

Tema 5

El Nivel Interno



Apartado 3

Métodos de organización y acceso a los datos: Acceso Directo

Contenidos

- Acceso directo
- Hashing básico
- Hashing dinámico

Contenidos

- Acceso directo
- Hashing básico
- Hashing dinámico

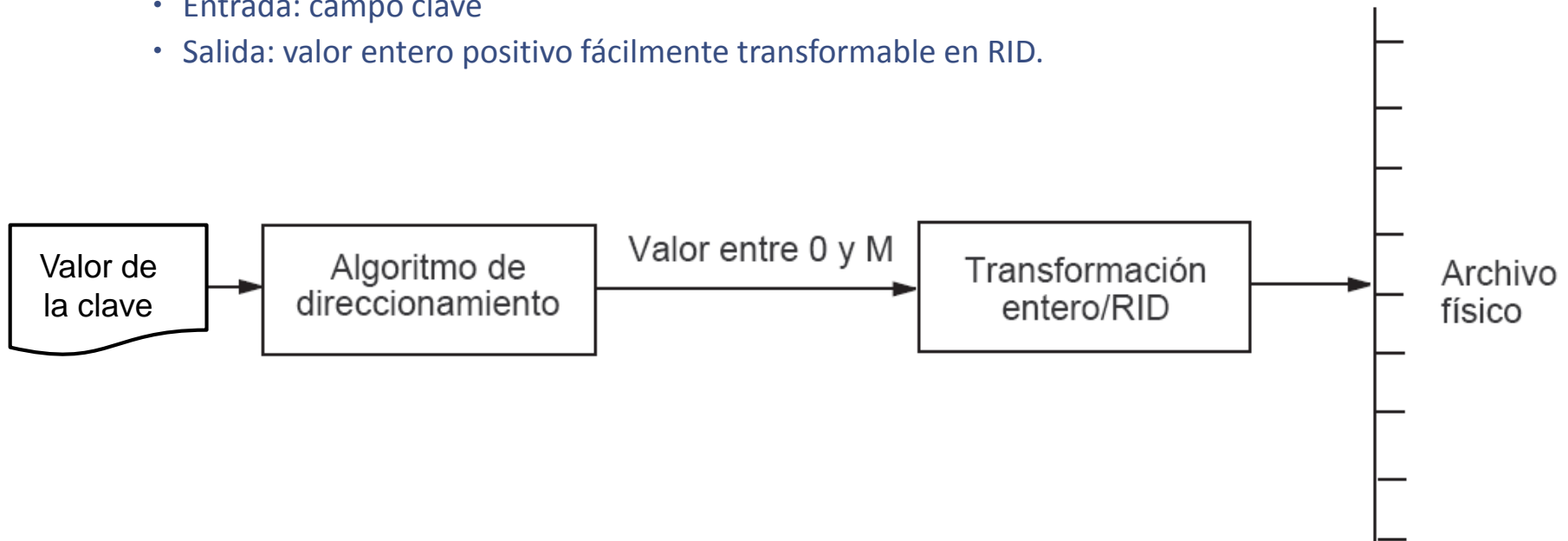
Acceso directo

- Otra forma de acceder a un registro almacenado
 - No hay una estructura adicional
 - Se usa un algoritmo que nos indique directamente la posición del registro deseado.
- Acceso directo:
 - Calcular directamente la dirección de un registro mediante la aplicación de algún algoritmo o función sobre un campo determinado del mismo.
 - El campo debe identificar unívocamente al registro

Acceso directo

- Funcionamiento:

- Normalmente no es posible establecer una clave física que sea totalmente correlativa y única para cada registro.
- Hay que buscar un algoritmo que transforme los valores de un cierto campo en una dirección.
 - Entrada: campo clave
 - Salida: valor entero positivo fácilmente transformable en RID.



Acceso directo

- Los algoritmos de direccionamiento no suelen mantener el orden de la clave.
 - Los registros no están almacenados según el orden de su clave física.
 - Problemas con la recuperación por intervalos.

Acceso directo

- Hay una gran variedad de algoritmos:
 - Dependen del tipo de clave:
 - Si la clave es alfanumérica, hay que transformarla a un valor numérico.
 - Suelen estar basados en un mecanismo de generación de números pseudoaleatorios:
 - *Cuadrados centrales:*
 - Se eleva la clave al cuadrado y se eligen tantos dígitos centrales como sea necesario.
 - *Congruencias:*
 - Se divide la clave por M y se toma el resto. (M suele ser primo).
 - *Desplazamiento:*
 - Se superponen adecuadamente los dígitos binarios de la clave y luego se suman.
 - *Conversión de base:*
 - Se cambia la base de numeración y se suprimen algunos dígitos resultantes.

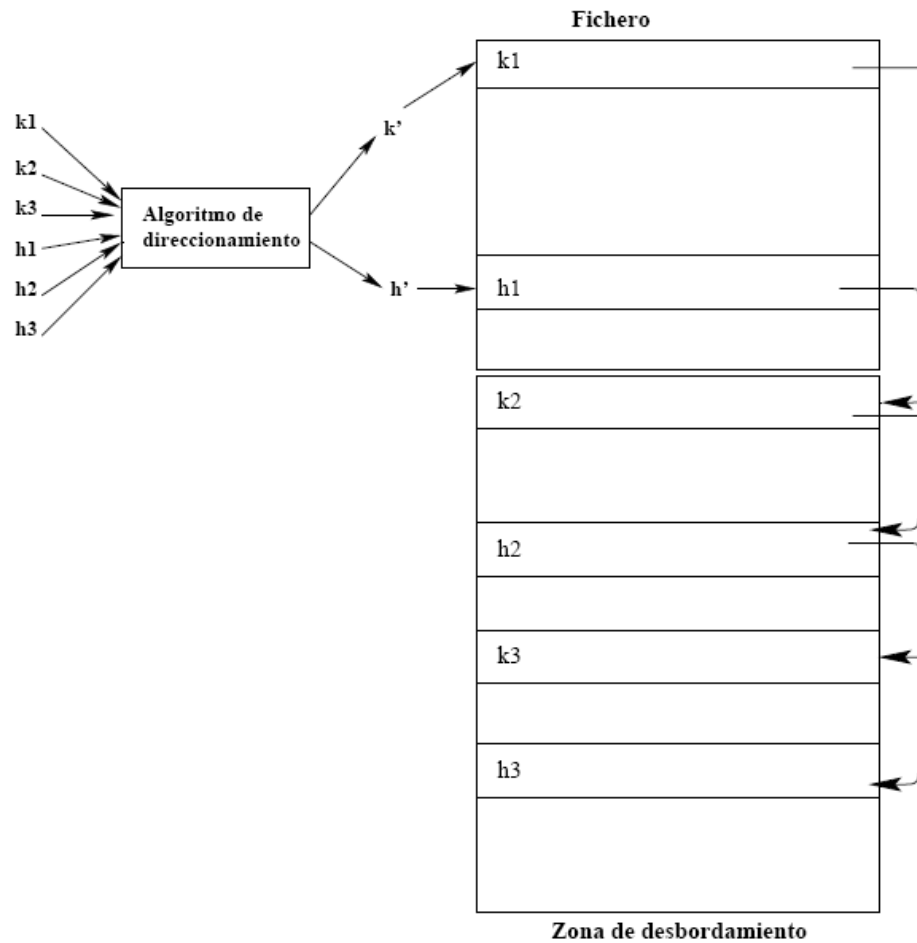
Acceso directo

- Problemas
 - Salvo que el campo clave se diseñe para ello, es prácticamente imposible encontrar una transformación que dé un valor entero positivo en un rango de valores limitado tal que:
 - No haya dos valores distintos de clave que den lugar al mismo número.
 - Colisiones.
 - Los algoritmos producen huecos:
 - Zonas vacías del rango de salida, no asignadas por el algoritmo.
 - Se traducen en huecos en el fichero de datos.

Acceso directo

- Para gestionar colisiones y huecos:
 - Combinar el acceso directo con una gestión mediante listas de colisión:
 - Zona de desbordamiento
 - Colisión:
 - El registro problemático se almacena en la zona de desbordamiento.
 - Los sinónimos (registros con claves que producen colisión) se conectan mediante una lista.

Acceso directo



Acceso directo

- Si crecen las listas de sinónimos:
 - El acceso directo puro no resulta adecuado.
 - Mantener listas.
 - Zona de desbordamiento casi como el fichero original.
- Han aparecido técnicas más sofisticadas:
 - Hashing

Contenidos

- Acceso directo
- Hashing básico
- Hashing dinámico

Hashing básico

- Si el problema principal es que los valores de las claves no están uniformemente distribuidos en el intervalo $[0, M]$:
 - Se acumulan en una parte de este intervalo.
 - Solución:
 - Asignar más espacio a esa parte del intervalo.
- Técnica:
 - Se divide el espacio del fichero en “cubos” (buckets).
 - El algoritmo de direccionamiento asigna cubos, no direcciones concretas.
 - En cada “cubo” puede haber más de un registro.
 - Ciertos rangos de valores tienen asignados más cubos que otros.
 - Se complementa con el uso de “cubos de desbordamiento”.

Hashing básico

- Parámetros:
 - Número de cubos.
 - Tamaño de los cubos (relación con bloques físicos)
 - “slots”
 - La transformada clave/dirección, que debe tener en cuenta la distribución de la clave según rangos para que unos cubos no se llenen mucho y otros se queden muy vacíos.

Hashing básico

- Para insertar un registro:
 - Transformar la clave.
 - Localizar el cubo correspondiente.
 - Si hay sitio se inserta el registro y hemos terminado.
 - Si no hay sitio, se sitúa el registro en un cubo de desbordamiento conectándolo con el cubo que realmente le corresponde mediante punteros.

Hashing básico

- El proceso de búsqueda:
 - Transformar la clave.
 - Localizar el cubo correspondiente.
 - Realizar una búsqueda secuencial dentro del cubo.
 - Si hemos encontrado el registro, el proceso termina.
 - En caso contrario, se impone “un barrido por punteros” a través de los cubos de desbordamiento.

Contenidos

- Acceso directo
- Hashing básico
- Hashing dinámico

Hashing dinámico

- El hashing básico sigue teniendo problemas:
 - Es necesario conocer la distribución previa de las claves para asignar adecuadamente los cubos.
 - En otro caso siguen apareciendo huecos/colisiones.
 - Al aumentar el número de registros, aumentan los registros en páginas de desbordamiento.
 - Se hacen necesarias las reorganizaciones.

Hashing dinámico

- Solución:
 - Trabajar de forma dinámica
- Se parte de una configuración uniforme y de pocos cubos
- Los restantes, se van generando conforme se necesiten.
 - Se asignan a los rangos conforme la afluencia de registros lo demanda.
 - Hashing dinámico o extensible

Hashing dinámico

- Técnica:

- El valor transformado del campo clave nos lleva a la entrada de una tabla índice que se almacena en memoria.
- Allí está la dirección del cubo donde se encuentran los registros que tienen asociado este valor transformado.
- Puede ocurrir que varias entradas de la tabla conduzcan al mismo cubo.

- Proceso:

- Inicialmente, todas las entradas apuntan al mismo cubo.
- A medida que vamos insertando registros, se van generando nuevos cubos y cambiando las salidas de la tabla índice.

Hashing dinámico

- Algoritmo de Hashing Dinámico

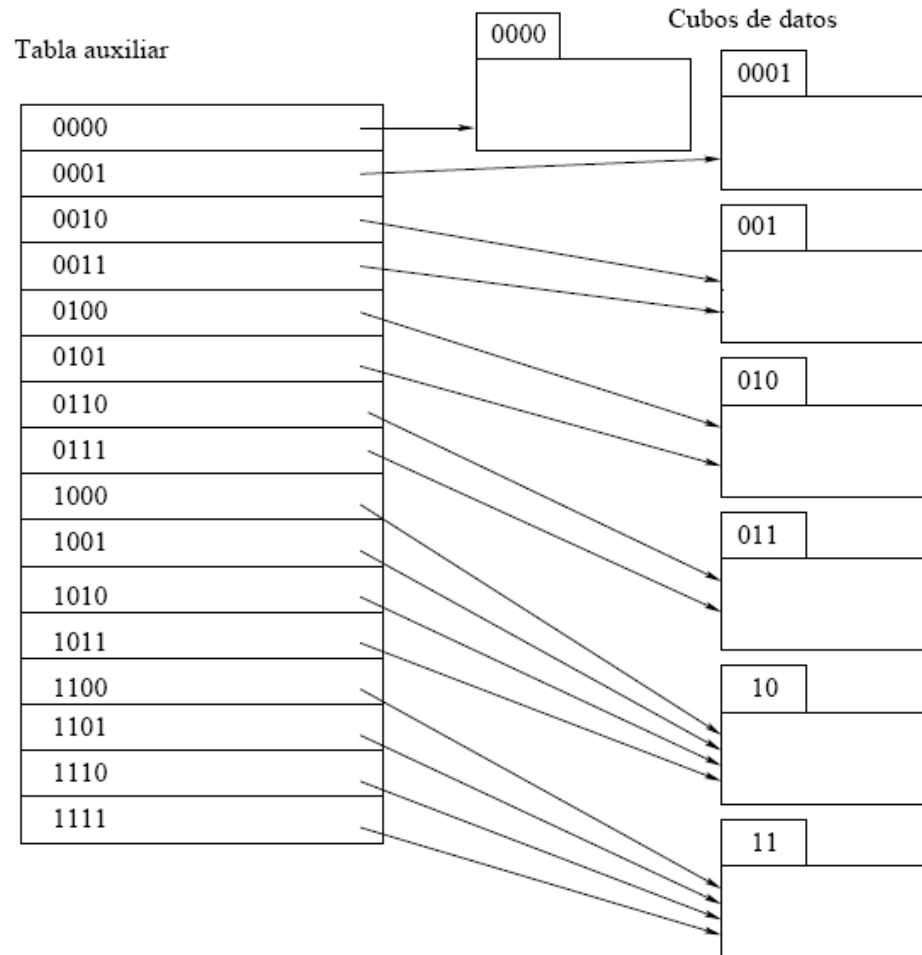
- Datos de partida:

- k = clave física para direccionar
- $k' = h(k)$ valor entero entre 0 y M
- n = número de bits que tiene k' en binario
- $d \leq n$, los d primeros* dígitos de k' seleccionan el cubo donde está el registro y se llaman pseudollave.
- $b < d \leq n$, inicialmente el archivo tiene 2^b cubos distintos, como máximo tendrá 2^d . Si son necesarios más, hay que aumentar d .

Hashing dinámico

- Algoritmo
 - Se considera una tabla índice en memoria con 2^d filas
 - En la primera columna de esta tabla (valores de campo clave) se sitúan todas las posibles sucesiones de d dígitos binarios
 - d es la “profundidad global” de la tabla
 - Todos los cubos tienen en principio “profundidad local” igual a b
 - En principio, todas las entradas cuyos b primeros dígitos son iguales apuntan al mismo cubo.
 - Allí se almacenan los registros cuyo valor de k' tiene esos b primeros dígitos.
 - Cuando se llena un cubo se divide en 2, poniendo en uno de ellos los registros con el dígito $b+1$ de k' a 0 y en otro los que lo tienen igual a 1. La profundidad local de estos cubos aumenta una unidad.

Hashing dinámico



Hashing dinámico

- El hashing dinámico supera los problemas clásicos del acceso directo.
- También tiene sus inconvenientes:
 - Utilizar una tabla índice adicional (nuevos accesos a disco si no cabe en memoria).
 - El tamaño de la tabla depende de “d”.

Contenidos

- Acceso directo
- Hashing básico
- Hashing dinámico