Hashing:

El mecanismo que trata de asegurar una recuperación rápida de registros, en un solo acceso promedio, lleva el nombre de dispersión o hashing.

Definiciones de hashing o dispersión:

Tecnica para generar una dirección base única para una clave dada. La dispersión se usa cuando se requiere acceso rápido mediante una clave.

Tecnica que convierte la clave asociada a un registro de datos en un numero aleatorio. El cual, posteriormente es utilizado para determinar donde se almacena dicho registro.

Tecnica de almacenamiento y recuperación que usa una función para mapear registros en direcciones de almacenamiento en memoria secundaria.

La técnica de dispersión, presenta una serie de atributos:

* No se requiere almacenamiento adicional. No es necesario tener una estructura auxiliar que actue como soporte para poder acceder rápidamente a la información.
* Facilita la inserción y eliminación rápida de registros en el archivo. En general, con un solo acceso a disco un registro puede ser insertado o eliminado del archivo.
* Localiza registros dentro del archivo con un solo acceso a disco.
* Si bien no es posible asegurar que cualquier registro sea encontrado en un solo acceso, la gran mayoría de las búsquedas, serán efectivamente con un solo acceso a disco.

Limitaciones de la técnica de dispersión:

* No es posible aplicarla en archivos con registros de longitud variable.
* No es aplicable la función de Hash, sobre una clave secundaria.

Tipos de dispersión:

Se denomica **Hashing** con espacio de direccionamiento **estatico**, a aquella política donde el espacio disponible para dispersar los registros de un archivo de datos, está fijado previamente.

Se denomina Hashing con espacio de direccionamiento dinamico, , a aquella política donde el espacio disponible para dispersar los registros de un archivo de datos, aumenta o disminuye en función de las necesidades de espacio que en cada momento tiene el archivo.

Parametros de la dispersión:

* Funcion de Hash
* Tamaño de cada nodo de almacenamiento
* Densidad de empaquetamiento
* Metodos de tratamientos de desbordes (Overflow)

Primer parámetro: Funcion de Hash

Primera definición: Esta función puede verse como una caja negra, que recibe como entrada una clave, y produce una dirección de memoria donde almacenar el registro asociado a la clave en el archivo de datos.

Segunda definición: Es una función que transforma un valor, que representa una llave primaria de un registro, en otro valor dentro de un determinado rango, que se utiliza como dirección disica de acceso para insertar un registro en un archivo de datos.

Como la función hash, podría retornar cualquier valor, es necesario, previo a retornar el resultado final, mapear dicho valor dentro del rango de valores posibles.

Uno de los objetivos fundamentales cuando se utiliza el método de dispersión o hashing, es la selección de la función de hash. Esta función debe esparcir los registros de la manera mas uniforme posible, es decir, que cada clave tenga asignada una dirección física distinta.

Si una función de hashing asigna a dos registros las mismas direcciones de memoria, ambos registros serian sinónimos para esa función de hashing.

Pero además, si dos registros tienen la misma dirección de memoria, se generaría una colision.

Entnces, las colisiones ocurren cuando dos o mas elementos diferentes tienen la misma dirección de hash.

Tambien ,en las funciones hash, puede ocurrir overflow.

Se produce overflow, si un nodo tiene la capacidad para un solo elemento y se produce una colision (se le asigna dos registros a esa direccion), se produciría overflow, porque el nodo no es capaz de almacenar dos elementos, sino que tiene la capacidad de almacenar un solo elemento.

Segundo parámetro: Tamaño de cada nodo de almacenamiento.

Dentro de los parámetros que afectan la eficiencia del método de dispersión esta presente el tamaño o capacidad del nodo.

El tamaño de un nodo, puede tener capacidad para mas deun registro

A mayor tamaño de almacenamiento, menos overflow y mayor fragmentación.

Si tenemos gran tamaño de almacenamiento por nodo, disminuye las posibilidades de tener overflow, pero con el costo de que vamos a tener espacio sin asignar, por eso tenemos fragmentación. Mientras menos overflow tenga, mas eficiente es el acceso, pero estamos pagando el costo de la fragmentación, por eso no hay un caso que sea el optimo, la respuesta correcta seria depende. Porque en algún lado ganamos, pero siempre por otro lado perdemos.

Tercer parámetro: Densisas de empaquetamiento:

Se define la densidad de empaquetamiento (DE) como la relación entre el espacio disponible para el archivo de datos y la cantidad de registro que integran dicho archivo.

DE = numero de registros del archivo / capacidad total del archivo

A menor densidad de empaquetamiento, menos overflow, y mas desperdicio de espacio.

Las colisiones aumentan a medida que el archivo se llena cada vez mas, ya que cuando hay mas registros almacenados, la probabilidad de que dos o mas registros se asignen a la misma dirección de hash (colision) aumenta.

Soluciones para las colisiones:

* Esparcir registros: Buscar métodos que distribuyan los registros de la forma mas aleatoria posible
* Usar memoria adicional: Distribuir pocos registros en muchas direcciones, baja la densidad de empaquetamiento.

-Disminuye las colisiones y por ende el overflow

- Desperdicia espacio

- Colocar mas de un registro por dirección: Direcciones con n claves, mejoras notables.

Este tema que viene ahora, no lo entendí muy bien, mas que nada el tema de las probabilidades.

Probabilidad que un nodo (cubeta) tenga k elementos:

Para calcular la probabilidad, primero se necesita tener la densisas esperada.

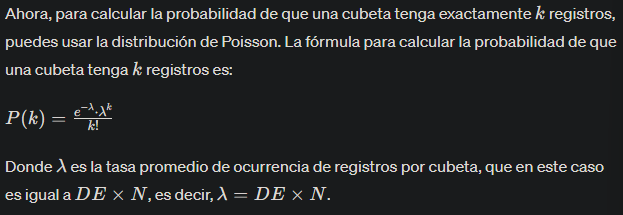
R = (numero de registros del archivo)

N= (capacidad del archivo)

DE = ( R / N )

Ejemplo:

N = 1000 , R = 1000 =>>> DE = 1000 / 1000 = 1 %100

 A = DE x N

Calculamos P(0) , reemplazamos La K, por el 0: 