**Cuestionario SO**

1. *¿Qué es la gestión o administración de la memoria?*

La gestión de memoria consiste en la asignación de una memoria física de capacidad limitada a los diversos procesos que la soliciten con el objetivo de activarse y entrar en ejecución.

1. *Mencione y explique brevemente los cinco requisitos para la gestión de la memoria*

Según stallings menciona que existen cinco requisitos para la gestión de memoria:

• Reubicación: cuando se ejecute un programa, los programadores no sabrán en qué lugar de la memoria se encuentra. Mientras el programa esté en ejecución, esté podrá ser movido al disco y devolverlo a la memoria principal en un lugar diferente, a esto se le conoce como reubicación.

• Protección: el sistema operativo sólo puede combinar el nivel de referencia y nunca puede hacer una referencia al bloque de control de procesos. *Los procesos no deberían ser capaces de referenciar el espacio de memoria de otro proceso sin permiso*. Es imposible comprobar las direcciones absolutas de los programas puesto que estos pueden ser reubicados

• Compartición: en la compartición, el segmento de datos y el segmento de código comparten el mismo segmento de código, pero cada uno conserva su independencia trabajando individualmente

• Organización lógica: los programas son escritos en módulos, los módulos se pueden escribir y compilar por separado. A los módulos se les puede dar diferente grado de protección: de sólo lectura, de sólo ejecución

• Organización física: la memoria disponible para un programa y sus datos puede ser insuficiente. El solapamiento permite asignar la misma zona de memoria a diferentes módulos.

Los registros y el caché son administrados por el procesador.

Ram y memoria secundaria son administrados por el sistema operativo

1. *Defina particionamiento de memoria*

Para stallings, la operación principal de la gestión de la memoria es traer los procesos a la memoria principal para que el procesador los pueda ejecutar. En casi todos los sistemas multiprogramados modernos, esto implica el uso de un esquema sofisticado denominado memoria virtual.

Una técnica más sencilla y anterior a la memoria virtual es el particionamiento fijo, que se ha dejado

de utilizar y solo queda en SO obsoletos.

1. *¿Cómo trabaja genéricamente el sistema de administración de memoria de multiprogramación con particiones fijas y con particiones variables? Ventajas y desventajas entre ellas*

**Particionamiento fijo:** existe de igual y distinto tamaño, fue una técnica utilizada por los primeros mainframes de IBM, el SO ‘OS/MFT’ (Multiprogramado con un número fijo de tareas)

Ventajas:

-Simplicidad

-Resolución de direcciones de tiempo de carga.

-Puede limitarse simplemente con un espacio de direccionamiento acorde al compilador.

Desventajas:

-Rigidez

-Multiprocesamiento limitado

-Desperdicio de espacio

**Particionamiento dinámico:** un SO importante que utilizó esta técnica fue el SO de mainframes de IBM, el SO ‘OS/MVT’ (Multiprogramado con un número variable de tareas).

Ventajas: No existe la fragmentación interna.

Desventajas: Fragmentación externa, se debe compactar la memoria. El compactado toma tiempo

1. *Explique qué se entiende por fragmentación externa e interna.*

**Fragmentación externa:** (es externa porque lo que se fragmenta es el proceso en sí, y el disco o los espacios de memoria son ocupados en su totalidad. *Se llama así porque los procesos que entran en memoria no ocupan posiciones continuas*). Esto ocurre cuando el primer bloque libre de memoria no es suficiente para que el siguiente programa lo use, *ejemplo:*

un sistema carga tres programas en la memoria, cada uno ocupando 50 megabytes o MB. El segundo programa termina, dejando ese bloque de 50 MB libres. Si el siguiente programa a iniciar requiere de 100 MB, no sería capaz de utilizar ese bloque de 50 MB de espacio libre, y el sistema le asigna el siguiente intervalo de 100 MB libres. Esta brecha de memoria queda sin usarse hasta que algún programa requiera menos de 50MB. Con el tiempo estos espacios pueden sumarse

**Fragmentación interna:** (es interna por que es la misma memoria la que se encuentra fragmentada) se deriva del hecho de que mientras la memoria está denominada en bytes, la unidad más pequeña disponible generalmente es mayor que eso debido a las reglas de memoria de direccionamiento

1. *Paginación y segmentación. Definición y características principales de cada una de las técnicas.*

**Paginación simple:***Características:*

*●* El programa (proceso) no necesita estar contiguo en la memoria.

● Los procesos y la memoria física se dividen en particiones igual tamaño.

● Los procesos se dividen en páginas (pages).

● La memoria física se divide en marcos (frames).

● Elimina la fragmentación externa, pero sufre un poco menos de la interna.

*Definición:* La paginación permite que la memoria de un proceso no sea contigua, y que a un proceso se le asigne memoria física donde quiera que ésta esté disponible. (la paginación evita el gran problema de acomodar trozos de memoria de tamaño variable en el almacenamiento auxiliar)

**Segmentación Simple:**

*Características:*

● Un programa y sus datos asociados (imagen del proceso) se puede subdividir

en segmentos.

● Por lo tanto, el proceso (sus segmentos), no necesitan estar contiguos en memoria.

● Además, los segmentos tienen tamaños variables.

● Cada segmento de un proceso representa una parte desde el punto de vista del

programador (del compilador): código, pila, datos, montículo y bibliotecas.

● Elimina la fragmentación interna, pero sufre la fragmentación externa.*Definición:* La segmentación es un método por el cual se consigue aumentar el rendimiento de algunos sistemas electrónicos digitales. Es aplicado, sobre todo, en microprocesadores.

*Características extras de la segmentación simple:*

\*Los procesos se dividen en un conjunto de segmentos de longitud variable (aunque con una longitud máxima)

\*Los segmentos de un proceso no necesitan encontrarse contiguos en memoria

\*Los procesos suelen dividirse en tres segmentos:

~Segmento de código (el código máquina del proceso)

~Segmento de pila (La pila asociada al proceso)

~Segmento de datos (Los datos asociados al proceso)

*Ventajas de la segmentación simple:*

~La fragmentación interna no existe (como en la partición dinámica)

~La fragmentación externa será menor que en la partición dinámica, dado que tendremos el proceso divido en partes pequeñas

*Desventajas de la segmentación simple:*

~Resulta similar a la partición dinámica (similar empleo de segmentos)

~Es necesario cargar en memoria todos los segmentos de un programa en ejecución.

~El programador debe ser consciente de la limitación de tamaño máximo de los segmentos

En comparación con la partición dinámica, radica en que, con segmentación, un programa puede ocupar más de una partición y estas no tienen por qué ser contiguas

**Segmentación paginada:**

*Características:*

● Los segmentos de un proceso se paginan.

● Se combinan las ventajas de la segmentación y la paginación.

● Genera un poco de sobrecarga.

*Definición:* La segmentación paginada es un esquema en el que cada segmento se divide en páginas. Por tanto, la dirección lógica se compone por:

~El número de segmento.

~El número de página que hay dentro del segmento.

~El desplazamiento dentro de la página.

Combinando ambos modelos, la segmentación paginada ofrece lo mejor de la paginación y lo mejor de la segmentación:

No hay fragmentación externa.

~Soporte a estructuras de datos que cambian.

~Soporte a compartición y protección.

~Soporte a funcionalidades de hardware.

1. *Memoria virtual. Concepto. Ventajas y desventajas.***Definición de memoria virtual:** fue idea por Fotheringham, en 1961. Se conoce actualmente como **memoria virtual.** Según Tannenbaum (2005), la idea básica detrás de la memoria virtual es que cada programa tiene su propio espacio de direcciones (lógicas), el cual se divide en trozos llamados páginas

*Características:*

● Se parte sobre la base de la paginación simple y sus ventajas.

● Puede contener procesos más grandes que la memoria física/real.

● Mayor multiprogramación, más procesos en memoria física/real.

● Menor restricciones para los desarrolladores de software.

● El espacio de memoria se compone de la memoria principal y secundaria (disco).

*Ventajas:*

● Permite optimizar uso de memoria

● Mantiene más procesos en memoria principal

● Permite que un proceso sea más grande que toda la memoria principal.

● Elimina las restricciones en cuanto al desarrollo de aplicaciones (más fácil, más rápido, se justifica más el uso de aplicaciones con más lógica en memoria.)

*Desventajas:*

● Sobre carga por gestión compleja de memoria.

● Costo asociado a la transformación de direcciones.

● Memoria adicional que se requiere para almacenar las tablas que debe mantener el SO.

● Arma de doble filo, la ‘MV’ puede ser una herramienta muy poderosa para incrementar la performance del pc, pero se encuentra limitado a los dispositivos que contendrán la memoria virtual, ciertos aspectos como:

~ Velocidad del procesador

~ Tamaño de la memoria real / virtual

1. *¿Qué es una dirección lógica y una dirección física?*

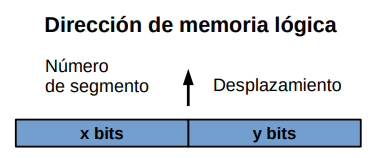
**Una dirección física:** es una dirección de memoria que está en una parte del disco.

**Una dirección lógica:** es aquella que se compone de un inicio de segmento + un desplazamiento, que juntos forman una dirección física de memoria.

Estas últimas tienen un tamaño:

Cantidad máxima de segmentos por procesos ->

Tamaño máximo de segmento ->



1. *Describa brevemente los formatos de archivos ejecutables Common Object File Format (COFF), Extensible Linkable Format (ELF) y Portable*

*Executable (PE)*

**Common Object File Format:** este formato, es una especificación de formato para archivos ejecutables, código objeto y bibliotecas compartidas, usadas en sistemas Unix.

**Extensible Format:** ELF (anteriormente llamado Extensible Linking Format), es un formato de archivo estándar común para archivos ejecutables, código objeto, bibliotecas compartidas y volcados al núcleo.

**Portable Executable:** es un formato para archivos ejecutables, de código objeto, bibliotecas de enlace dinámico (DLL en Windows), para archivos fuentes FON, 1 y otros usados en versiones de 32 y 64 bits (En Linux es ELF).

10)*En la gestión de memoria virtual, ¿Qué es y cuándo se produce lo que se denomina petición de página (fallo de página)?*

En un sistema de memoria virtual paginada, un fallo de página *es una excepción arrojada cuando un programa informático requiere una dirección que no se encuentra en la memoria principal actualmente*. Aunque el termino sugiere un mal funcionamiento, se trata de un procedimiento normal dentro de la marcha del programa.

11) *Defina el concepto de hiperpaginación (thrashing)*

Para Silva (2015), el rendimiento degradado en un SO es motivado por el intercambio excesivo y, se lo conoce como **hiperpaginación** (thrashing, paliza)

Un proceso está hiper paginado si pasa más tiempo paginado que ejecutando las instrucciones del programa.

12) *Algoritmos de ubicación (Primer ajuste, Siguiente ajuste, Mejor ajuste y Peor ajuste). Explique brevemente cada uno de ellos:*

**Primer Ajuste:** este algoritmo asigna la primera partición disponible (estado = 0) y suficiente para la tarea que solicita, la búsqueda se hace a partir de la última partición asignada, la manera de asignar es FIFO.

**Siguiente Ajuste:** este algoritmo llevara frecuentemente a la asignación de bloques libre del final de la memoria. El resultado es que el bloque de memoria libre más grande, que suele aparecer al final del espacio de memoria, se divide rápidamente en fragmentos pequeños.

**Mejor Ajuste:** este algoritmo asigna la partición más pequeña disponible pero suficiente para la tarea que lo solicita, la búsqueda de bloques se realiza desde el principio de la tabla hasta encontrar la partición adecuada, previo a esto se debe realizar una reordenación de la tabla.

**Peor Ajuste:** este algoritmo ajusta la partición más grande disponible y suficiente para almacenar la tarea solicitada, la búsqueda se realiza en toda la tabla de memoria, por lo que se sugiere que la tabla se encuentre ordenada de bloque más grande a más chico.

13) *Algoritmos básicos (FIFO, LRU, Optimo y Reloj) de reemplazo de páginas. Explique brevemente cada uno de ellos.*

Algoritmos básicos: luego de un fallo de página, y con la memoria principal (RAM) llena, en el SO debe utilizar un algoritmo de reemplazo (FIFO,LRU,Optimo,Reloj).

**FIFO:**

~Es una política de reemplazo sencilla de implementar.

~ Reemplaza la página que lleva más tiempo en memoria principal.

~ Se supone que una página que lleva mucho tiempo en memoria principal puede haber dejado de usarse.

Lo anterior, no siempre se presenta como un razonamiento correcto.

**LRU:** (Es el menos utilizado, la idea del algoritmo LRU si no se ha accedido a un dato en el periodo reciente, se puede considerar que es poco probable que se acceda a él en el futuro. Por lo tanto, cuando el espacio esta lleno, los datos más antiguos no utilizados se eliminan primero). Características:

~ Es una política de reemplazo difícil (imposible) de implementar. Necesita conocer el futuro.

~ Reemplaza la página que tiene más alejada su siguiente referencia.

~ Es un estándar y se utiliza para contrastar con algoritmos reales.

~ Siempre obtiene los mejores resultados respecto de otros algoritmos.

**Optimo:** se aplica a problemas de optimización en los que partimos de una población finita y bien definida, la función objetivo converge hacia una solución óptima y los parámetros a evaluar son de tipo continuo. Es el tipo de algoritmo más empleado y fácil de programar

*Características:*

COMPLETAR

**Reloj:** lo que hace es similar a una lista circular, de forma que, al llegar al último elemento de la lista, pasa automáticamente al primero. Los elementos no se mueven al final de la cola cuando son accedidos, simplemente se cambia el bit de referencia a 1.

*Características:*

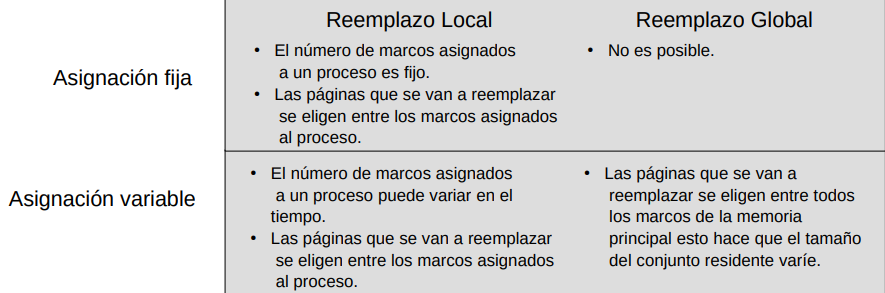
~ Es una política de reemplazo bastante sencilla de implementar. Muy utilizada

~Intenta igualar los resultados de la política LRU.

~Incluye un bit adicional (bit de usado) en cada uno de los marcos de páginas.

~ Reemplazará la página que se encuentre en un marco con el bit usado en cero. Se recorren los marcos de páginas de forma circular para encontrar la página candidata. Si en el recorrido se pasa por un marco con el bit de usado en 1 se modifica a 0 y se pasa al siguiente marco de página.

1. **Politicas de recuperación: indica en qué momento se trae una página a memoria principal**. Existen dos maneras que son paginación bajo demanda y paginación adelantada.
   * **Bajo demanda:** una página se lleva a memoria solo cuando se hace referencia a una posición de dicha página.
   * **Adelantada (prepagin):** se llevan a memoria además otras páginas, diferentes de la que ha causado el fallo de página.
2. **Politicas de ubicación:** la política de ubicación indica en qué parte de la memoria física van a residir las páginas o segmentos de la memoria de un proceso.
   * **Con sistemas de segmentación pura:** se utilizan políticas del estilo mejor ajuste, primer ajuste, peor ajuste o siguiente ajuste.
   * **Con sistemas de paginación pura o combinados con segmentación:** es irrelevante dado que cualquier combinación página-marco tiene la misma eficiencia.
3. **Políticas (algoritmos) de reemplazo:** la política de reemplazo, es el aspecto de gestión de la memoria más estudiado. Cuando todos los marcos de memoria principal están ocupados y es necesario traer una nueva página para resolver un fallo de página, la presente política determina qué página de las que actualmente están en memoria va a reemplazarse.
   * **Bloqueo de marcos:** algunos marcos de la memoria principal pueden encontrarse bloqueados. Gran parte del núcleo del SO se almacena en marcos que están bloqueados, esto implica una restricción para los algoritmos de reemplazo.
   * **Algoritmos básicos:** óptimo, usado menos recientemente (LRU), FIFO y reloj.
4. **Gestión del conjunto residente**

****

1. **Políticas de limpieza:** la política de limpieza es la opuesta a la política de recuperación, se encarga de determinar cuándo una página que está modificada se debe escribir en disco.
   * **Bajo demanda:** se escribe a memoria secundaria sólo cuando se ha seleccionado para un reemplazo
   * **Adelantada:** se escribe las páginas modificadas antes de que sus marcos de páginas se necesiten.

**Concepto de archivo:** se le llama **archivo** a un conjunto de información relacionada, definida por su creador y almacenada en una computadora.

**Concepto de sistema de archivos:** la forma que tiene una computadora de organizar, nombrar, almacenar y manipular a los archivos es lo que denominamos **sistema de archivos** (filesystem)

**¿Para qué sirven los directorios?** Sirven para organizar y proporcionar información acerca de la estructuración de los archivos en los sistemas de archivos, también para evitar ambigüedades (no puede haber en el mismo lugar dos archivos con el mismo nombre).

*#Nota: en GNU/Linux todo es un archivo, y su directorio es en forma de árbol, todo parte desde la rama principal “/”*

**Características de enlaces**

|  |  |  |
| --- | --- | --- |
|  | Duro | Simbólico |
| Si se cambia algo en un archivo, ¿afecta a los dos? | SI | SI |
| En caso de borrar el archivo original, ¿se puede acceder mediante el enlace? | SI | NO |
| ¿Se pueden crear enlaces a carpetas? | NO | SI |
| Tamaño | Ocupan menos | Ocupan más |
| Acceso al contenido | Es más rápido | Es más lento |
| Ubicación | Solo se pueden usar en la partición que lo crearon | Se pueden usar en cualquier ubicación, partición y sistema de archivos de nuestro disco |
| Permisos | El grupo y propietario del enlace son los mismos que el original | ----- |
| Cambios de Ubicación del archivo original | Si lo cambias (original) el enlace duro no se rompe | Si lo cambiamos, el enlace simbólico se romperá |

**Sistema de directorios en GNU/Linux:** es por nivel jerárquico, seguido de “/” se encuentran:

* **/bin:** binarios de comandos esenciales (chmod, ls, pwd, cp y rm)
* **/boot:** archivos estáticos del cargador de arranque
* **/dev:** archivos de dispositivo
* **/etc:** configuración específica del equipo (host, passwd, group, shadow, fstab)
* **/lib:** bibliotecas compartidas esenciales y módulos del núcleo
* **/media:** punto de montaje para medios extraíbles
* **/mnt:** punto de montaje para montar un sistema de archivos temporalmente.
* **/opt:** paquetes de software de aplicaciones complementarias
* **/sbin:** binarios esenciales del sistema (fdisk, fsck, mkfs, mkswap, swapon y swapoff)
* **/srv:** datos de los servicios proporcionados por este sistema.
* **/tmp:** archivos temporales
* **/usr:** jerarquía secundaria (bin, include, lib, local, sbin, share, src y local/src)
* **/var:** datos variables

**Rutas:**

~ Relativas => ../directorio/

~ Absolutas => /home/jero/desktop/

**Montaje de sistemas de archivos:**

~ Montar (mount)

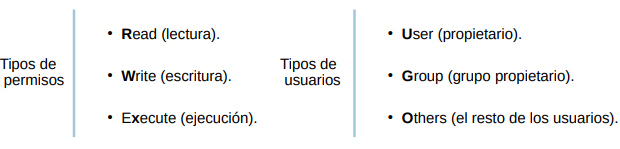
~ Desmontar (unmount)

Podes montar/desmontar un sistema de archivos como por ejemplo un pendrive, lo montas, lo usas y lo desmontas para sacarlo del usb.

**Operaciones sobre un directorio:**

* **Creación:** se crea un directorio, está vacío excepto por “.” y “..” que el sistema coloca de manera automática.
* **Borrado:** se puede eliminar solo un directorio vacío (“.” Y “..”)
* **Abrir y cerrar:** los directorios se pueden leer, listar sus archivos, etc. Cuando se leyó un directorio, se debe cerrar para liberar espacio en la tabla interna.
* **Renombrar:** (en general) los directorios son archivos y se les puede cambiar el nombre como a los archivos.
* **Vincular:** técnica que permite a un archivo aparecer en más de un directorio. Esta llamada al SO especifica un archivo existente y el nombre de una ruta, creando el vínculo correspondiente.

**Protección:** puede contar con acceso total (sin protección), **acceso restringido** (protección por propietario y grupo) o acceso controlado (listas de control de acceso – ACL).

**Tipos de permisos:** 

*Hablando de los archivos:*

R **->** leer un archivo**.**

**W** -> escribir un archivo.

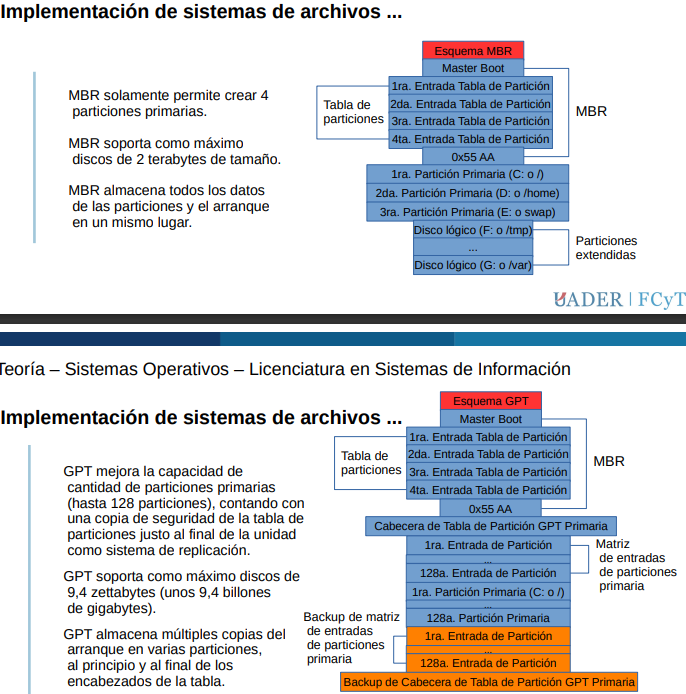
**X** -> ejecutar un archivo.

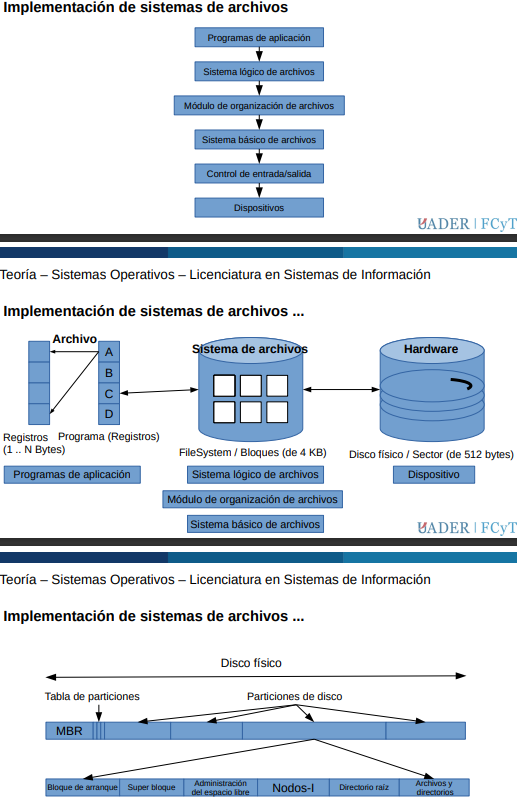
*Hablando de directorios:*

**R** -> listar archivos del directorio.

**W** -> crear, renombrar y borrar archivos.

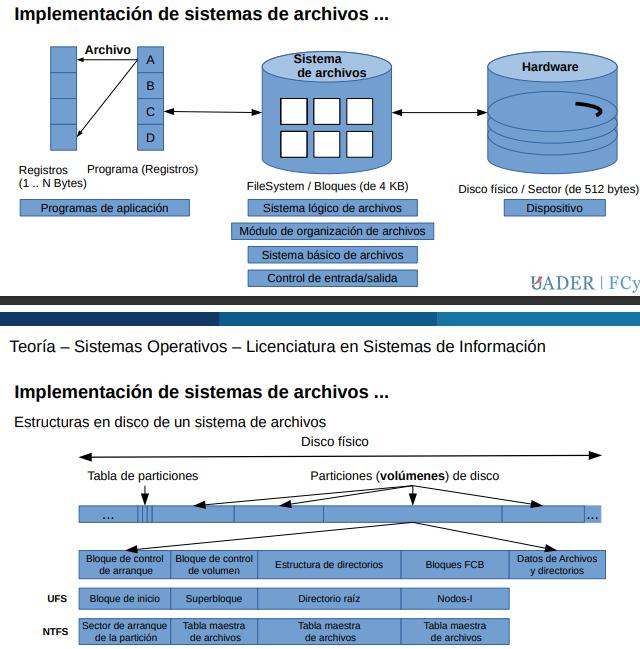
**X** -> leer, escribir y ejecutar los archivos.





**Implementación de sistemas de archivos:** puede ser en dispositivos físicos de almacenamiento secundario. Por ejemplo, discos IDE, discos SATA, unidades de discos externas USB y unidades de discos SCSI.

* **El nivel mas bajo** es *control E/S****,*** está compuesto por controladores de dispositivos y rutinas de tratamiento de interrupción, que se encargan de transferir la información entre la memoria principal y disco.
* **El sistema básico de archivos** sólo necesita enviar comandos genéricos al controlador de dispositivos apropiado, con el fin de leer y escribir bloques físicos en el disco. Cada bloque físico se identifica mediante su dirección de disco numérica (ej: unidad 1, cilindro 73, pista 2, sector 10).
* **El módulo de organización de archivos** tiene *conocimiento* acerca de los archivos y de sus bloques lógicos, así como de sus bloques físicos. Conociendo el tipo de asignación de archivos utilizado y la ubicación del archivo, el módulo de organización de archivos puede traducir las direcciones lógicas de bloque a direcciones físicas de bloque, que serán utilizadas por el sistema básico de archivos. También gestiona el espacio libre.
* **El sistema lógico de archivos:** gestiona la info de metadatos. *Los metadatos* incluyen toda la estructura del sistema de archivos, excepto los propios datos (el contenido del archivo). *El sistema gestiona la estructura de directorio* para proporcionar al módulo de organización de archivos la información que éste necesita, a partir de un nombre de archivo simbólico. (mantiene la estructura mediante bloques de control de archivo).
* **Bloque de control de archivo:** contiene info acerca del archivo, incluyendo su propietario, permisos y ubicación del contenido.



**Estructuras en memoria de un sistema de archivos:**

* Tabla de montaje de cada volumen en el sistema.
* Caché de la estructura de directorio accedida recientemente.
* Tabla global de archivos abiertos en el sistema
* Tabla de archivos abiertos de cada proceso

**Métodos de asignación:**

* **Asignación contigua:** los bloques de memoria están uno al lado del otro, poseen un inicio y una longitud
* **Asignación enlazada:** se enlazan los bloques de memoria para un mismo archivo posee punteros con la referencia a el próximo bloque de memoria (inicio y fin, no necesitan estar ordenados)
* **Asignación indexada:** poseen un bloque de inicio y la referencia a el próximo bloque

**Gestión del espacio libre en disco:**

* **Vector de bits:** La lista de espacio libre se implementa como un mapa de bits o vector de bits. Cada *bloque está representado por 1 bit*. Si el bloque está libre, el bit será igual a 1; si el bloque está asignado, el bit será 0.
* **Lista enlazada:** consiste en enlazar todos los bloques de disco libres, manteniendo un puntero al primer bloque libre en ubicación especial del disco y almacenándolo en la memoria caché. El primer bloque tendrá un puntero al siguiente bloque libre del disco, así sucesivamente

**Recuperación**: Los archivos y directorios se mantienen tanto en memoria principal como en disco, y debe tenerse cuidado para que los fallos del sistema no provoquen una pérdida de datos o una incoherencia en los mismos

* **Comprobación de coherencia:** compara los datos de la estructura de directorios con los bloques de datos del disco y trata de corregir todas las incoherencias que detecte
* **Copia de seguridad y restauración:** Los discos magnéticos fallan en ocasiones y es necesario tener cuidado para garantizar que los datos perdidos debido a esos fallos no se pierdan para siempre. Deben utilizarse programas del sistema para realizar copias de seguridad de los datos del disco a otro dispositivo. Luego se podrá restaurar el sistema con algunas de las copias de seguridad

**Sistemas de archivos con estructura de registro:** se utilizan algoritmos de recuperación basados en registro que se utilizan en las bases de datos. Estos algoritmos de registro se han aplicado con éxito al problema de la comprobación de coherencia. Las implementaciones resultantes se conocen con el nombre de sistemas de archivos orientados a transacciones y basados en registro (o en diario)

* **File allocation table (FAT):** lanzado en 1977 por Marc McDonald (Microsoft), posee las versiones FAT12/FAT16/FAT32 y ExFAT, actualmente se utiliza FAT32 en usb.
  + **Limites:**
    - 6.3(archivos binarios)/9 caracteres(archivos ASCII) de longitud máxima de nombre de archivo.
    - No se permiten 0x00 y 0xFF como primer carácter en las entradas de directorio.
    - No tiene limite definido ya que no posee una jerarquía de directorio.
    - Se desconoce el tamaño máximo de archivo, el tamaño máximo por volumen y el número máximo de cantidad de archivos.
  + **Metadatos:**
    - *No implementa premisos de archivos POSIX.* No aplica seguridad.
    - Utiliza y almacena **fecha y hora** de **creación** de los archivos, también así de la **ultima lectura o acceso** al archivo
    - No usa la **fecha y hora del último cambio de los metadatos**
    - No implementa control de acceso ACL
  + **Características:**
    - No soporta enlaces duros ni simbólicos.
    - No usa journaling para los bloques ni para los metadatos
    - No es sensitivo a MAYUS y minúsculas.
* **UFS y ext…**
  + **Límites:**
    - **255 bytes** de longitud máxima de nombre de archivo
    - Entrada de directorio: **cualquier byte excepto NULL**
    - No posee limite definido para longitud máxima de pathname.
    - **2GB** es el tamaño máximo de archivo y por volumen
    - Se desconoce el número máximo de cantidad de archivos
  + **Metadatos:**
    - **Implementa permisos de archivos POSIX.**
    - No usa **fecha y hora de creación** de los archivos
    - Usa y almacena la **fecha y hora del último acceso** o lectura de archivo, así también el **último cambio** de los metadatos
    - No implementa lista de control de acceso (ACL)
  + **Caracteristicas:**
    - Soporta enlaces duros y simbólicos
    - No usa journaling para bloques ni metadatos.
    - Es sensitivo a MAYUS y minúsculas
  + **Inodos:** ¿Qué es un inodo? es una estructura de datos que almacena información sobre un archivo de nuestro sistema de archivos. Un inodo no tiene nombre y se identifica mediante un número entero único. El nombre al cual apunta el inodo es guardado en una tabla que guarda el kernel. Cada inodo únicamente puede contener datos de un solo archivo del sistema de archivos.
    - El inodo es el FCB (file control block) del sistema de archivos
    - Existe un inodo por archivo
    - También contiene los atributos del archivo
    - Posee punteros que permiten acceder a los datos
    - Tiene un tamaño fijo de 128 bytes.}

**Dispositivos E/S**

*Categorías*

* Legibles para el **usuario** (impresoras, pantallas, teclados y ratones)
* Legibles para la **máquina** (unidades de disco, unidades de cintas, sensores y controladores)
* De **comunicación** (controladores de líneas digitales, módems y placas de red)

*Diferencias entre las categorías de dispositivos E/S:*

Velocidad de transferencia, aplicación, complejidad de control, unidad de transferencia, representación de datos, condiciones de error.

Velocidades de transferencia entre dispositivos E/S legibles para el usuario:

El más rápido es el monitor (10^9), seguido de la impresora láser (10^6), después el ratón (10^2) y por último el teclado (10^1)

Velocidades de transferencia entre dispositivos E/S legibles para la máquina:

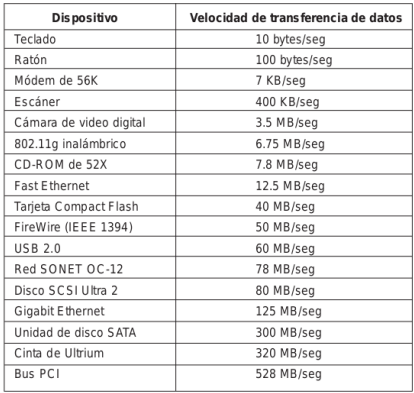
Primero esta el disco rígido (10^7/8) y por último el disco flexible (10^5)

Velocidades de transferencia entre dispositivos E/S de comunicación:

Primero esta la placa de red Gigabit ethernet (10^9) y por último el modem (10^4)

E/S por software: SO, Módulo de E/S del SO, los drivers de teclado, ratón, disco, etc.

E/S por hardware: Los dispositivos y sus controladores (teclado con controlador de teclado, ratón con controlador de ratón, etc).



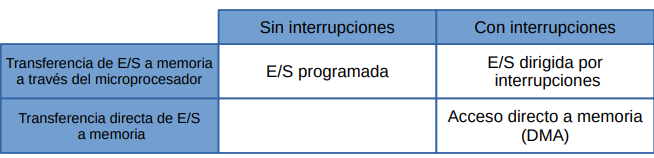
**Organización del sistema E/S:**

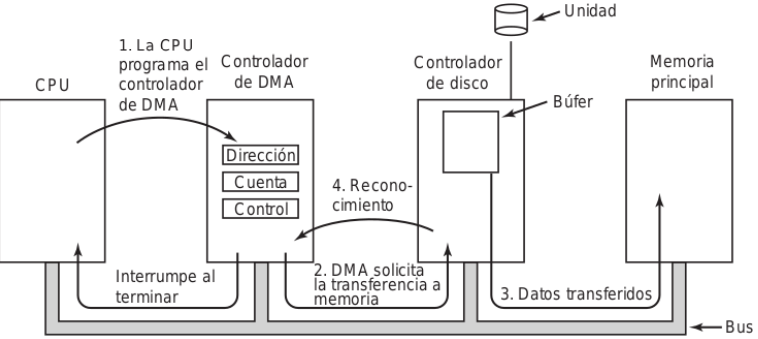
**~E/S Programada:** el procesador envía un mandato de E/S, a petición de un proceso, a un módulo de E/S. A continuación, ese proceso realiza una espera activa hasta que se complete la operación antes de continuar.

**~E/S por interrupciones:** el proce emite un mandato E/S a petición de un proceso y sigue ejecutando las sig instrucciones, siendo interrumpido por el módulo de E/S cuando éste ha completado su trabajo. Pueden ser instrucciones del mismo proceso en el caso que ese proceso no necesite esperar hasta que se complete E/S, caso contrario se suspende el proceso en espera de la interrupcion y se realiza otro trabajo

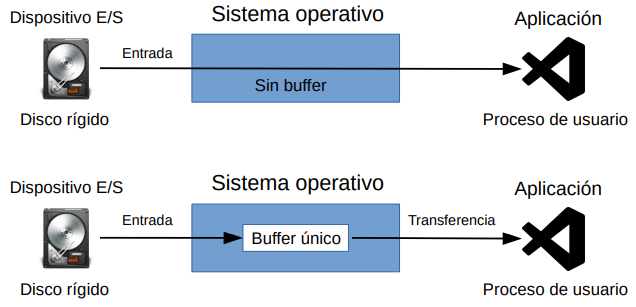
**~Acceso directo de memoria:** (DMA = direct memory access) este modulo controla el intercambio de datos entre la mem principal y un modulo E/S.

**Técnicas E/S:**

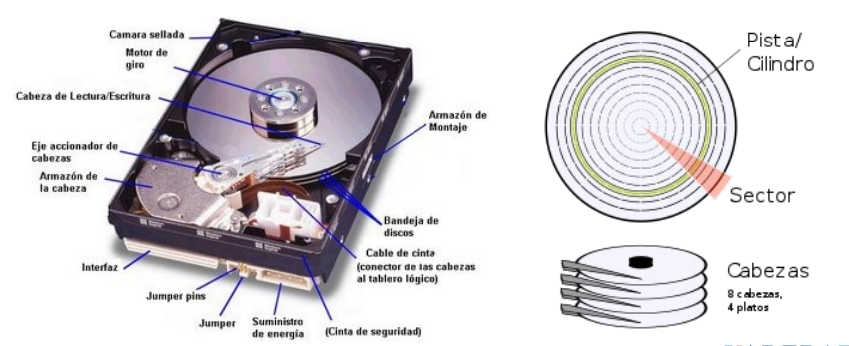


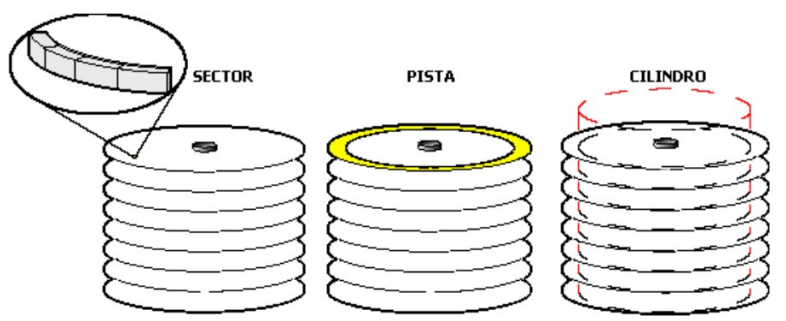


**Buffer:** es un área de memoria q almacena datos temporalmente mientras se transfieren entre dispositivos o entre un dispositivo y una aplicación



**Planificación de disco:**





**Tiempos de acceso a un sector del disco rígido**

Tiempo de acceso a un sector = tiempo de búsqueda + latencia rotacional + tiempo de transferencia

* **Tiempo de búsqueda:** es el tiempo que le lleva al disco posicionar el brazo en la pista (track) deseada.
* **Latencia rotacional:** es el tiempo en que tarda el sector en posicionarse debajo de la cabeza de lectura/escritura.
* **Tiempo de transferencia:** es el tiempo necesario para mover los datos

**Algoritmos de planificación de discos:** su objetivo es minimizar los tiempos de acceso

FCFS(FIFO), SSFT, SCAN, C-SCAN, LOOK, C-LOOK (este es el más rápido), FSCAN Y N step SCAN

FSCAN: usa dos colas de pedidos de acceso (activa y pasiva), se procesan los pedidos de la cola activa haciendo uso del algoritmo SCAN.

Los nuevos pedidos se incorporan siempre a la cola Pasiva.

Cuando se procesan todos los pedidos de la cola Activa, la Pasiva pasa a ser Activa y la Activa pasa a ser Pasiva (rotan)

N step SCAN: usa N colas de pedidos, los nuevos pedidos se incorporan a varias colas Pasivas de tamaño N.

Los pedidos de la cola Activa se atienden con el uso del algoritmo SCAN, cuando se procesan todos los pedidos de la cola Activa, una de las Pasivas pasa a ser la Activa y la Activa pasa a ser una Pasiva más (también rota)

**Raid:** La matriz redundante de discos independientes (RAID) es un método que mejora el rendimiento de varios discos de almacenamiento conectados a una computadora.

Características: varios discos físicos se muestran como un disco lógico, los datos se almacenan de forma distribuida (stripping), tiene redundancia para reconstruir los datos.

Puede ser por software (flexible) y hardware (rendimiento), brinda un mayore rendimiento y más seguridad.

Niveles de raid:

* Estándares: del 0 al 6
* Anidados: RAID 0+1, 1+0, 30, 50
* Propietarios: Linux MD y RAIDZ

Los más utilizados:

**Raid 0:** conjunto dividido (2 discos como mínimo)

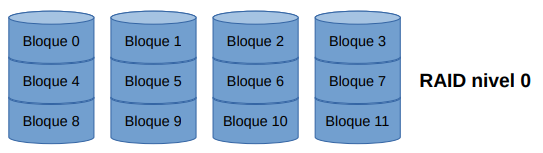
**Raid 1:** conjunto espejo (2 discos como mínimo)

**Raid 5:** conjunto dividido con una paridad distribuida

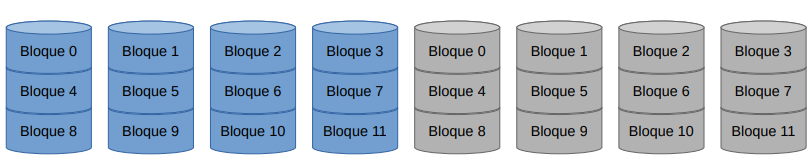
(3 discos como mínimo)

Todos los niveles:

*RAID 0:* No tiene redundancia (BLOQUE = BANDA)



*RAID 1:* No tiene redundancia



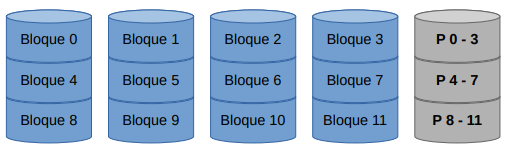
*RAID 2:* Redundancia por método de haming, bandas a nivel de bit. Acceso paralelo y se sincronizan los discos.



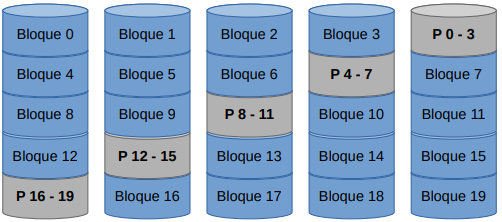
*RAID 3:* redundancia por paridad, bandas a nivel de bit , acceso paralelo y se sincronizan los discos



*RAID 4:* redundancia por paridad y acceso independiente



*RAID 5:* redundancia por paridad y acceso independiente



*RAID 6:* tiene doble redundancia por paridad y acceso independiente

