

Conceptos de Sistemas Operativos  
Practica 6  
Ramiro Cabral

# 1 Administracion de E/S

## 1.1 Metas del SO

El SO debe garantizar el manejo de los dispositivos de E/S de manera uniforme, estandarizada. Tambien se deben ocultar la mayoria de los detalles del dispositivo en las rutinas de niveles mas bajos, en terminos de operaciones comunes como "read", "write", etc.

- **Eficiencia:** Los dispositivos de I/O pueden resultar muy lentos respecto a la memoria y a la CPU. El uso de la multi-programacion permite que un proceso espere por la finalizacion de su I/O mientras que otro proceso se ejecuta.
- **Planificacion:** Organizacion de los requerimientos a los dispositivos.

## 1.2 Clasificacion de los dispositivos

- **Orientados a bloques:** transmiten datos en bloques (paquetes) y por esa razon son usados a menudo para la transmision paralela de datos utilizando el buffer de datos del S.O. Por ejemplo: disco magnetico.
- **Orientados a flujos:** utilizan transmision en secuencia de datos sin usar buffer, transmitiendo un bit o un byte a la vez. Aceptan o entregan un flujo de datos sin considerar estructuras de bloques. No son direccionables, por lo que no permiten operaciones de busqueda.

Para realizar una correcta administracion de los dispositivos de I/O, el SO debe considerar sus diferencias:

- Heterogeneidad de dispositivos.
- Caracteristicas de los dispositivos.
- Velocidad.
- Nuevos tipos de dispositivos.
- Diferentes tipos de dispositivos.

### 1.3 Tecnicas de E/S

Existen distintas tecnicas de E/S, en las que tanto el software (SO) y el hardware trabajan en conjunto:

- **E/S programada:** El procesador envia un mandato de E/S, a peticion de un proceso, a un modulo de E/S. A continuacion, el proceso realiza una espera activa hasta que se complete la operacion antes de continuar.
- **E/S dirigida por interrupciones:** La CPU emite un mandato de E/S a peticion de un proceso y continua ejecutando las instrucciones siguientes, siendo interrumpido por el modulo de E/S cuando este ha completado su trabajo.
- **DMA (Direct Memory Access):** Un modulo de DMA controla el intercambio de datos entre la memoria principal y un modulo de E/S. El procesador manda una peticion de transferencia de un bloque de datos al modulo de DMA, y resulta interrumpido solo cuando se haya transferido el bloque completo.

A su vez, las distintas tecnicas de E/S pueden trabajar de dos formas:

- **Memory mapped:** Los dispositivos y memoria comparten el espacio de direcciones, I/O es equivalente a escribir/leer en la memoria, por lo que no hay instrucciones especiales para la memoria.
- **E/S aislada:** Hay un espacio separado de direcciones para I/O. Se necesitan lineas de I/O. Se requieren de instrucciones especiales de I/O (conjunto limitado).

### 1.4 Drivers

Consisten en una interfaz entre el SO y el Hardware cargadas como modulos en el espacio de memoria del Kernel, conteniendo el codigo dependiente del dispositivo para manejar el mismo y traducir requerimientos abstractos en los comandos para el dispositivo manejado.

## 2 HDDs

### 2.1 Organizacion Fisica de un HDD

- **Plato:** Es una de las superficies del disco. Cada plato tiene dos caras.

- **Pista:** Es una circunferencia en la superficie del disco. Son concéntricas y cada una contiene a varios sectores.
- **Cilindro:** Se forma al alinear verticalmente todas las pistas en los diferentes discos del disco duro que se encuentran en la misma posición radial.
- **Sector:** Unidad de almacenamiento más pequeña de un disco duro. Cada sector contiene una cantidad fija de bytes (generalmente, 512 bytes). como unidad de asignación de espacio en el disco. La cantidad de sectores en un cluster puede variar.
- **Clusters:** Los clusters son grupo de sectores contiguos que se utilizan
- **Cabezal:** Es el elemento que lee y escribe los datos en el disco. Hay un cabezal por cada superficie del disco.

La capacidad de un disco está dada por:

- $W$ : Cantidad de caras.
- $X$ : Cantidad de pistas.
- $Y$ : Cantidad de sectores por pista.
- $Z$ : Tamaño de cada sector.

$$\text{Capacidad} = W \cdot X \cdot Y \cdot Z \quad (1)$$

## 2.2 Tiempo de acceso a un HDD

El tiempo de acceso a un HDD está dado por:

- **Seek Time:** Tiempo que tarda el cabezal en posicionarse sobre la pista deseada.
- **Latency time:** Tiempo que pasa desde que el cabezal se posiciona en el cilindro hasta que el sector en cuestión pasa por debajo de la misma. Si no se conoce, se considera que es igual a lo que tarda el disco en dar media vuelta.
- **Transfer time:** Tiempo de transferencia del sector (bloque) del disco a memoria.

Para el **Almacenamiento secuencial** el tiempo de acceso esta dado por:

$$seek + latency + (tiempo\_transf\_bloque \cdot cantidad\_bloques) \quad (2)$$

Para el **Almacenamiento aleatorio**, el tiempo esta dado por:

$$(seek + latency + tiempo\_transf\_bloque) \cdot cantidad\_bloques \quad (3)$$

## 2.3 Algoritmos de planificacion en un HDD

El objetivo de los algoritmos de planificacion es minimizar el movimiento de la cabeza, ordenando logicamente los requerimientos pendientes al disco, considerando el numero de cilindro en cada requerimiento.

La atencion de requerimientos a pistas duplicadas se resuelven segun el algoritmo de planificacion:

- **FCFS**: First Come First Served, se atienden de manera separada (tantas veces como se requieran).
- **SSTF**: Shortest Seek Time First, selecciona el requerimiento que requiere el menor movimiento del cabezal.
- **SCAN**: Barre el disco en una direccion atendiendo los requerimientos pendientes en esa ruta hasta llegar a la ultima pista del disco y cambia la direccion. Cuando llega al final del disco, comienza a barrer en sentido contrario.
- **LOOK**: Similar al SCAN, pero no llega hasta la ultima pista del disco, sino que llega hasta el ultimo requerimiento de direccion actual.
- **C-SCAN**: Circular SCAN, similar al SCAN, pero restringe la atencion en un solo sentido. Al llegar a la ultima pista del disco en el sentido actual, vuelve a la pista del otro extremo (salto) y sigue barriendo en el mismo sentido.
- **C-LOOK**: Circular LOOK, se comporta de manera similar al LOOK, pero restringe la atencion en un solo sentido, al llegar a la ultima pista de los requerimientos en el sentido actual vuelve a la primer pista mas lejana del otro extremo y sigue barriendo en el mismo sentido.

### 2.3.1 Atencion de Page Faults

Los fallos de pagina poseen mayor prioridad que los requerimientos convencionales, por lo tanto deben ser atendidos inmediatamente despues del requerimiento que se esta atendiendo actualmente. Una vez que no existan mas requerimientos por page faults en la cola, se procede:

- **FCFS**: en orden FCFS.
- **SSTF**: en orden SSTF.
- **SCAN/LOOK**: con el sentido que determina la atencion de los ultimos dos requerimientos (puede cambiar de sentido).
- **C-SCAN/C-LOOK**: con el sentido original, el sentido no cambia.

## 3 Inodos

Los inodos son estructuras auxiliares que permiten en sistemas UNIX, referenciar a los archivos y acceder a ellos.

- Poseen informacion de bajo nivel de los archivos.
- Se identifican con un numero.
- Contienen metadata del archivo y punteros a los bloques de datos en el disco que conforman el archivo.