**Clase de Repaso-Temas hablados**

1-**Concepto de Protección**

**Paginación simple**

**Dijo que pertenece a la materia arquitectura y que no profundizamos**

**Puede ser que el bit de validez no sea obligatorio**

**Address Space ID lo usamos cuando no podemos usar el bit de validez**

**un proceso puede tener uno o varios espacios de direcciones.**

**no obligatoriamente son contiguos**

**El bit de validez lo usamos para apagar las otras tablas de página**

El problema que en el espacio de alocación de tabla de página, es contiguo.

Por lo tanto para saber la direccion que debo leer la cpu viene con una dirección lógica

y va a la PTBR.

La PTBR para cada proceso tiene la dirección base,la dirección en la ram donde empieza.

-Si hay un programador escribe código en absoluto.

**La cuenta es:**

(sale de la PTBR)dirección base + numero de pagina x longitud de registro

Si el programador pone un número de página mayor a la longitud del proceso, entonces si se excede va a parar a otro proceso.

También por overflow si le ponemos un número muy excedido se corre un bit y hasta que puede ir al kernel.(se hace una cuenta que tiene una dirección muy alta)

También podemos pasar a otra página cuando se excede

**Solución:**

**El bit de validez de dirección base**

Uso lo bit adicionales para proteger, cuando se activa un proceso le voy a poner en “1” a cada uno de los elementos de la tabla de página, y los otros en “0”

**EN MEMORIA VIRTUAL**

Lo usamos para apagar o dar como inválido, en la tabla de página una dirección base de una página no cargada.

**Están los 2 bits:el validez base(tiene que estar si o si para saber si se cargo la pagina) y lectura**

#EL MISMO BIT DE VALIDEZ LO USAMOS EN MEMORIA VIRTUAL

lo usamos cuando generamos una tabla de página ponemos en 0 y ponemos el 1 aquellos marcos, la memoria siempre tiene una dirección cargada.Puede ser que sea todo ceros pero es una dirección.

**Bit de lectura o escritura**

El mismo criterio se utiliza otro bit, en lo caso cuando debemos proteger que una página sea de solo lectura.

Si una rutina está escrita en varios procesos, no podemos sobreescribir los otros procesos.

esto para páginas compartidas

ejemplo:

2- **¿Porque se produce la paginación Jerárquica?**

Se produce porque la tabla de página pasa a ser muy grande

Cada proceso tiene una tabla de página, propiamente hablada.Hay un espacio que se guardó el kernel para guardar la tabla de páginas.

Cuando las tablas de páginas son muy grandes osea tienen cientos de miles de entradas

tengo un problema de fragmentación externa, termina un proceso y entra otra y deja un hueco en la memoria.

Como son computadoras que no se apagan la 24 hs, se pierde mucho espacio de memoria.

Es contiguo porque estoy trabajando con la PTBR.

Definición de el:

Se le asigna a cada proceso una página externa y esa pagina externa es la que va a la PTBR y va a leer la pagina externa, toma la dirección base del proceso que se está ejecutando y hace la cuenta.

**3-¿El numero de pagina denuevo se multiplica por la longitud?**

Si pero eso para la página externa(lee todos los registro donde estan cada una de las direcciones base)

(sale de la PTBR)dirección base + numero de pagina x longitud de registro

**Ejemplo que hizo:**

**cuando tenes muchos archivos en un mueble y ahora tenes 10**

**mueble 4 página 80**

página externa que numero de mueble

página interna que carpeta dentro del mueble

página externa te activa uno

página interna te dice que leer

4-**¿Las páginas externas pueden llegar a tener distintas longitud?**

No, porque tenes una tabla de páginas externas.

Una sola tabla de pagina externa y muchas tablas páginas internas.

5-**Tablas de páginas INVERTIDA (Tablas de Marcos)**

Introducción

Cuando vos tenes un nivel de memoria virtual muy grande como en el caso de mainframe

que multiplican por 20 o distinto. Una manera de trabajarlo es con la tablas de página Invertida.

Lo que hace:

Es decirte para cada marco, que pagina de qué proceso está cargado ahi.

Tenemos cargado la tabla de marco, que pagina de qué proceso está cargado es una sola y se guarda en la MMU, hay una copia en la RAM.

P4,proceso 5(leo los registro)

Lectura

Es un hardware especial como la TLB que permite la lectura simultánea de todos los datos cuando está en la MMU. Hoy en día la MMU son parte del procesador(Intel i3,i5).

La TLB está en la MMU

¿Porque esta la copia en la Ram?

la razón es que cantidad de registro que tenes en la MMU es limitada.

Si tu ram creció más que la cantidad de registro que tiene, trabaja contra la ram.

6- **Requerimiento de paginación jerárquica y segmentación paginada Duda del Parcial.**

Los requerimientos de hardware son la PTBR y la TLB para paginacion.

Se dicen que son de hardware porque son registró especiales, diseñados para su uso dentro de la MMU.

Se guarda cuando se diseña la MMU,se diseña las tablas correspondiente

En el caso de la TLB se trata de registros asociativos, no se hace una lectura secuencial.

Segmentación simple pura no está implementada

Se necesita un hardware para la administración de requerimientos, que es el hardware Trap que compara la longitud del desplazamiento con el tamaño de segmento.

Segmentación paginada

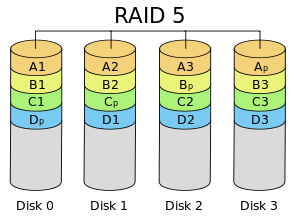
STBR es el equivalente para segmentos, se lo suele mencionar PTBR porque es el término.

7-**¿Quién tiene registro asociativos?**

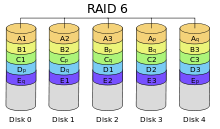
La TLB tiene registros Asociativos para el uso de la PTBR.

8- **RAID 5 y 6, ¿como distingo cuanto fallos de discos me permite tener?**

El Raid 5 por definición te permite uno solo,cualquiera.



El Raid 6 por definición te permite dos , cualquiera.(No se suele usar)



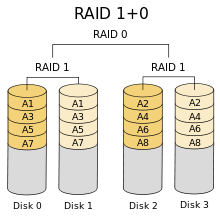
**Ejemplo de práctica:**

Como se sabe en una empresa aérea la información es de vital importancia, por lo tanto dichos datos están alojadas en lo que se denominan storage de disco. Decir que tipo de RAID de disco aconsejaría, además explicar cómo funciona el tipo de RAID elegido.

**Resolución:**

**Lo mejor es un 0+1( espejo de divisiones)**

Consiste básicamente en un nivel principal de tipo RAID 1 que hace las funciones de replicar los datos que se encuentran en un primer subnivel en un segundo. A su vez existirá un subnivel RAID 0 que hará las funciones propias de éste, es decir, almacenar los datos de forma distribuida entre las unidades que en él se encuentran.



El despejado no me baja la performance de consulta.

Si no nos permitieran usar una combinación de los raid, podríamos usar un raid 6.

**#Esto no puede tomar porque el raid diferido no se vio en la cátedra.**

Raid 4

Es de escritura y entra en una cola.Es un disco mas cada 10 discos,va a estar espejado.

Raid 0+1 y afuera el 4,el bit de paridad lo guardas en el raid 4 porque está diferido que no afecta a la cola.

El problema del raid 4: Toda operación de ejecución está terminada, cuando graba en el último y se hace un cuello de botella. (No sería un 4 puro, es un 4 diferido) para perderlo se te tiene que romper el original y el espejo.

El problema de meterle un 5, es que cada operación se empieza desparramar porque está dividido en segmentos.

9- **Función Hash y tabla invertida.¿Pueden soportar las 2 el procesador?**

Eso está definido por el fabricante, hoy en dia los procesador de 64 bits usan la funcion hash y la tabla invertida.La tabla invertida se suele usar para rutinas compartidas y no para procesos principales.

10- **La Fat tiene una fila por cada sector del disco.**

La fila pueden tener varias columnas,puedo agregarle columnas.Minimo puedo agregarle “uno” porque el puntero es la cadena de ida, con uno alcanza.

Se coloca las 2 de ida y vuelta porque si se pierde un dato lo puedo recuperar.

Cuando se formatea por fat ,se guarda la fat y también se graban los punteros en cada sector. Si perdimos la FAT puedo recuperarla leyendo el disco.

11- **Administración de espacio libre en el FILESYSTEM ¿Nosotros que teníamos?**

**Espacio asignado y espacio libre.**

La fat soporta los archivos libre por el archivo empty.Cada vez que necesito un sector se lo saco al archivo empty, porque todos los archivos están encadenados.

Voy a tener una lista encadenada en la fat por cada uno de los archivos más una lista por cada elemento vacío,soporta no solo la asignación de espacio asignado, sino también la asignación del espacio libre.

En los sistemas indexados voy a tener una lista encadenada simple, con agrupación o una con agrupación contigua.

Ejemplo:

Si tenemos una fila por cada sector del disco, va a ver sectores que no van a estar ocupados por el archivo empty.

Hay una entrada por cada sector del disco, o está asignado a un archivo o está asignado al archivo empty.

Al principio se le asigna todo al archivo empty, en la tabla de la fat tendríamos “fat name empty (puntero) ” siempre existe el archivo empty

12- **En agrupación,también es lista encadena pero ¿qué sucede?**

En agrupación lo que sucede es que tiene una pre asignación de sectores.

La ventaja que tiene es la contigüidad en el momento de la escritura evita la fragmentación del archivo en todo el disco.

Los cabezales no van sector por sector, hay 5 están:

1- FC FS

Va a ver una cola que todo lo que me importa es la pista, solo mueve la cabeza a la proxima que este en la cola. Va estar saltando. (NO SE USA)

2- Scan(concepto)

El brazo del disco empieza en uno de los extremos del disco y se mueve hacia el otro extremo,sirviendo solicitudes que ya estaban esperando hasta que llega al otro extremo.Donde el movimiento se invierte y el servicio continua.

El problema es que hay pistas favorecidas.

3- C-scan (concepto)

Proporciona un tiempo de espera más uniforme que el Scan.

La cabeza se mueve desde un extremo al otro.Sirviendo solicitudes en su camino.

Cuando alcanza el otro extremo, inmediatamente retorna al principio del disco,sin

servir solicitudes en el viaje de regreso.

trata a los cilindros como una lista circular que se envuelve desde el último cilindro

al primero. (Promedio tarda la mitad)

4-look

Lo mismo en forma circular pero directo.

5- C-Look

El brazo solo va tan lejos como este la última solicitud en cada dirección, entonces

se devuelve sin ir al extremo.(Si no hay nadie se vuelve)

12- **¿Se puede tener algoritmo distintos para cada cabezal?**

No, porque tenemos un solo cabezal tenemos un solo algoritmo

**Mencionó:**

No importa en el caso de contigüidad, no hay problema severo de fragmentación que hace que un archivo esté al lado del otro,.

Si nos importa que si cada archivo está fragmentado pero no me importa que si disco está fragmentado.

13- **Filesystem Actual**

Los filesystem actual van a pre asignarle típicamente 8 sectores cuando se crea un archivo del disco. Cuando se supera el 70 - 75 % te pide la mitad de lo que se pre asignó para tener memoria libre.

Agrupación:

En un sector lo utilizo para guardar muchos punteros cada uno a uno libre.

Cantidad libre de conteo:

Cada sector tiene un solo puntero que es al proximo libre, me obliga a un recorrido muy grande.

14-**¿Lo filesystem tienen mecanismos de seguridad para los archivos?**

Se vieron en la práctica, tiene permiso,dueño, grupo.

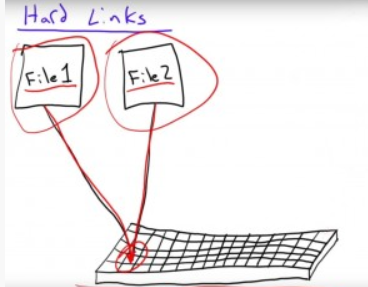
También los filesystem pueden encriptar los datos,depende de cada filesystem

15- **Link duros y blandos (Simbólicos y lógicos )**

Cuando es un link duro pueden ser que tenga el mismo archivo tenga 2 nombres

**Enlaces duros(Simbólicos):** es un archivo que apunta al mismo contenido almacenado en disco que el archivo original.

Por lo tanto los archivos originales y los enlaces duros dispondrán del mismo inodo y consecuentemente ambos estarán apuntando hacia el mismo contenido almacenado en el disco duro.



**ventajas:**

Los enlaces duros ocupan menos tamaño en el disco duro que los enlaces simbólicos

El acceso al contenido a través de un enlace duro es más rápido que en los enlaces simbólico.

Realizar copias de seguridad incrementales ahorrando espacio en disco duro y un tiempo considerable

**Enlaces blandos(Lógicos)**:los enlaces simbólicos apuntan al nombre de un archivo y posteriormente el archivo apunta a un contenido almacenado en nuestro disco duro.

cada enlace simbólico dispone de su propio número de inodo y es diferente al del archivo original. Por lo tanto podremos crear enlaces simbólicos de archivos y de carpetas aunque estén en discos duros diferentes o en particiones diferentes.

**#Salvo que seas root no podes garantizar que un archivo fue borrado.**

Windows soporta link blando.

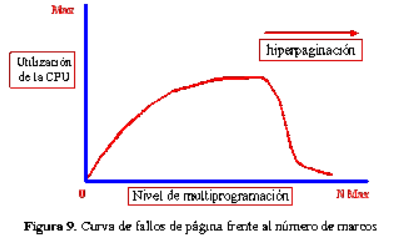
Unix soporta link duros y blandos.

16**¿Si le hago un remove al link?**

Si lo hice sobre mi link, se borro el link sea duro y blando.Si soy el que crea el archivo pero hubo un link duro, tiene copiada la cantidad.

17 **¿Qué es thrashing y cómo podemos evitar caer en eso?**

Si el número de marcos asignados a un proceso de baja prioridad desciende por debajo del número mínimo requerido por la arquitectura del computador, debemos suspender la ejecución de ese proceso.



***#se genera con el segmento de dato, nunca con lo de código.***

**Cómo evitar caer en thrashing**

**MODELO SERVIDOR**

Modelo de conjunto de trabajo:

Es la herramienta para reemplazar el segmento en la paginación jerárquica o simple, como no sabemos cuales son las páginas de la misma rutinas, miramos para atrás para saber cuales son las páginas más solicitadas.

No lo necesita en segmentación porque el segmento me guarda la direccion de pagina.

Ejemplo de matriz de un millón por un millón(no entra porque necesita mucho):

Si el sistema operativo pertinentemente va a pre sacrificar mientras su pila de macos libres, esté vacía.

Modelo de porcentaje de fallas:

Si un proceso supera el 5% o 2% constante de fallo de página

**#EN EL MODELO DE USUARIO NO PUEDE EVITARLO**