviaciones típicas de las latencias.
- Problema: diferentes latencias según la posición; peor en los extremos, mejor en el centro.

Ascensor (SCAN) II

Vemos que la cabeza I/O atraviesa un total de **208** cilindros

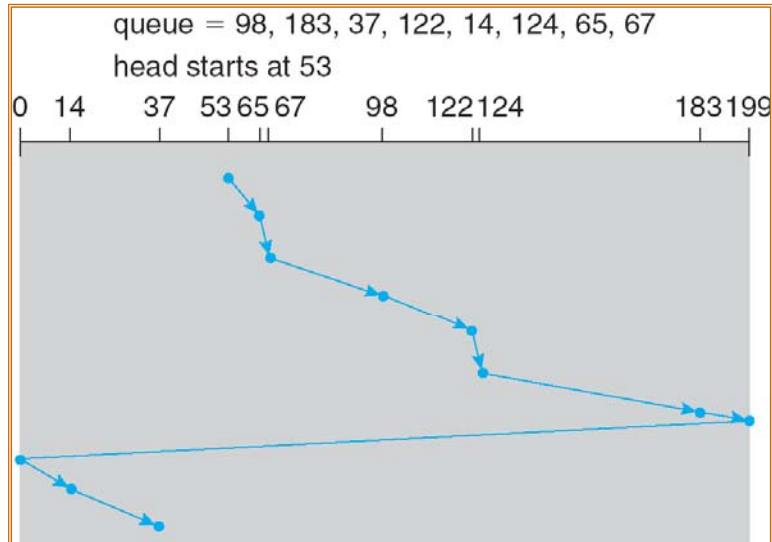


Ascensor cíclico (SCAN) I

- Proporciona un tiempo de espera más uniforme que el SCAN.
- El brazo de disco se mueve hacia un extremo sirviendo peticiones en su recorrido. Cuando alcanza el extremo final, inmediatamente se mueve al principio del disco sin servir peticiones, desde donde se mueve hacia el extremo final sirviendo las restantes peticiones.
- Trata los cilindros como una lista circular donde el siguiente al último cilindro es el primero.
- Se usa frecuentemente

Ascensor cíclico (SCAN) II

Vemos que la cabeza I/O atraviesa un total de **345 cilindros - 199**:
Va del último al primero en un único movimiento y tarda muy poco

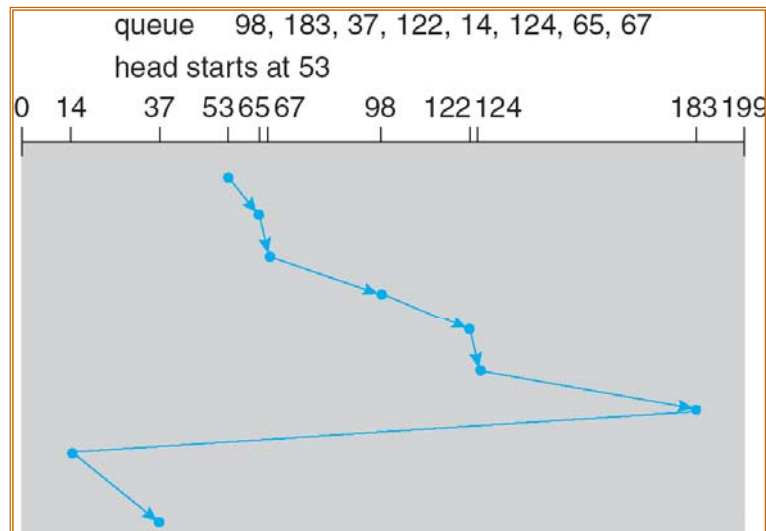


C-LOOK I

- Versión del C-SCAN
- El brazo únicamente va hasta la petición más extrema en cada dirección, sin llegar a los extremos del disco.

C-LOOK II

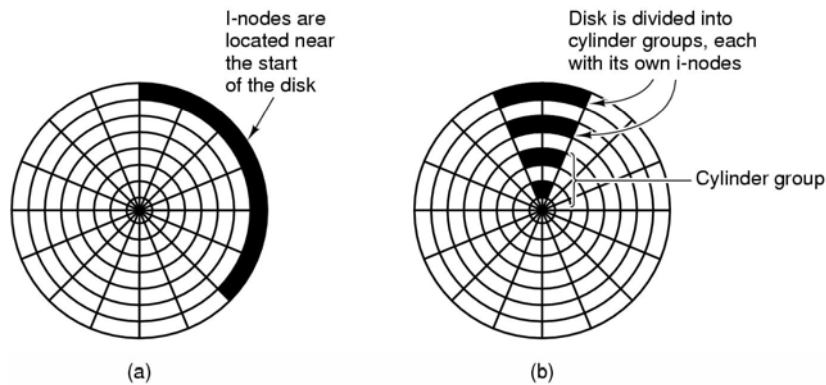
Vemos que la cabeza I/O atraviesa un total de ? cilindros



Selección de un algoritmo de planificación de disco

- SSTF es frecuente y parece el más natural
- SCAN y C-SCAN tienen mejor rendimiento para sistemas que usan mucho el disco .
- La eficiencia depende del número y tipo de peticiones.
- Las peticiones al servidor de disco están influenciadas por el método de ubicación de ficheros.
- El algoritmo de planificación de disco puede estar en un módulo separado del SO, permitiendo que se sustituya por otro algoritmo si es necesario.
- Algoritmo estándar en casi todos los sistemas: C-SCAN.
- SSTF y LOOK son alternativas razonables como algoritmo por defecto.

Eficiencia en el sistema de archivos



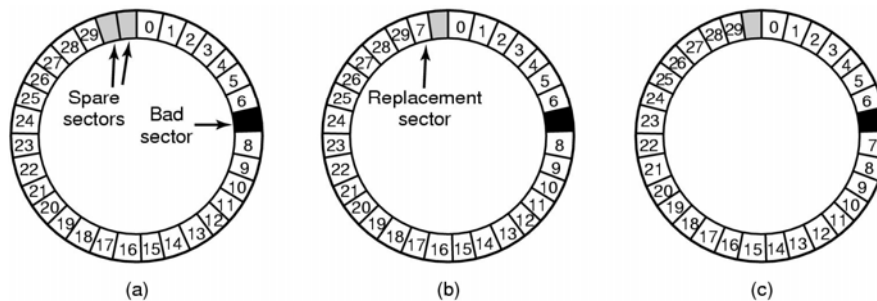
- I-nodes situados al inicio del del disco
- Disco dividido en grupos cilindros
 - Cada uno con sus propios bloques e i-nodos

79

Gestión de errores de disco

- Errores **transitorios**
 - Debidos a la existencia de partículas de polvo en la superficie del disco cuando se efectúa la operación de E/S, a pequeñas variaciones eléctricas en la transmisión de datos, fallos de calibración de cabezas, ...
 - Se detectan porque el ECC de los datos no coincide con el calculado y se resuelven repitiendo la operación de E/S.
 - Si después de un cierto número de repeticiones no se resuelve el problema, el manejador concluye que la superficie del disco está dañada y lo comunica al nivel superior de E/S.
- Errores **permanentes** se tratan de distintas formas.
 - Errores de aplicación: poco que hacer.
 - Errores del controlador: tratar de reiniciar el controlador.
 - Errores de superficie del dispositivo: sustituir el bloque por uno de repuesto.

Gestión de errores



- Una pista de disco con un sector defectuoso
- Se sustituye el sector malo por un *spare*
- Se desplazan los sectores para evitar el malo

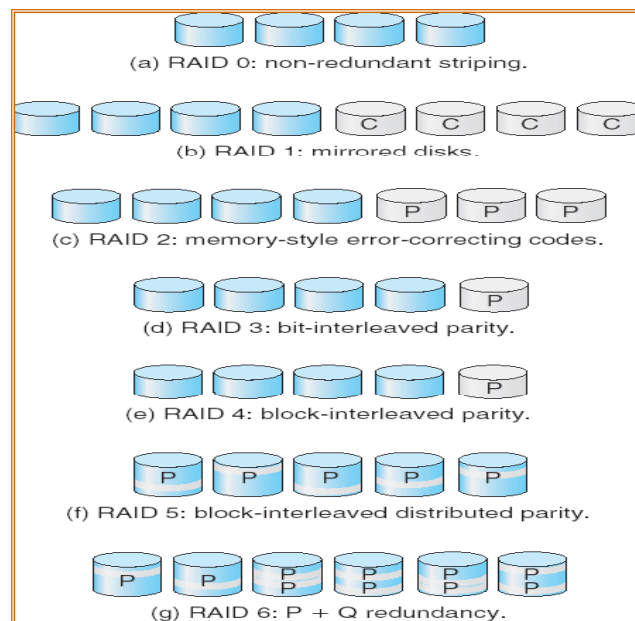
Fiabilidad y tolerancia a fallos

- El sistema de E/S es uno de los componentes del sistema con mayores exigencias de fiabilidad, debido a que se usa para almacenar datos y programas de forma permanente.
- Las principales técnicas usadas para proporcionar esta fiabilidad son:
 - Códigos correctores de error, como los existentes en las cabeceras y colas de los sectores.
 - Operaciones fiables, cuya corrección se puede verificar antes de dar el resultado de la operación de E/S. Esta técnica se implementa mediante técnicas de almacenamiento estable.
 - Redundancia, tanto en datos replicados como en código de paridad para detectar errores y recuperarlos. Esta técnica se implementa mediante el uso de dispositivos **RAID** (*Redundant Array of Inexpensive Disks*).
 - Redundancia hardware, como el sistema de Windows NT que permite conectar un disco a través de dos controladores.

RAID Structure

- RAID – Múltiples discos proporcionan fiabilidad via redundancia.
- Se utiliza un grupo de discos como una unidad de almacenamiento.
 - *Discos espejo (Mirroring or shadowing).*
 - *Bloques de paridad intercalados (menos redundancia).*
- Hay 6 niveles RAID:

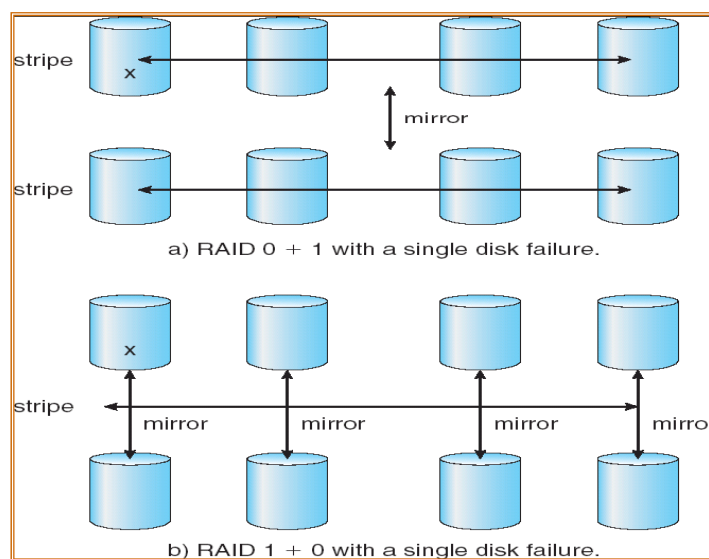
Niveles RAID

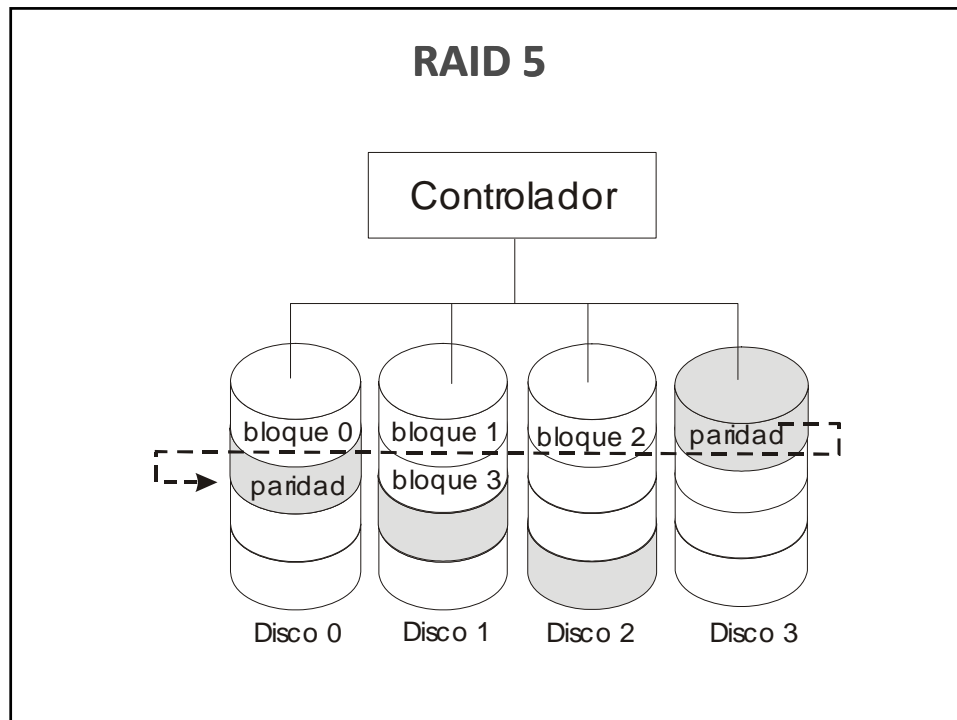


Organizaciones RAID

- **RAID 0:** la información se graba y se lee en paralelo entre varios discos físicos
- **RAID 1:** Dos discos en configuración “espejo”. Uno contiene los datos y el otro una réplica exacta
- **RAID 2:** Array de discos sincronizados, con varios discos redundantes que contienen el código de corrección Hamming
- **RAID 3:** Array de discos sincronizados, con un disco redundante que contiene el bit de paridad
- **RAID 4:** Array de discos independientes, con un disco de control de paridad dedicado
- **RAID 5:** Array de discos independientes, con control de paridad distribuido
- **RAID 6:** Array de discos independientes, con control de CRC distribuido

RAID (0 + 1) y (1 + 0)





Entrada/Salida

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- **Almacenamiento terciario**
- El reloj
- El terminal
- La red
- Servicios de entrada/salida

Dispositivos de almacenamiento terciario

- La característica que define el almacenamiento terciario es Low cost.
- Generalmente, el almacenamiento terciario es mobile:
 - Discos floppy, más rápidos que los discos duros y capacidad de **1 MB**.
 - CD-ROMs
 - Cintas magnéticas

Discos removibles

- Un disco magneto-óptico almacena datos en una superficie rígida de material magnético cubierta con una capa de plástico.
 - Una cabeza laser amplifica un campo magnético para grabar un bit.
 - La luz laser se utiliza también para leer los datos.
- Los discos ópticos no utilizan magnetismo, emplean materiales especiales que se alteran con la luz del laser.

Discos WORM

- **WORM ("Write Once, Read Many Times") se pueden escribir sólo una vez .**
- **Una capa de papel aluminio entre dos capas de plástico.**
- **Para escribir un bit, se quema con laser un pequeño agujero en el papel de aluminio del dispositivo; la información se puede destruir pero no alterar.**
- **Muy duraderos y fiables**
- **Los *Read Only* disks, como CD-ROM y DVD, vienen de fabrica con los datos pre-grabados.**

Cintas magnéticas

- **Fue el primer dispositivo de lamacenamiento secuendario.**
- **Comparada con los disos, es más barata y almacena más datos, pero el acceso aleatorio es mucho más lectos: ~1000 veces más lentoaque el disco.**
- **Típicamente almacena 20-200GB .**
- **Una vez que los datos están bajo la cabeza I/O, la velocidad de transferencia es comparable a la del disco.**
- **Se utiliza para backups, o para guardar datos que apenas se usan.**
- **Tecnologías comunes son 4mm, 8mm, 19mm, LTO-2 and SDLT**

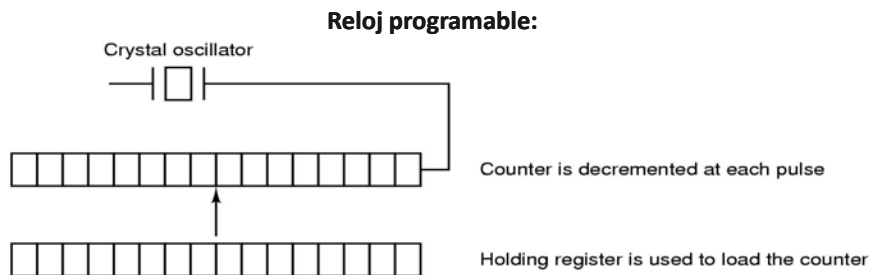
Entrada/Salida

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- Almacenamiento terciario
- **El reloj**
- El terminal
- La red
- Servicios de entrada/salida

El reloj

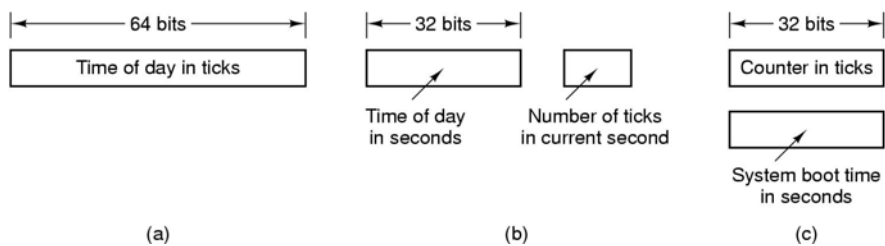
- **Varias acepciones:**
 - Reloj del procesador
 - Reloj del sistema (que mantiene fecha y hora)
 - Temporizador que activa periódicamente al S.O.
- S.O. vinculado con las dos últimas acepciones
- ¿Reloj es un dispositivo de E/S?
 - Sí, ya que implica registros de E/S e interrupciones
 - Generalmente el S.O. le da tratamiento específico

Reloj hardware



- **Circuito temporizador que genera señal periódica (*tick*)**
 - Conectado a línea de interrupción de alta prioridad
 - Frecuencia programable (actúa como divisor de frecuencias)
 - Modo operación programable (único disparo, onda cuadrada)
 - Generalmente circuito con múltiples temporizadores
 - No todos conectados a línea de interrupción
- **Reloj alimentado por batería que mantiene hora y fecha**
 - Consultado por S.O. en su arranque
 - También denominado reloj CMOS

El reloj Reloj software



Tres maneras distintas de mantener la hora del día

Driver de reloj

- Labor principal de SO: manejo de interrupciones de CK
 - También iniciación y gestión de llamadas relacionadas
- Compromiso al fijar frecuencia de interrupción:
 - Demasiado alta: Excesiva sobrecarga por tratamiento de int.
 - Demasiado baja: Limitada precisión en medida del tiempo
 - Ejemplo típico: 100 Hz (1 interrupción cada 10 ms)
- Minimizar el trabajo realizado por la rutina de interrupción
 - Ya que mientras no se atiende int de disp. de menos prioridad
- Solución típica: Dividir trabajo asociado a int. de reloj
 - Operaciones más urgentes realizadas por rutina de interr.
 - Resto: tratamiento posterior fuera de la rutina (*int. software*)

Driver de reloj

Funciones del manejador del reloj:

- Mantenimiento de fecha y hora
- Gestión de temporizadores
- Contabilidad y estadísticas
- Soporte para la planificación de procesos

Mantenimiento de fecha y hora

- En arranque SO programa temporizador y lee fecha
 - A partir de entonces se actualiza la hora en cada int CK
- Almacenamiento de fecha y hora:
 - Espacio suficiente
 - Debe poder seguir usándose en un futuro lejano
 - Diferencias entre horarios de países:
 - Se almacena en UTC y bibliotecas se encargan de conversión
 - Unidades de tiempo desde fecha fija en el pasado
 - UNIX: segs. o μ segs. desde 1-1-1970
 - Windows: centenas de nsegs. desde 1-1-1601
- Servicios para leer y cambiar hora (sólo superusuario)
 - Peligroso retrasar hora:
 - Algunos SS.OO. permiten hacerlo gradualmente

Gestión de temporizadores

- Algunos procesos necesitan esperar un plazo de tiempo
 - S.O. ofrece servicios para ello
- Propio S.O. también lo requiere
 - P.ej. módulo de comunicaciones o manejador de disquete
- Múltiples temporizadores a partir del temporizador HW
 - Lista de temporizadores activos:
 - Elemento: **plazo restante** (*ticks*) + **función a ejecutar** al cumplir
 - Posible organización de la lista:
 - Orden creciente y con plazos restantes relativos a anteriores
 - Ejemplo: *temp1 5 ticks, temp2 8 ticks y temp3 8 ticks*:
 - *[temp1 5] → [temp2 3] → [temp3 0]*
- Si la gestión del temporizador consume mucho tiempo- > ejecutada fuera de rutina de interrupción

Contabilidad y estadísticas

- Ejecución periódica de rutina de int. reloj
 - Permite muestreo de diversas variables
- Contabilidad de uso de procesador de cada proceso
 - Proceso que estaba ejecutando cuando int. de reloj
 - Se le carga uso de procesador en ese intervalo
 - Se distingue si estaba en modo usuario o sistema
- Perfiles de ejecución
 - Tiempo que consumen las distintas partes del programa
 - Rutina de int. Muestra el PC del proceso interrumpido
 - Se obtiene histograma de la ejecución del programa

Soporte para planificación de procesos

- Algoritmos de planificación basados en el tiempo
 - Rutina de int. CK conlleva acciones de planificación
- En *Round-Robin*:
 - En cada interrupción se descuenta tiempo a la rodaja
 - Cuando llega a cero → **Replanificación**
- Algoritmos que recalculan periódicamente la prioridad
 - P.ej. dependiendo de uso de UCP/proceso en último intervalo

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- Almacenamiento terciario
- El reloj
- **El terminal**
- La red
- Servicios de entrada/salida

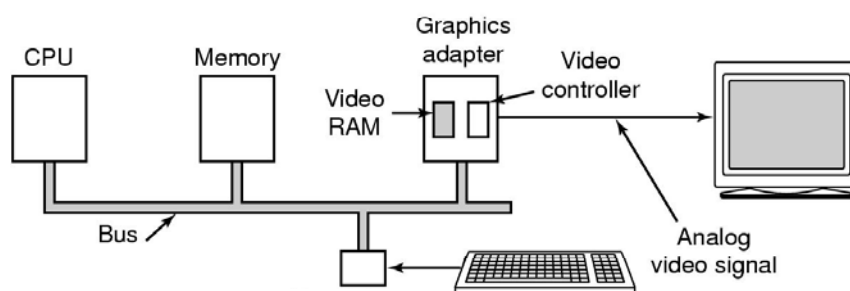
El terminal

- Teclado + pantalla
- Dependiendo de forma de conexión:
 - Terminales serie
 - Terminales proyectados en memoria
- Dependiendo de tipo de información manejada:
 - Terminales en modo alfanumérico (texto)
 - Memoria de vídeo contiene códigos ASCII
 - Controlador de vídeo genera patrones de pixels
 - Terminales en modo gráfico
 - Memoria de vídeo contiene matriz de pixels
 - SW debe transformar de código ASCII a patrón de pixels
- Exposición se centra en terminales en modo texto

Modo de operación del terminal

- Similar en todos los tipos de terminales
 - Diferencia: ¿Qué se hace por HW y qué por SW?
- Teclado genera interrupción al pulsar tecla
 - S.O. lee código de tecla de registro de controlador de teclado
 - Conversión a ASCII y manejo de teclas modif. por SW
- Salida para terminal mapeado en Memoria
 - Pantalla: matriz de pixels con memoria de vídeo asociada
 - Memoria de vídeo directamente accesible al procesador
 - Escritura en pantalla requiere escritura en memoria de vídeo
 - Controlador de vídeo lee de memoria de vídeo y refresca pantalla
 - Carácter ASCII → Patrón de pixels
 - Secuencias de escape:
 - Permiten operaciones especiales (p.ej. borrar la pantalla)

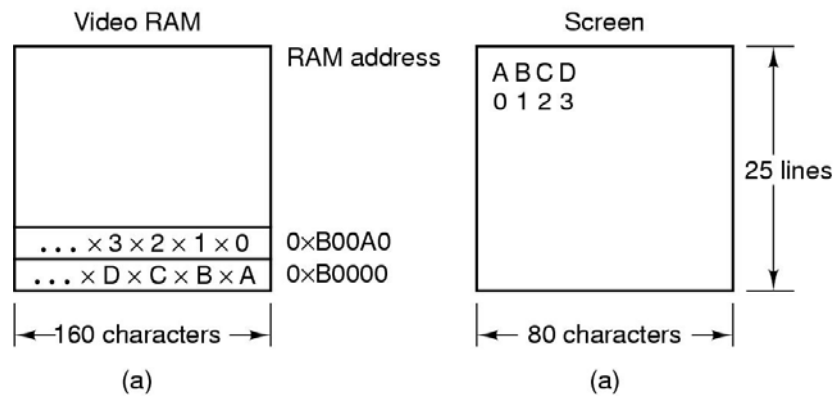
Hardware de terminal I



Terminal mapeado en Memoria

- El driver escribe directamente en la RAM de vídeo

Hardware del terminal II



- Una imagen de la RAM de video
 - Display sencillo monocromático
 - Modo carácter
- Correspondencia en la pantalla
 - los Xs son bytes de atributos

107

Entrada/Salida

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- Almacenamiento terciario
- El reloj
- El terminal
- **La red**
- Servicios de entrada/salida

La red

- Dada su creciente importancia, soporte de S.O. cada vez mejor
- Algunos S.O. le dan tratamiento diferente a otros dispositivos
 - En Linux no hay archivos en /dev para la red
- Software de red organizado en tres niveles:
 - Nivel de interfaz a las aplicaciones
 - Típicamente, *sockets* (*Winsock* en Windows) (capítulo 10)
 - Puede considerarse como nivel de sesión OSI
 - Nivel de protocolos
 - Capa(s) que implementa(n) transporte y red OSI (o TCP/IP)
 - Incluye funciones de encaminamiento
 - Nivel de dispositivo de red
 - Manejadores de dispositivos de red (nivel de enlace OSI)

Niveles del software de red

Niveles del software de red

Interfaz a las aplicaciones		
Nivel de transporte		
Nivel de red		
Ethernet	SLIP

- **Niveles trabajan de forma independiente**
 - Uso de interfaces estándar entre niveles
- **Mensaje desciende como resultado de llamada de aplicación**
 - Se va añadiendo información de control
- **Menaje asciende como resultado de interrupción de recepción**
 - Se va eliminando información de control
- **Para implementación eficiente, minimizar copia de información**

Entrada/Salida

- Caracterización de los dispositivos de E/S
- Drivers: software de E/S dependiente del dispositivo
- Software de E/S independiente del dispositivo
- Almacenamiento secundario
- Almacenamiento terciario
- El reloj
- El terminal
- La red
- **Servicios de entrada/salida**

Servicios de entrada/salida

- Servicios relacionados con el reloj
 - Servicios de hora y fecha
 - Para leer y cambiar (sólo superusuario) fecha y hora
 - Temporizaciones
 - Tanto síncronas (bloqueantes) como asíncronas
 - Servicios de contabilidad
 - Ejemplo: tiempo de procesador consumido por un proceso
- Servicios de entrada/salida
 - Acceso a dispositivos a través de interfaz de archivos
 - Programas independientes del medio que usan
 - Sólo específicos para operaciones dependientes de dispositivo.
 - Por ejemplo, desactivar eco de terminal

Servicios de fecha y hora en POSIX

*time_t time (time_t *t);*

- Devuelve número de segundos desde 1-1-1970 en UTC
- Algunos UNIX tienen *gettimeofday* con precisión de µsecs.
- Funciones de biblioteca para convertir a año, mes, ..., segundos
 - *gmtime* (UTC) y *localtime* (horario local)

*int stime (time_t *t);*

- Fija hora según parámetro (segundos desde 1-1-1970 en UTC).
- Sólo para superusuario
- Algunos UNIX tienen *settimeofday* con precisión de µsecs.

Servicios de temporización en POSIX

unsigned int alarm (unsigned int segundos);

- Establece temporizador del plazo especificado
- El proceso solicitante no se bloquea:
 - Cuando se cumple plazo recibe señal **SIGALRM**
- Sólo uno activo por proceso
 - Si había uno activo, lo desactiva y devuelve plazo pendiente
- Temporizadores más avanzados (**setitimer**):
 - Mayor resolución y diversos modos de operación

Servicios de contabilidad en POSIX

- Diversas funciones, se presenta una frecuentemente usada:
 - Obtiene tiempo de procesador consumido por un proceso

*clock_t times (struct tms *info);*

- Proporciona info. sobre tiempo de ejecución del proceso e hijos:
 - En parámetro de salida, tiempo en modo usuario y sistema consumido por proceso e hijos
 - Devuelve valor relacionado con tiempo real en el sistema
 - p. ej. *ticks* de reloj desde arranque del sistema
 - Este valor no se usa de forma absoluta
 - Se realizan dos llamadas y se calcula la diferencia