

# PRINCIPIOS DE SISTEMAS OPERATIVOS

## Gestión de E/S

# Contenidos

1. Introducción
2. Dispositivos de E/S
3. Problema de Diseño de SO
4. E/S en HD y SSD
5. Algoritmos de Planificación

# Introducción (I)

- La Entrada/Salida (E/S) es una unidad básica de la PC a través de la que se realiza la comunicación con el exterior.
  - Cargar programas, introducción de datos para ser procesados, visualizar resultados
- Tradicionalmente, ignorada en favor del diseño de la CPU

# Introducción (II)

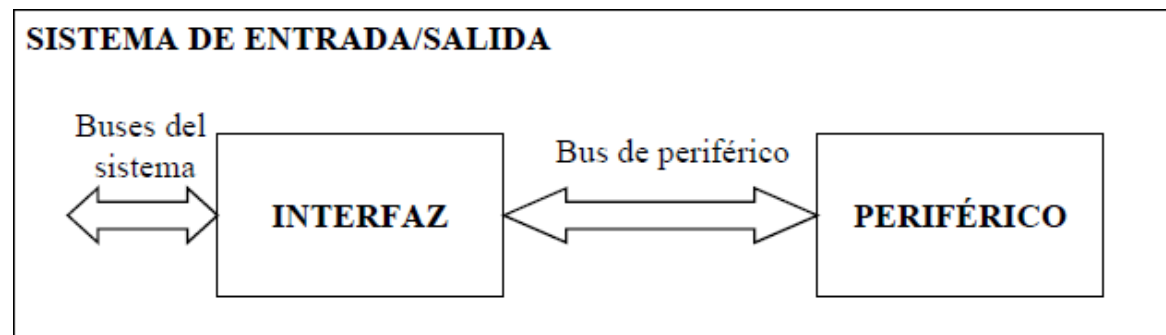
- Las E/S tiene un impacto muy significativo tanto en el rendimiento como en el **coste del sistema**
  - El diseño de la E/S, además del rendimiento, ha de contemplar aspectos de ***expansibilidad y recuperación ante fallos***.
  - Conforme las CPU's son más rápidas, tanto las máquinas grandes como pequeñas usan los mismos microprocesadores siendo entonces **la memoria y E/S características diferenciadoras de las mismas**.
  - El éxito de aplicaciones recientes (multimedia, internet,...) fuertemente relacionadas con el ***ancho de banda de la E/S***.
  - Las máquinas interactúan con las personas a través de su sistema de E/S.

# Introducción (III)

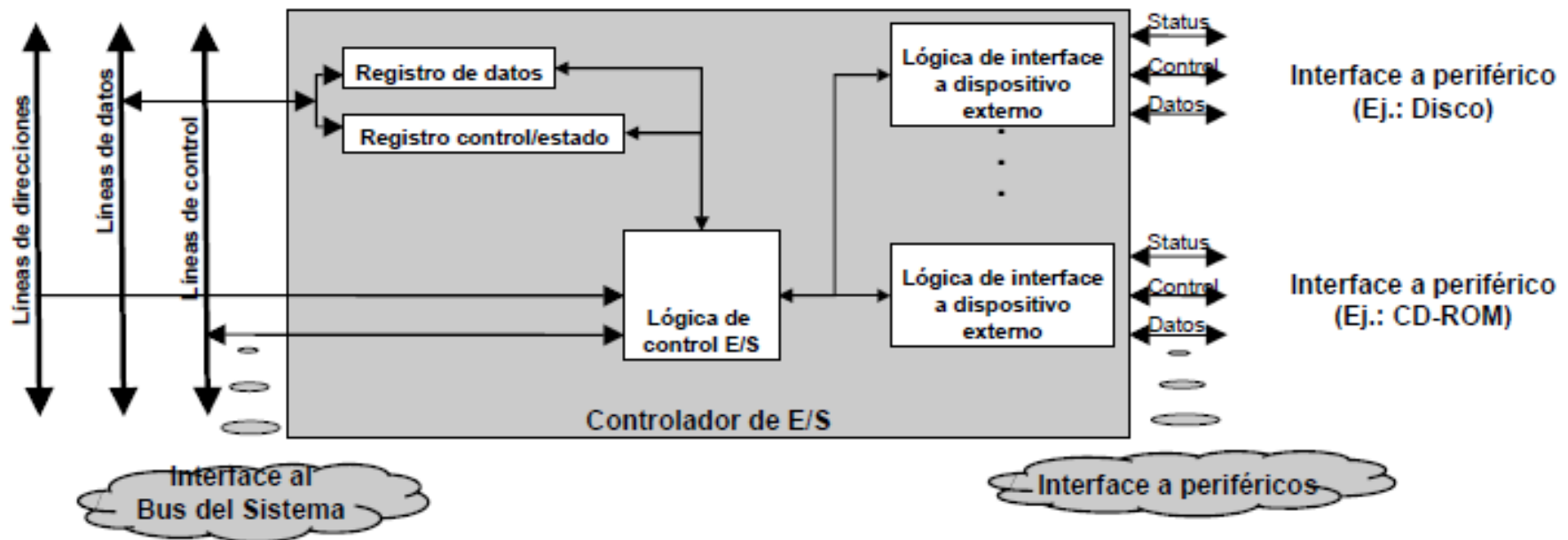
- La E/S tiene un impacto significativo tanto en la **Medida del rendimiento**
  - Difícil de medir y depende con frecuencia del entorno o aplicación
  - Medidas empleadas: Productividad (Operaciones E/S/Seg., Transacciones/Seg), Tiempo de respuesta

# Elementos Básicos del Sistema de E/S

- Interfaz o controlador del periférico
  - Sistema mixto hardware/software que permite la comunicación entre la CPU/memoria y el periférico
- Periférico
  - Dispositivo hardware (electrónico, mecánico u óptico) que posibilita la comunicación con el exterior.
  - Puede ser de almacenamiento o de E/S de datos.



# Estructura Típica de un Controlador de E/S



# Controlador de E/S

- Registros para la entrada y salida de datos
- Registro de control y estado
- Funciones del controlador:
  - Adaptación de longitud y formato de los datos.
  - Adaptación de señales eléctricas para la conexión de uno o varios periféricos
  - Adaptación de los tiempos de transferencia de CPU y periférico.



# Operación de E/S (generalizada)

- Comprobar si el dispositivo está listo (**leer registro de estado**)
- Enviar parámetros de la operación (**registro de control**)
- Transferir el dato (**registro de datos**)
- Terminación (**registro de control**)

# Dispositivos de e/s

Definiciones, características, técnicas para E/S

# Dispositivo de E/S

- Todos aquellos dispositivos conectados a un computador y diferentes a: CPU, Memoria
- El código para manejar la E/S de estos dispositivos es amplio y complejo.
- Resuelve los problemas de:
  - Sincronización
  - Detección de interrupciones.
  - Llamadas al sistema.

# Categorías de los Dispositivos

- **Bloques.** Se pueden direccionar, se puede escribir o leer cualquier bloque del dispositivo realizando primero una operación de posicionamiento sobre el dispositivo.
  - SSD's, memoria, CDs, unidades de cinta entre otros
- **Caracteres.** Aquellos que trabajan con secuencias de bytes sin importar su longitud ni ninguna agrupación en especial. No son direccionables.
  - Teclado, la pantalla, tarjetas de red, las impresoras entre otros

# Tipos de Dispositivos

- **Legibles para el humano.** Usados en la comunicación con el usuario
  - Impresores, video, mouse, teclado.
- **Legibles para la máquina.** Adecuado para equipos electrónicos
  - SSD, HDD, cintas, sensores.
- **Dispositivos de Comunicación.** Apropriados para la comunicación con equipos lejanos.
  - Módems, tarjetas de red, hub, switchs

# Características de los Dispositivos

- Comportamiento
  - Entrada (teclado, ratón)
  - Salida (impresora, pantalla)
  - Lectura y escritura o almacenamiento (discos, unidades de memoria flash, SSDs y cintas magnéticas)
- Frecuencia de datos
  - Frecuencia máxima a la que pueden ser transferidos datos entre procesador y periférico o entre memoria y periférico.

# Diferencias en dispositivos de I/O (I)

- **Velocidad de datos** Existen diferencias de varios ordenes de la magnitud en la velocidad de transferencia de datos.
- **Aplicación**
  - El disco usado para grabar archivos requiere un software de administración de archivos
  - El HD/SSD usado por el área de swap (VM) necesita HW y SW especiales para soportarlo.
  - La terminal usada por el administrador puede tener alta prioridad.
- **Complejidad del control** El driver para la impresora puede ser más simple que el driver del HD

# Diferencias en dispositivos de I/O (II)

- **Unidad de Transferencia**

- La información se puede transmitir en unidades de bytes, caracteres o bloques.

- **Representación de datos**

- Diferentes dispositivos emplean diferentes esquemas de codificación de datos, en los códigos de caracteres y en los convenios de paridad.

- **Condiciones de error.**

- Sea en la manera de informar al SO, en sus consecuencias y en el rango disponibles de posibles respuesta.



# Algunos Dispositivos según sus atributos

Dispositivo	Objeto	Interacción	Velocidad de Transmisión (Kb/sg)
Teclado	Entrada	Humano	0.01
Ratón	Entrada	Humano	0.01
Micrófono	Entrada	Humano	0.02
Escáner	Entrada	Humano	200
Altavoces	Salida	Humano	0.6
Impresora de línea	Salida	Humano	1
Impresora láser	Salida	Humano	100
Pantalla grafica	Salida	Humano	30000
CPU a buffer	Salida	Humano	200
Terminal de red	Entrada/Salida	Máquina	0.05
Adaptador de LAN	Entrada/Salida	Máquina	200
Disco óptico	Almacenamiento	Máquina	500
Cinta magnética	Almacenamiento	Máquina	2000
Disco magnético	Almacenamiento	Máquina	2000

# Técnicas para E/S (I)

- **E/S Programada.**

- El procesador emite la orden de E/S. El procesador espera a que termine la operación antes de continuar.

- **E/S dirigida por Interrupciones.**

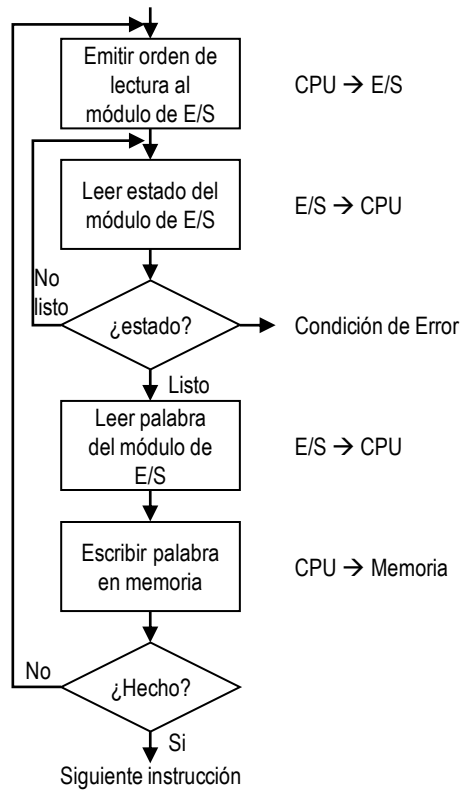
- La instrucción de E/S es emitida.
- El procesador continúa ejecutando instrucciones.
- El módulo de E/S envía un interrupción cuando está preparada para transmitir.

- **Acceso directo a la memoria (el DMA).**

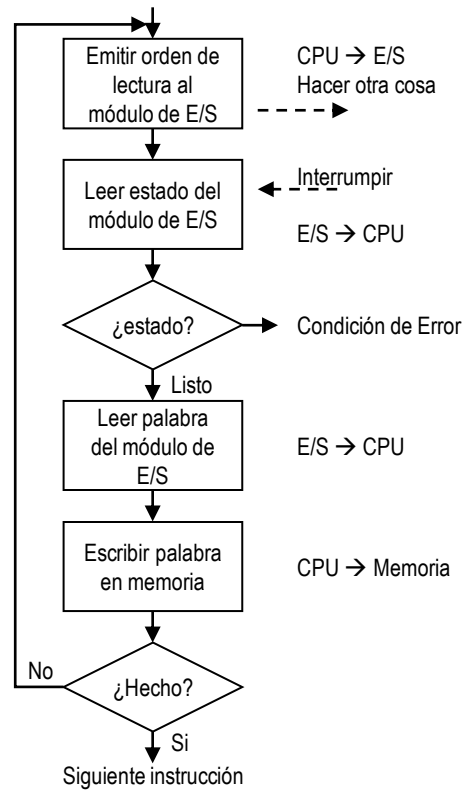
- Existe un módulo DMA que controla el intercambio de datos entre la memoria principal y el DES.
- El procesador es interrumpido sólo luego que el bloque entero ha sido transferido

# Técnicas para E/S (II)

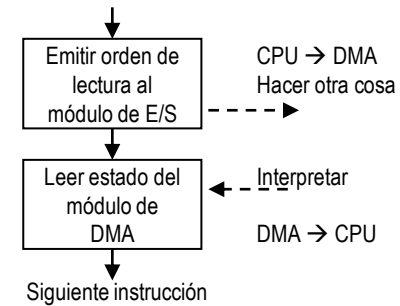
## E/S Programada



## E/S Dirigida por interrupciones



## DMA



# Evolución de la función de I/O

1. El  $\mu P$  **controla directamente** el dispositivo de E/S.
2. Controlador o **módulo de E/S**. El  $\mu P$  usa la E/S programada sin interrupciones. El  $\mu P$  no necesita conocer el dispositivo de E/S
3. Controlador o **módulo de E/S** con **interrupciones**.
  - El  $\mu P$  no gasta el tiempo esperando el termino de una operación de E/S.
4. Acceso directo a la memoria
  - Los bloques de los datos se mueven en memoria sin que participe el  $\mu P$ , este solo participa al inicio y al final.
5. Módulo de E/S con un procesador separado.
6. El módulo de I/O tiene su propia memoria local. Es una computadora.

# PROBLEMAS DE DISEÑO DE SISTEMAS OPERATIVOS

*Eficiencia, generalidad, organización de E/S*

# Eficiencia

- El dispositivo de E/S puede convertirse en un **cuello de botella**.
- La mayor parte son **lentos** comparados con RAM o el  $\mu$ P.
- La **multiprogramación** considera que ciertos procesos están esperando la E/S mientras otros se ejecutan.
- La E/S no puede **mantener la velocidad** del  $\mu$ P.
- Se puede **usar el área de swap** para introducir más procesos listos para ejecución y por lo tanto tener al  $\mu$ P ocupado, pero esta es una operación de E/S.

# Generalidad

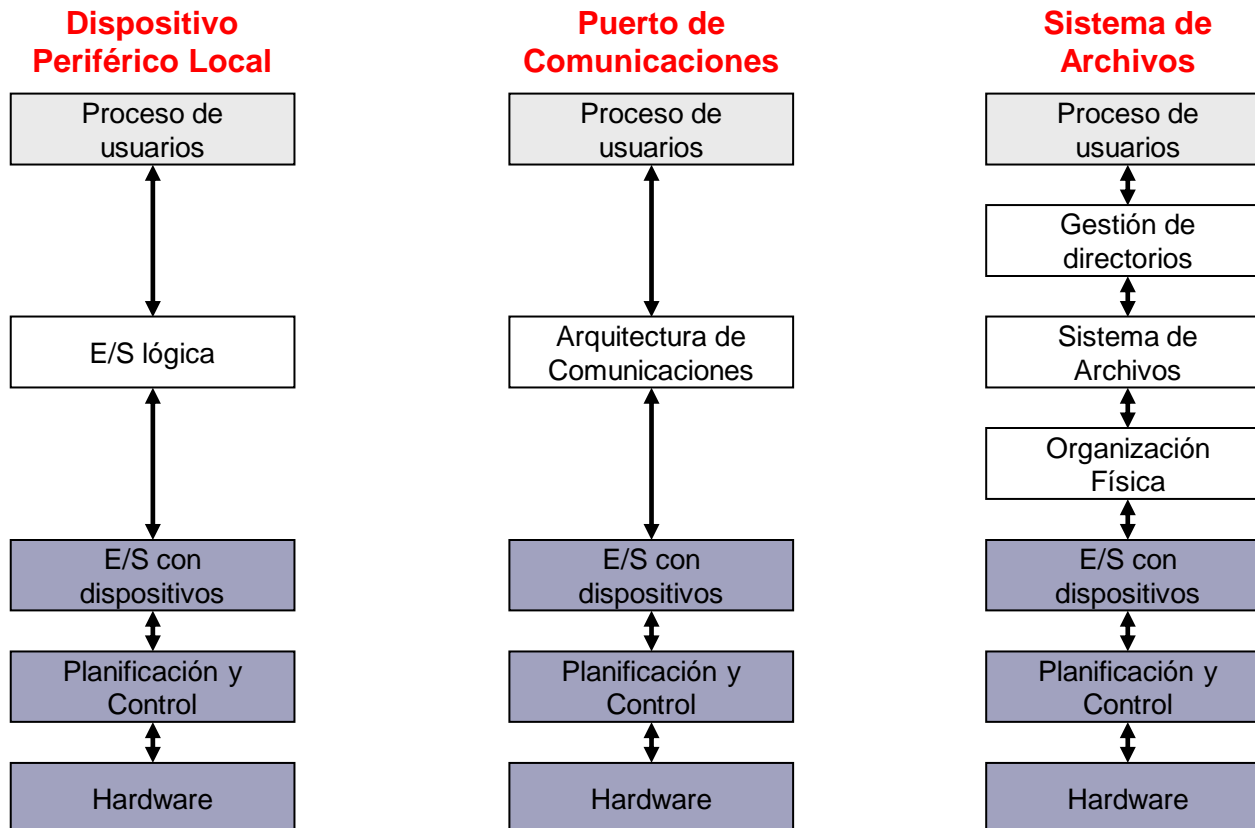
- En interés de la simplicidad y la extensión de errores:
  - Es deseable manejar todos los dispositivos de E/S de una manera uniforme.
    - Procesos → Dispositivos de E/S
    - SO → Dispositivos de E/S y operaciones.
  - En la práctica es muy difícil conseguir la generalidad, dado la diversidad de características de los dispositivos de E/S
  - Se puede usar enfoques jerárquicos y modulares.
  - Los niveles superiores pueden ver los dispositivos de E/S en términos generales tal como:
    - de lectura, escritura, apertura, cierre
    - bloqueo, desbloqueo

# Estructura lógica de las funciones de E/S

- El SO es de naturaleza jerárquica
- Las funciones se separan en función a:
  - Complejidad
  - Rangos de tiempo.
  - Nivel de abstracción.
- Cada nivel ofrece sus servicios al nivel superior.
  - Los niveles inferiores se enfrentan a rangos de tiempos menores, dado que tratan directamente con el dispositivo de E/S



# Organización de la E/S

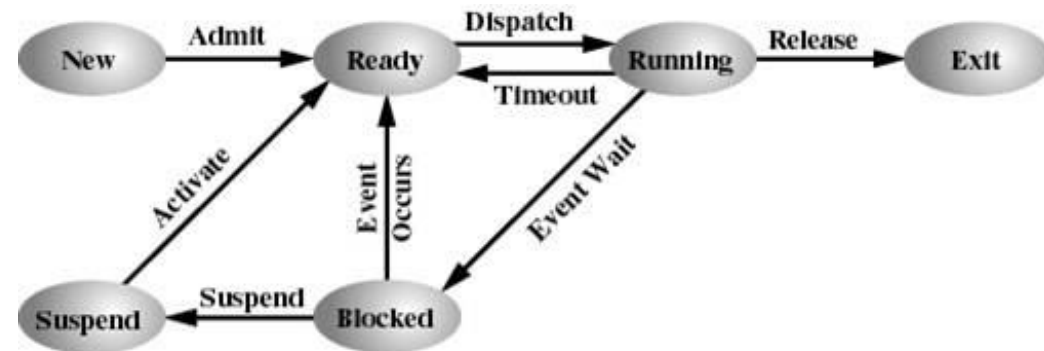


# BUFFERING de E/S

Almacenamiento intermedio de E/S

# Razones para almacenamiento temporal

- Los procesos deben esperar hasta que se complete la E/S.
- Las páginas deben permanecer en RAM durante E/S, sino se perderían. Sólo algunas partes del proceso pueden ser enviadas al disco.
- Riesgo de interbloqueo:
  - Proceso emite la orden de E/S
  - Se suspende temporalmente el proceso.
  - Se le expulsa antes de terminar la operación.
  - Se le bloquea esperando que la operación termine.
  - Mientras tanto la operación de E/S se bloquea esperando a que el proceso vuelva a memoria

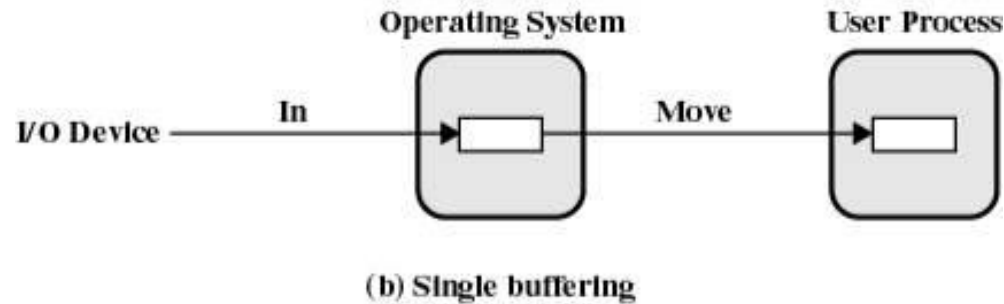
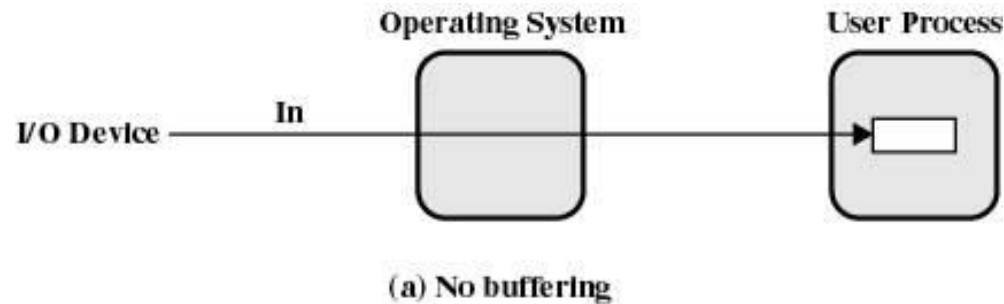


# I/O Buffering

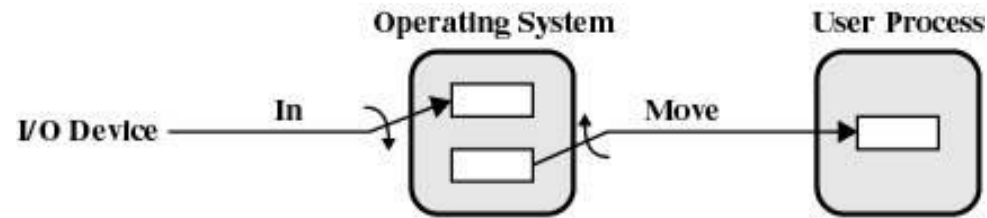
- Es conveniente hacer la transferencia de datos por adelantado a la petición y realizar la transferencia un tiempo después.

# Buffer Sencillo

- El sistema operativo asigna un buffer en la memoria principal para solicitudes de E/S



# Buffer doble y circular



(c) Double buffering



(d) Circular buffering

# Entrada y salida en disco

*Definición, estructura y clasificación*

# Discos

- Características

- Almacenamiento no volátil
- Altas capacidades de almacenamiento, barato y relativa lentitud.
- Nivel más bajo de la jerarquía de memoria

- Tipos

- Flexibles (legados)
- Duros



By Evan-Amos - Own work, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=27940250>



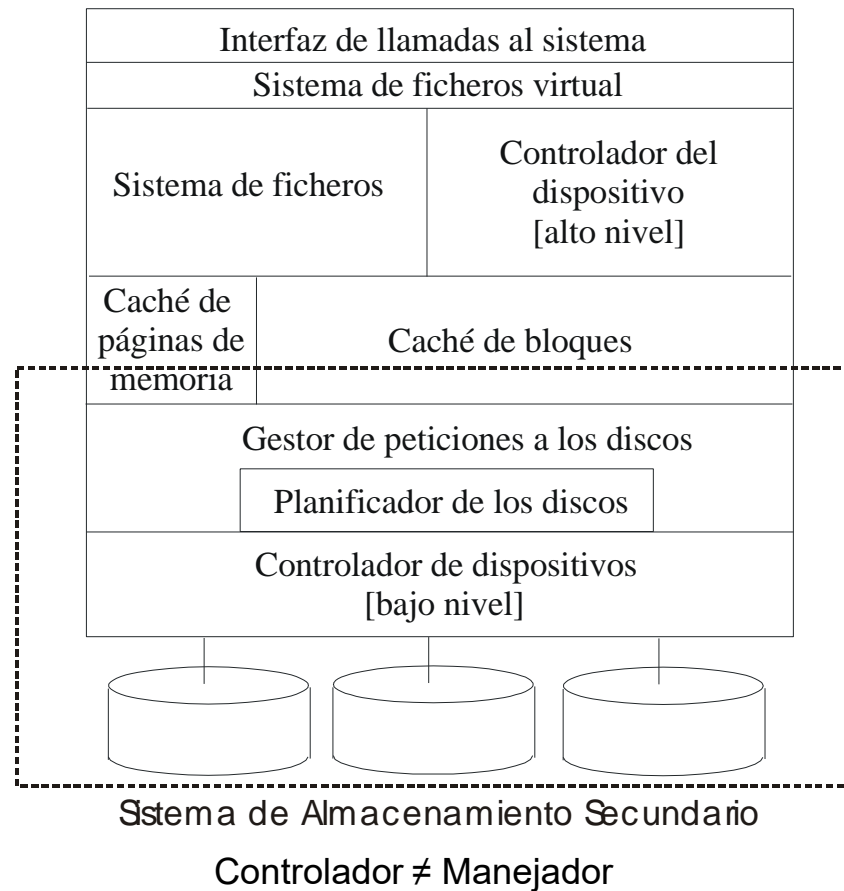
# Estructura de los Discos (I)

- 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.
- En la jerarquía de E/S se colocan justo debajo de la memoria RAM.

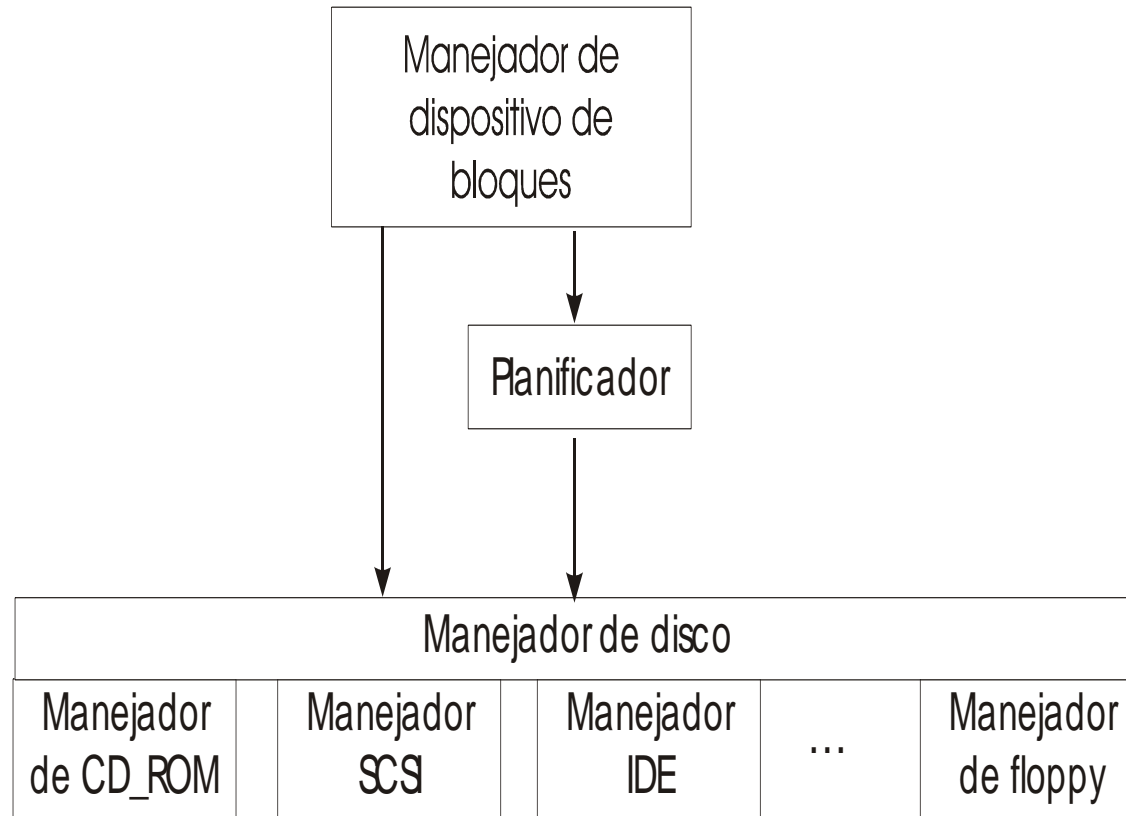
# Estructura de los Discos (II)

- Hay dos elementos 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



# Estructura de un manejador de disco



# Manejadores de disco

## 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.

# Discos Magnéticos (I)

- **Plato:** cada uno de los discos que hay dentro del *disco duro*.
- **Cara:** cada uno de los dos lados de un *plato*.
- **Cabeza:** número de cabezales.
- **Pistas:** una circunferencia dentro de una *cara*; la *pista 0* está en el borde exterior.
- **Cilindro:** conjunto de varias *pistas*; son todas las circunferencias que están alineadas verticalmente (una de cada *cara*).

# Organización del disco duro (I)

- Sector

- Cada una de las divisiones de una pista.
- El tamaño del sector no es fijo, se ubica entre 512 bytes y 4 KiB.
- Antiguamente el número de sectores por pista era fijo, lo cual desaprovechaba el espacio significativamente, ya que en las pistas exteriores pueden almacenarse más sectores que en las interiores.

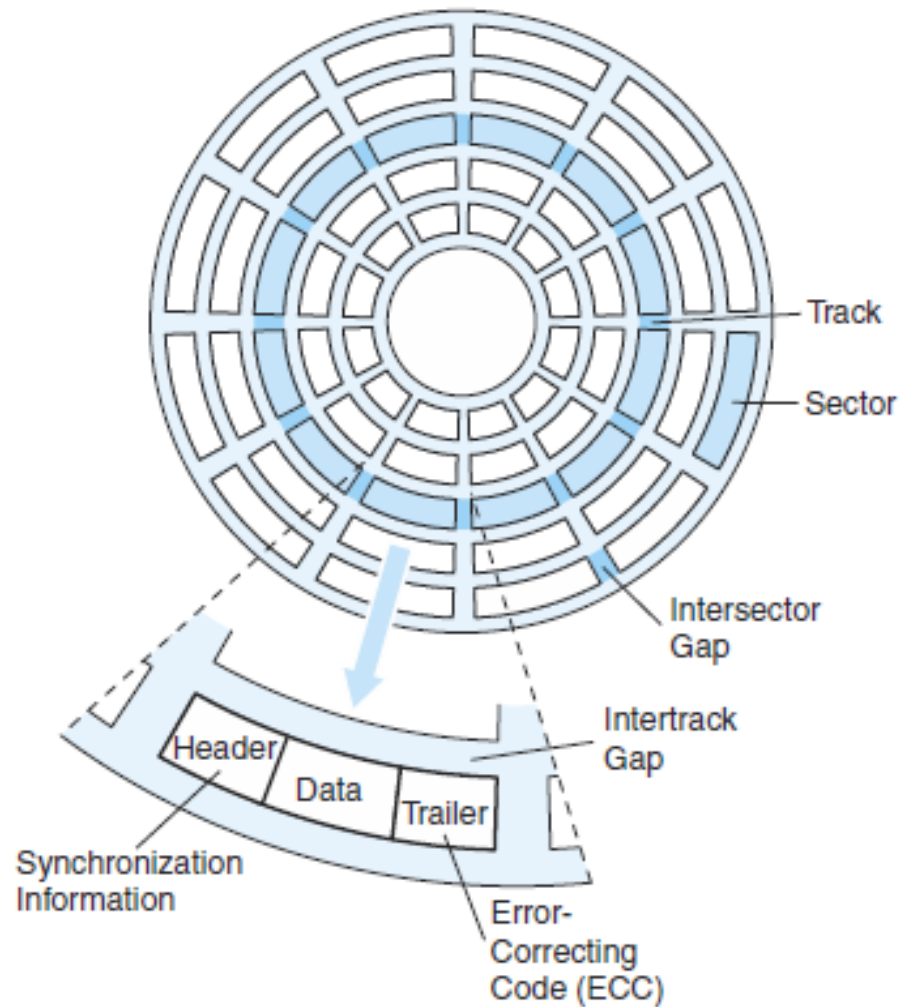
# Organización del disco duro (II)

- **Direcccionamiento**

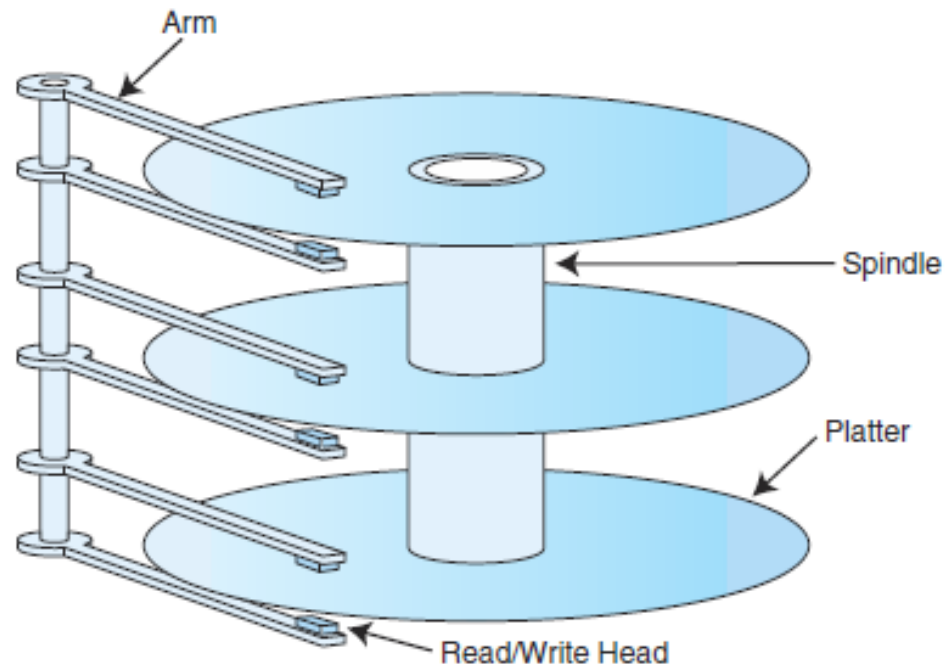
- El primer sistema de direccionamiento fue el **CHS** (cilindro-cabeza-sector), ya que con estos tres valores se puede situar un dato cualquiera del disco.
- Más adelante se creó otro más sencillo, **LBA** (direccionamiento lógico de bloques), que consiste en dividir el disco entero en sectores y asignar a cada uno un único número. Éste es el que actualmente se usa.



# Organización del disco duro (III)



# Organización del disco duro (IV)



# Discos Magnéticos - Acceso a datos

- Posicionar el cabezal sobre la pista (tiempo de posicionado o seek)
- Esperar que el disco rote y el sector deseado llegue a la cabeza de lectura/escritura (latencia rotacional)
- Tiempo para transferir un bloque (sector) de bits (tiempo de transferencia)

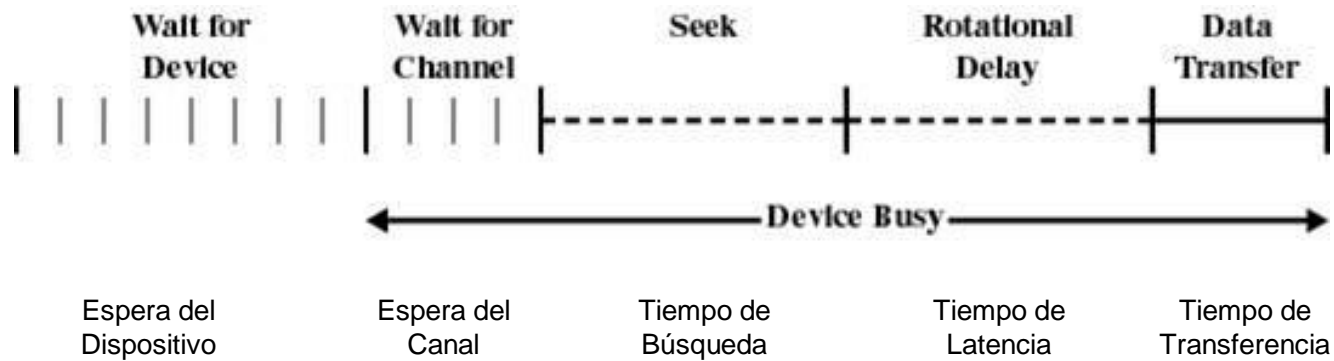
# Características de los Discos (I)

- Tiempo de posicionado
  - Tiempo para posicionar la cabeza sobre la pista deseada
- Latencia rotacional
  - Tiempo necesario para que el sector deseado pase por debajo de la cabeza una vez está posicionada en la pista correcta (latencia media: tiempo para girar 0.5 vuelta)
- Tiempo de transferencia
  - Tiempo para la transferencia de un bloque de bits (un sector).
- Tiempo de controlador
  - Tiempo que añade el controlador del dispositivo para realizar la E/S.
- Tiempo de espera, si existen otros procesos accediendo al disco

# Características de los Discos (II)

- Valores típicos
  - **Latencia rotacional**
    - Velocidad de giro : 3600 - 7200 rpm
    - 16 - 8 ms por revolución o vuelta
    - Latencia rotacional media (0.5 vuelta): 8 ms (3600 rpm) - 4ms (7200 rpm).
  - **Tiempo de transferencia depende de:**
    - Tamaño del sector, Velocidad de giro, Densidad de almacenamiento, Diámetro de los discos
    - Valores típicos de velocidad de transferencia: 2 - 15 MB/seg
    - Usando cache de disco se aumenta la velocidad de transferencia hasta los 40 MB/sg o más.

# Tiempos de transferencia



# Tiempo de Acceso al Disco (I)

- Ejemplo
  - Tamaño del sector: 512 bytes
  - Velocidad: 5400 rpm
  - Tiempo de posicionado: 12 ms
  - Velocidad de transferencia: 4 MB/seg
  - Tiempo de controlador: 1 ms

# Tiempo de Acceso al Disco (II)

- $TACC = T. \text{ posicionado} + \text{latencia rotacional} + T. \text{ de transferencia} + T. \text{ Controlador}$

$$TACC = 12 \text{ ms} + ((0.5 * (60/5400)) * 1000) + 512 / (4 * 1024 * 1024) + 1$$

$$TACC = 12 \text{ ms} + 5.5 \text{ ms} + 0.1 \text{ ms} + 1 \text{ ms}$$

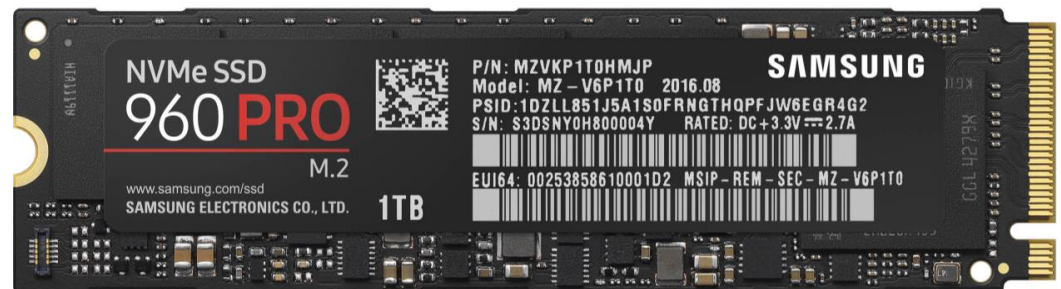
$$TACC = 18.6 \text{ ms}$$

- EXTENSIÓN: Normalmente las pistas exteriores se graban con densidad menor.
- Densidad de bits constante  $\rightarrow$  se graban más sectores en las pistas exteriores.
- Si número de bits/pulgada es cte  $\rightarrow$  implica leer y escribir a velocidad variable  $\rightarrow$  necesaria interfaz inteligente.



# Solid-State Storage Device (I)

- Unidad de Estado Sólido o SSD (Solid-State Storage Device) es un dispositivo de almacenamiento de datos que usa una memoria no volátil, NAND-based flash (creada por Fujio Masuoka en los 1980's).



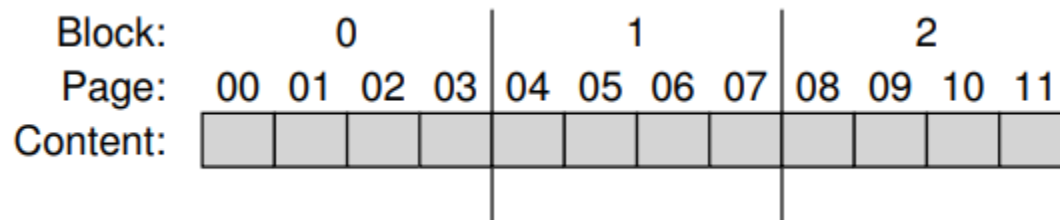
# Solid-State Storage Device (II)

- Se pueden encontrar en múltiples **formatos** o dimensiones físicas
  - Basadas en de 3.5"/2.5"/1.8" (al estilo de los HDD)
  - M.2 SSD
  - mSATA
  - PCIe
- Por otra parte, también se presentan distintas **interfaces** para comunicación y transmisión de datos
  - SATA
  - PCIe
  - NVMe



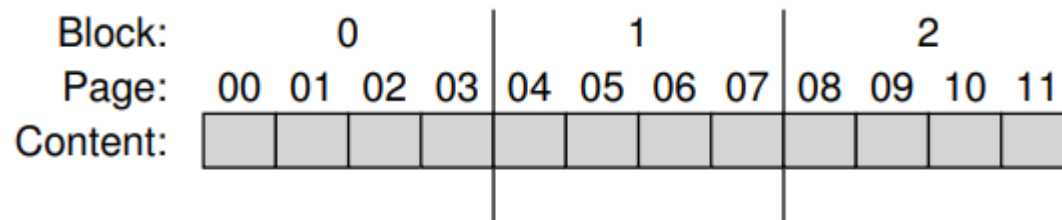
# Solid-State Storage Device (III)

- Las Flash poseen propiedades que las hacen únicas y con las cuales debemos lidiar a la hora de gestionar el acceso a las SSD basadas en estas memorias:
  - Las operaciones de escritura (una página, por ejemplo) conllevan el borrado y escritura de una porción significativamente mayor (por ejemplo, un bloque).
  - El desgaste ocasionado por las operaciones de escritura.



# Solid-State Storage Device (IV)

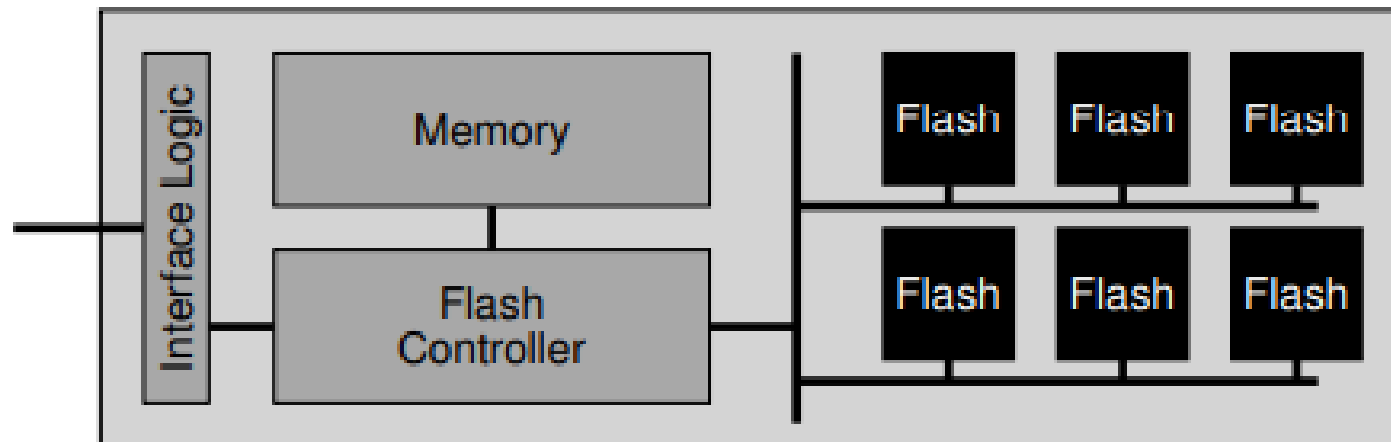
- Operaciones básicas
  - Lectura (de una página)
  - Borrado (un bloque)
  - Programación (de una página)



		iiii	<i>Initial: pages in block are invalid (i)</i>
Erase()	→	EEEE	<i>State of pages in block set to erased (E)</i>
Program(0)	→	VEEE	<i>Program page 0; state set to valid (V)</i>
Program(0)	→	<b>error</b>	<i>Cannot re-program page after programming</i>
Program(1)	→	VVEE	<i>Program page 1</i>
Erase()	→	EEEE	<i>Contents erased; all pages programmable</i>

# Flash chips (I)

- La interfaz de almacenamiento estándar se basa en el “bloque”. Los bloques (sectores) de tamaño 512 bytes (o más grandes) se pueden leer o escribir, dada una dirección de bloque.
- Internamente, un SSD consta de una cierta cantidad de chips flash (para almacenamiento persistente).
- También contiene cierta cantidad de memoria volátil (por ejemplo, SRAM); dicha memoria es útil para el almacenamiento en caché y almacenamiento en búfer de datos, así como para **tablas de mapeo**.



# Flash chips (II)

- La Capa de Traducción Flash (FTL) proporciona la funcionalidad para las operaciones de lectura y escritura.
- La FTL toma solicitudes de lectura y escritura en bloques lógicos (que comprenden el interfaz del dispositivo) y los convierte en lectura, borrado y programación (escritura) de bajo nivel.

# FTL por Mapeo directo (muy mal enfoque)

- El mapeo directo es la organización más simple de un FTL.
- Una lectura en la página lógica N se asigna directamente a una lectura de la página física N.
- Una escritura en la página lógica N es más complicado:
  - El FTL primero tiene que leer en todo el bloque que contiene la página N;
  - Luego, tiene que borrar el bloque;
  - Finalmente, el FTL programa (escribe) tanto las páginas antiguas como la nueva.

# FTL estructurado por registros (I)

- Debido a los inconvenientes del mapeo directo, la mayoría de los FTL actuales están estructurados por registros (Log-Structured FTL).
- Tras una escritura en el bloque lógico N, el dispositivo agrega la escritura al siguiente lugar libre en el bloque en el que se está escribiendo actualmente.
- Para permitir lecturas posteriores del bloque N, el dispositivo mantiene una tabla de mapeo (en su memoria y persistente, de alguna forma, en el dispositivo); esta tabla almacena la dirección física de cada bloque lógico en el sistema.



# FTL estructurado por registros (II)

Asumamos las siguientes operaciones

- Write(100) with contents a1
  - Write(101) with contents a2
  - Write(2000) with contents b1
  - Write(2001) with contents b2
- Bajo el supuesto de que todos los bloques del SSD son no válidos (i = INVALID) , deben borrarse antes para después programarse (escritura).

Block:	0				1				2			
Page:	00	01	02	03	04	05	06	07	08	09	10	11
Content:												
State:	i	i	i	i	i	i	i	i	i	i	i	i

# FTL estructurado por registros (III)

- Write(100) with contents a1
  - Write(101) with contents a2
  - Write(2000) with contents b1
  - Write(2001) with contents b2
- Cuando se recibe la primera escritura (bloque lógico 100), el FTL decide escribir el Bloque 0, que contiene las páginas 0, 1, 2 y 3
- Primero, debe emitirse un comando de borrado al bloque 0 para después guardar el dato.

Block:	0				1				2			
Page:	00	01	02	03	04	05	06	07	08	09	10	11
Content:												
State:	E	E	E	E	i	i	i	i	i	i	i	i

# FTL estructurado por registros (IV)

- Write(100) with contents a1
  - Write(101) with contents a2
  - Write(2000) with contents b1
  - Write(2001) with contents b2
- El bloque 0 está listo para programarse. Por lo general los SSD's escriben sus páginas en orden. En este caso se escribe el bloque lógico 100 en la página física 0.

Block:	0				1				2			
Page:	00	01	02	03	04	05	06	07	08	09	10	11
Content:	a1											
State:	V	E	E	E	i	i	i	i	i	i	i	i

Table:	100 → 0												Memory
Block:	0				1				2				Flash Chip
Page:	00	01	02	03	04	05	06	07	08	09	10	11	
Content:	a1												
State:	V	E	E	E	i	i	i	i	i	i	i	i	

# FTL estructurado por registros (V)

- Write(100) with contents a1
  - Write(101) with contents a2
  - Write(2000) with contents b1
  - Write(2001) with contents b2
- Para las siguientes escrituras, la SSD encuentra una ubicación que usualmente corresponde al próximo espacio libre en la página.
- La tabla de mapeo facilita la asociación lógica-física del bloque lógico y la página física.

Table:	100	→	0	101	→	1	2000	→	2	2001	→	3	Memory
Block:	0				1				2				Flash Chip
Page:	00	01	02	03	04	05	06	07	08	09	10	11	
Content:	a1	a2	b1	b2									
State:	V	V	V	V	i	i	i	i	i	i	i	i	

# FTL estructurado por registros (VI)

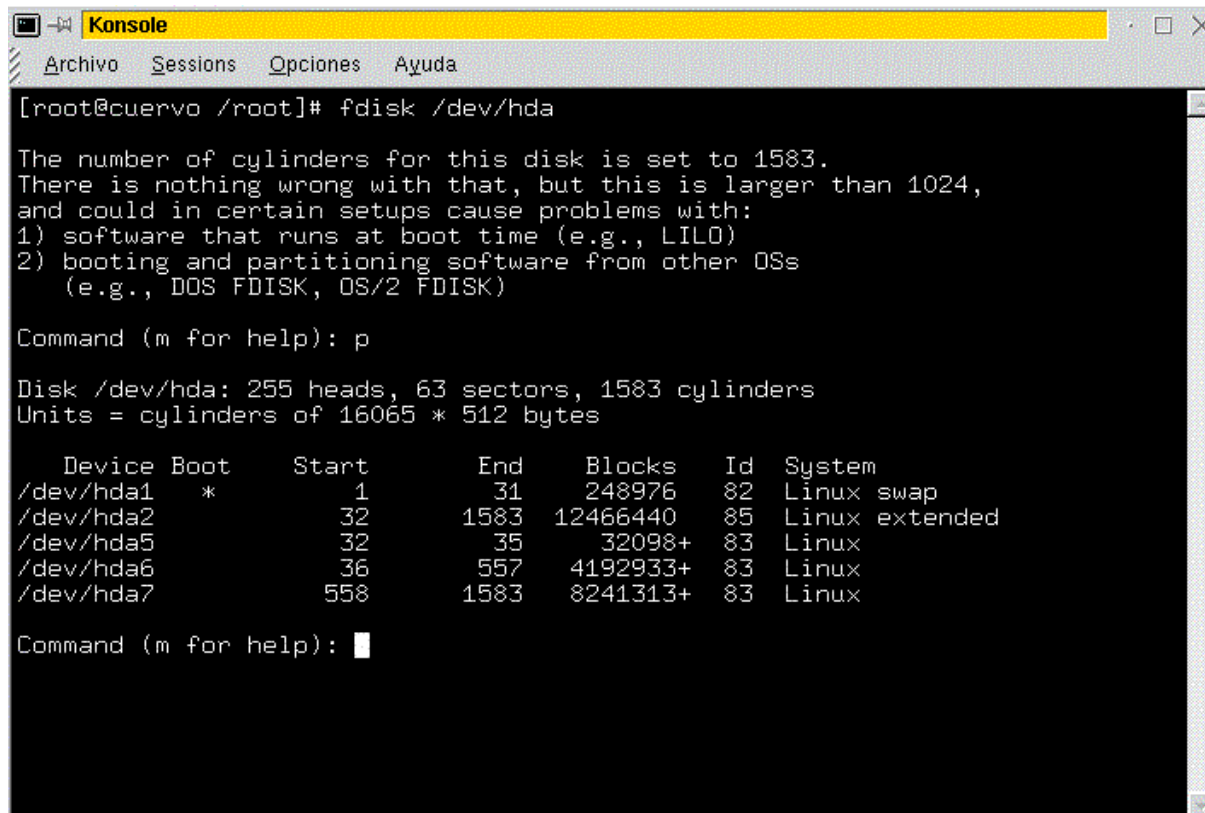
- Write(100) with contents a1
  - Write(101) with contents a2
  - Write(2000) with contents b1
  - Write(2001) with contents b2
- El enfoque basado en registros mejora el rendimiento.
- Los borrados solo se requieren de vez en cuando y se evita por completo la costosa lectura, modificación y escritura del enfoque por mapeo directo. A la vez que mejora en gran medida la confiabilidad.
- El FTL ahora puede distribuir escrituras en todas las páginas, realizando lo que se denomina **nivelación de desgaste** y aumentando la vida útil del dispositivo.

Table:	100	→	0	101	→	1	2000	→	2	2001	→	3	Memory
Block:	0				1				2				Flash Chip
Page:	00	01	02	03	04	05	06	07	08	09	10	11	
Content:	a1	a2	b1	b2									
State:	V	V	V	V	i	i	i	i	i	i	i	i	

# Estructura lógica de los Discos

- Los HDs se manejan como vectores grandes de *bloques lógicos*, siendo el bloque la unidad mínima de transferencia
- El vector de bloques lógicos se proyecta sobre los sectores del disco secuencialmente:
  - Sector 0: primer sector de la primera pista del cilindro más externo
  - El mapa se hace primero en esa pista, luego en las restantes pistas de ese cilindro y luego en los restantes cilindros
- El manejador de disco no sabe nada de la organización de los ficheros, sólo de particiones y bloques.
- En el sector 0 se guarda la *tabla de particiones*.
- Particiones activas o de sistema: permiten arrancar desde un sistema operativo.

# 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           55       4192933+    83   Linux
/dev/hda7           558        1583     8241313+    83   Linux

Command (m for help): █
```

# ALGORTIMOS DE PLANIFICACION DE DISCO

*Planificación del disco y algoritmos de planificación*



# Planificación de disco

- El SO es responsable de usar el hardware de forma eficiente.
- El tiempo de acceso tiene dos componentes principales:
  - *Búsqueda*: tiempo que tarda el brazo del disco para mover las cabezas hasta el cilindro que contiene el sector deseado
  - *Latencia*: tiempo de espera adicional para que el disco gire hasta ponerse sobre el sector deseado
- Objetivo: **minimizar el tiempo de búsqueda**, que es directamente proporcional a la distancia de búsqueda
- Ancho de banda: **bytes transferidos / tiempo de transferencia**

# Algoritmos de Planificación

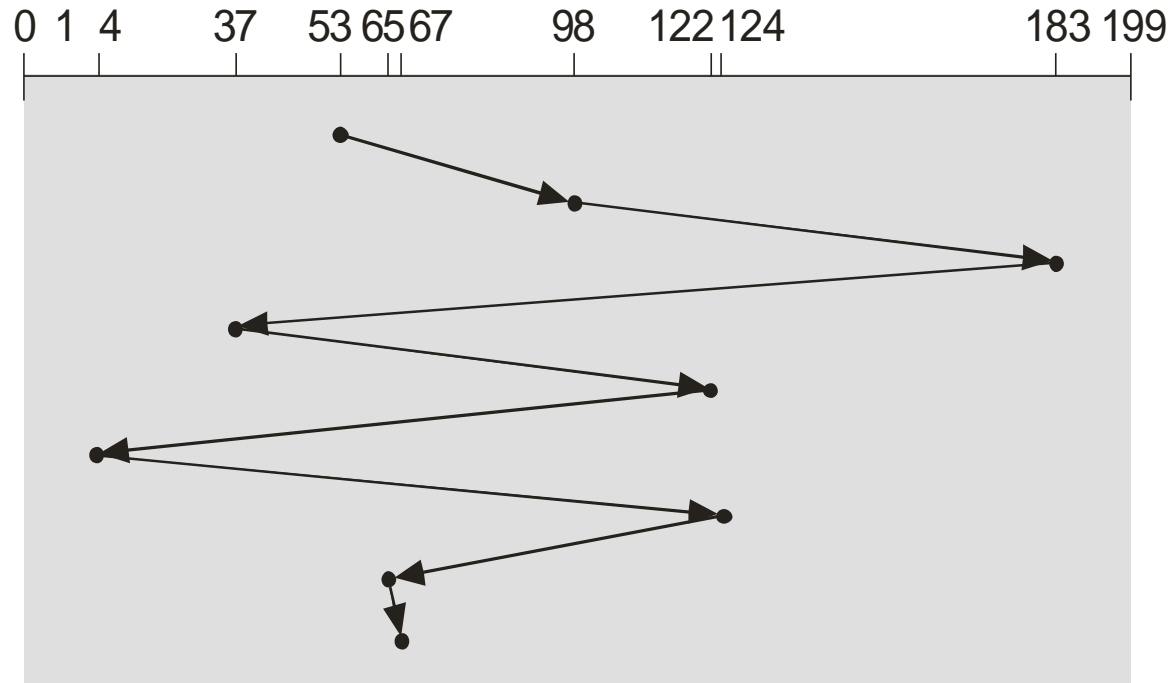
- FIFO, SSF, SCAN/CSCAN, otros
- Fundamentales para optimizar el acceso al disco.
- Criterios de planificación:
  - Optimizar el tiempo de búsqueda
  - Dar servicio determinista
- A continuación se estudian varios ejemplos usando la siguiente cola de peticiones:

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

Se asume que las cabezas están en 53

# FCFS (First Come, First Served)

queue = 98, 183, 37, 122, 14, 124, 65, 67  
head starts at 53



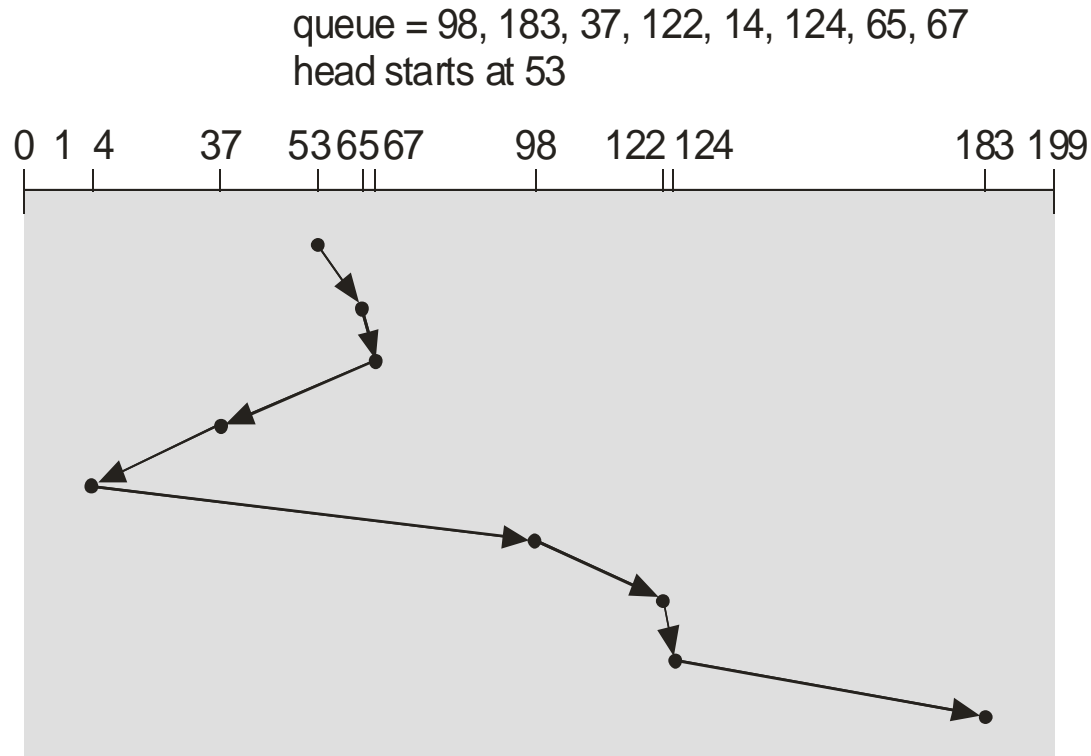
$$= (98-53) + (183-98) + (183-37) + (122-37) + (122-14) + (124-14) + (124-65) + (67-65)$$

$$= 640$$

# Shortest Seek Time First (SSTF) (I)

- Primero las peticiones que minimizan el movimiento de cabezas desde la posición actual
- La política SSTF es una variante de planificación de CPU con Shortest Job First (primero el más corto)
- Puede causar inanición de algunas peticiones
- Idea: maximizar el ancho de banda del disco

# Shortest Seek Time First (SSTF) (II)



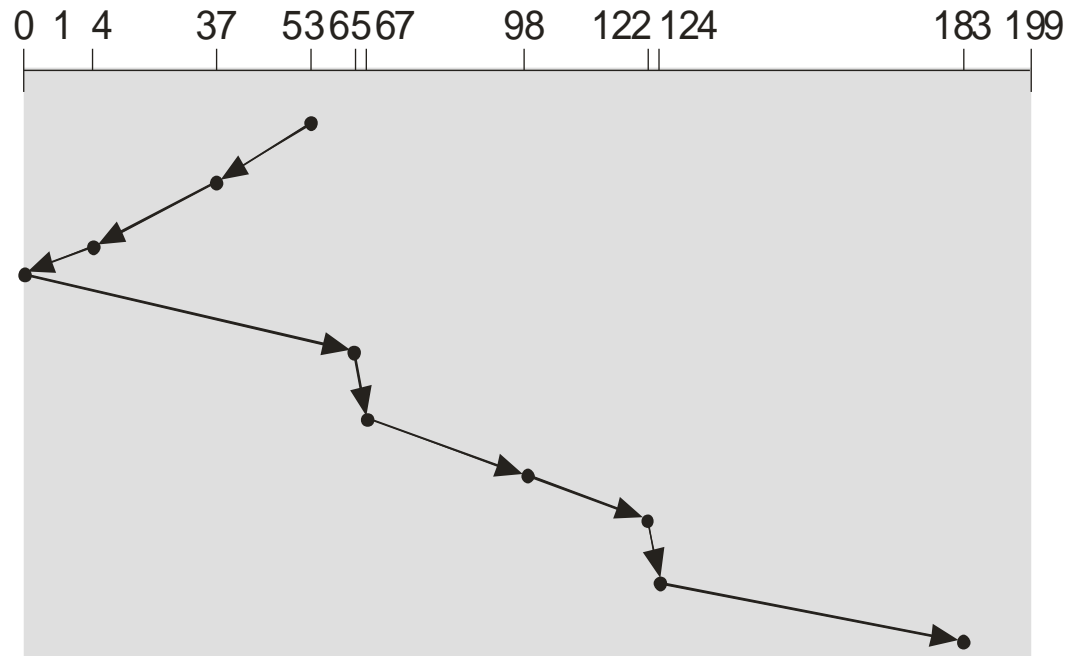
$$\begin{aligned}
 &= (65-53) + (67-65) + (67-37) + (37-14) + (98-14) + (122-98) + (124-122) + (183-124) \\
 &= 236
 \end{aligned}$$

# Ascensor (SCAN) (I)

- El movimiento del brazo empieza siempre en un extremo del disco y continúa hasta el otro. Allí se cambia el sentido y se vuelve el otro extremo
- Idea: evitar desplazamientos atrás y adelante
- Problema: puede retrasar mucho el servicio de algunas peticiones si no se insertan adecuadamente

# Ascensor (SCAN) (II)

queue = 98, 183, 37, 122, 14, 124, 65, 67  
head starts at 53



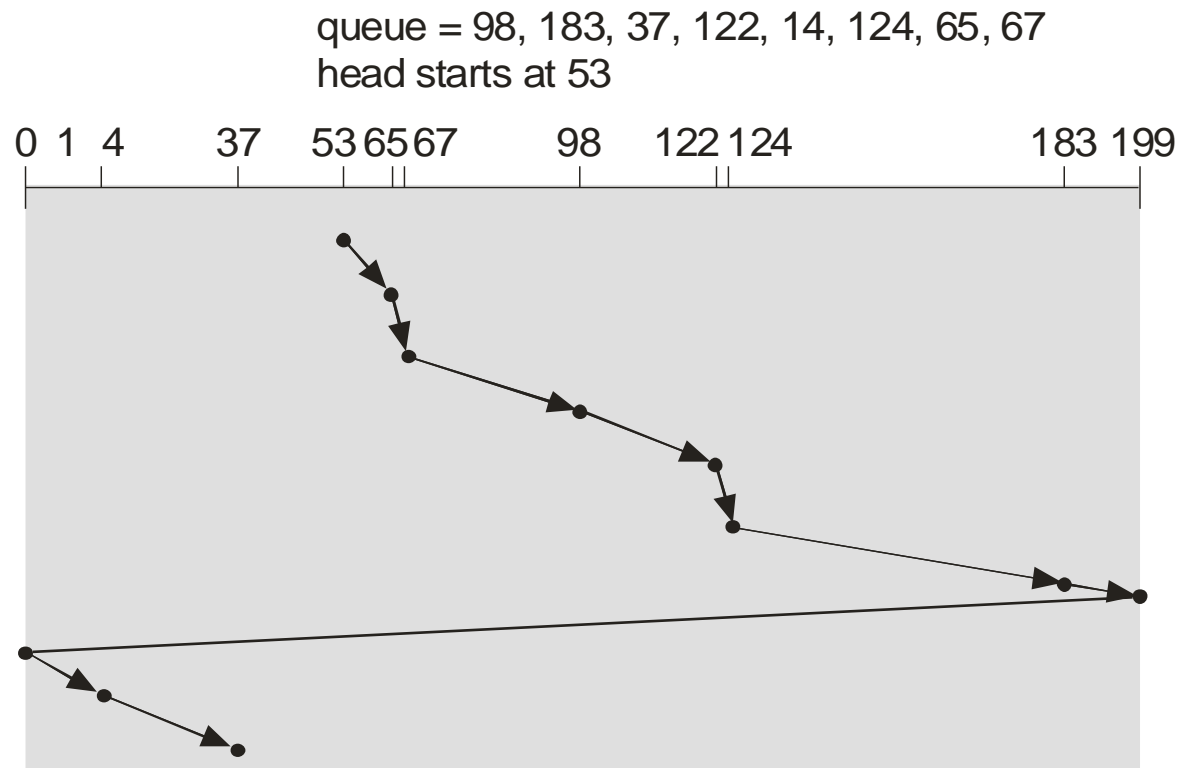
$$\begin{aligned}
 &= (53-37) + (37-14) + (14-0) + (65-0) + (67-65) + (98-67) + (122-98) + (124-122) + (183-124) \\
 &= 236
 \end{aligned}$$

# Ascensor cíclico (C-SCAN) (I)

- Variación del ascensor
- Las cabezas se mueven de un extremo a otro del disco y luego vuelven al principio
- No se atienden peticiones mientras las cabezas vuelven a la posición inicial
- Tiempo de espera más uniforme que el ascensor normal
- Trata los cilindros como una lista circular que enlaza el último cilindro con el primero
- El número es engañoso. Ir del último al primero se hace en un único movimiento y tarda muy poco
- Se usa frecuentemente



# Ascensor cíclico (C-SCAN) (II)



$$\begin{aligned}
 &= (65-53) + (67-65) + (98-67) + (122-98) + (124-122) + (183-124) + (199-183) + (199-0) + (14-0) + (37-14) \\
 &= 382
 \end{aligned}$$

# Selección de un Algoritmo de Planificación

- SSF es frecuente y parece el más natural
- SCAN y C-SCAN tienen 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.
- Algoritmo estándar en casi todos los sistemas: CSCAN

# BIBLIOGRAFIA

William Stallings. Operating Systems: Internals and Design Principles, 9th Edition.  
Pearson, 2018, ISBN 10: 1-292-21429-5

Tanenbaum, A. S., Bos, H. (2014). Modern Operating Systems. Boston, MA: Pearson.  
ISBN: 978-0-13-359162-0

Operating Systems: Three Easy Pieces Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau Arpaci-Dusseau Books August, 2018 (Version 1.00) <http://www.ostep.org>

Muchas gracias