

ESCUELA DE PREPARACIÓN DE OPOSITORES
E. P. O.

C/. La Merced, 8 – Bajo A Telf.: 968 24 85 54
30001 MURCIA

INF13 – SAI13

Ficheros. Tipos. Características. Organizaciones.

Esquema.

1	INTRODUCCIÓN.....	1
2	CONCEPTOS BÁSICOS.....	2
2.1	ALMACENAMIENTO DE LA INFORMACIÓN.....	2
2.2	EL GESTOR DE FICHEROS.....	3
2.3	RENDIMIENTO DE LA TRANSFERENCIA DE LA INFORMACIÓN.....	4
3	TIPOS DE ARCHIVOS.....	4
3.1	ARCHIVOS PERMANENTES.....	4
3.2	ARCHIVOS TEMPORALES.....	5
4	ORGANIZACIÓN DE ARCHIVOS.....	5
4.1	ORGANIZACIONES CONSECUTIVAS.....	6
4.2	ORGANIZACIONES DIRECCIONADAS.....	7
4.2.1	Métodos de direccionamiento.....	7
	Direccionadas directas.....	7
	Direccionadas dispersas.....	8
4.2.2	Algoritmos de direccionamiento.....	8
	Direccionamiento directo.....	9
	Direccionamiento asociado.....	9
	Direccionamiento calculado (hashing).....	9
4.3	ORGANIZACIONES INDEXADAS.....	10
4.3.1	Archivos de índices de árbol B ⁺	12
	Estructura de árbol B ⁺	12
	Organización de archivos con árboles B ⁺	13
4.3.2	Archivos de índices de árbol B.....	14
5	CONCLUSIONES.....	15

1 Introducción.

Los sistemas informáticos requieren, para la manipulación de la información, que ésta se encuentre previamente almacenada en alguno de los dispositivos de almacenamiento que estos sistemas reconocen. Si esta información puede ser obtenida en tiempo de computación, bien mediante un cálculo operacional, o bien mediante la entrada desde un terminal de entrada/salida y, este proceso tiene un bajo coste computacional y, además, puede ser reproducido, esta información puede ser almacenada en la memoria central de la computadora con el conocimiento de que será perdida una vez el proceso que manipula esa información abandone la memoria central de la computadora.

Pero son muy pocos los problemas en los que el dominio de la información satisface estas características. En la mayoría de los casos el volumen de información es tan elevado, o los procesos de cómputo, para obtenerla son tan costosos, que el alimentar a los procesos de manipulación de esa información con ésta en cada una de sus ejecuciones resulta algo inconcebible. Es en estos casos cuando el usuario de los sistemas informáticos requiere de los ficheros para el almacenamiento de la información que forma parte del dominio del problema que está tratando con un sistema informático.

Un fichero es una agregación de datos estructurados que es vista por el sistema operativo de la computadora como un todo, como una unidad estructural sobre la cual se pueden realizar una serie de operaciones dependiendo del dispositivo físico en el cual se encuentre y de la propia estructura interna de los datos en el mismo.

El conjunto de datos que forman parte de los ficheros se agrupan en unidades denominadas registros cuya función es representar a unidades de información del problema que se está tratando. El número y estructura de los datos que forman parte de los registros es determinado en la fase de análisis y diseño lógico del problema, mientras que la disposición de los registros y los datos en los ficheros (su estructura) es determinada en el diseño físico.

Existen múltiples estructuras diferentes en las cuales puede almacenarse la información dando lugar a lo que se conoce como organizaciones de la información y, en base a esas organizaciones los procedimientos que manipulan esa información son diferentes.

Este tema vamos a estudiar las organizaciones de ficheros que permiten estructurar la información correspondiente a los problemas para que pueda ser manipulada con un desempeño aceptable. Si bien las organizaciones pueden ser tan diferentes como las necesidades lo requieran y sería imposible atender a todas las posibles variaciones, sí se intentan cubrir aquellas organizaciones básicas en las cuales están basadas las demás.

2 Conceptos básicos.

Un fichero es una agregación de datos estructurados que es vista por el sistema operativo de la computadora como un todo, como una unidad estructural sobre la cual se pueden realizar una serie de operaciones dependiendo del dispositivo físico en el cual se encuentre y de la propia estructura interna de los datos en el mismo.

El conjunto de datos que forman parte de los ficheros se agrupan en unidades denominadas registros cuya función es representar a unidades de información del problema que se está tratando.

2.1 Almacenamiento de la información.

Los datos almacenados en la memoria de una computadora pueden mantenerse en ella como máximo, mientras se le suministre alimentación eléctrica. Al ser desconectada de la alimentación, la computadora pierde todos los datos almacenados en su memoria. Debido a ello, a este tipo de almacenamiento se le denomina con el término volátil.

Los dispositivos de almacenamiento de ficheros, por el contrario, mantienen la información grabada en ellos aunque no se suministre alimentación a los mismos, o a la computadora en la que se encuentran instalados. Esto es debido a la naturaleza de estos dispositivos. Por ejemplo, en los dispositivos de grabación magnética la información

permanece aunque no exista suministro eléctrico, ya que la información queda reflejada en una superficie magnetizable mediante cambios en dicha magnetización producidos por una cabeza lectora/grabadora. Esta magnetización, una vez creada, no necesita de alimentación para mantenerse sobre la superficie. La cabeza lectora/grabadora, cuando el dispositivo vuelve a su funcionamiento, es capaz de interpretar esas diferencias en la magnetización de la superficie y leer así los datos almacenados. Por ello, a este tipo de almacenamiento se le denomina con el término no volátil.

Otra terminología hace referencia al almacenamiento en la memoria de la computadora como almacenamiento primario, mientras que otros dispositivos que almacenan ficheros reciben el nombre de almacenamiento secundario.

2.2 El gestor de ficheros.

El gestor de ficheros de un sistema operativo es la parte del mismo que se encarga de administrar todo lo relacionado con los datos en los dispositivos de almacenamiento.

Cuando un programa requiere el acceso a un fichero, es el gestor de ficheros el encargado de asociar a los elementos lógicos del fichero utilizados por el programa las correspondientes localizaciones físicas donde se encuentran los datos.

El sistema operativo utiliza como unidad de transferencia de datos al disco el bloque. Un bloque puede estar formado por uno o varios sectores (físicos o hardware) para así llevar un control, por software, más eficiente de los datos almacenados en el disco. El número de sectores por bloque depende del sistema operativo e incluso, algunos sistemas operativos permiten que éste sea un parámetro que el administrador del sistema puede seleccionar en el proceso de instalación del sistema operativo.

Un bloque, por tanto, es una agrupación de datos que es considerada por el sistema operativo como la unidad de transferencia de información entre los dispositivos de almacenamiento secundario y la memoria de la computadora. En sistemas de ficheros como FAT, NTFS, CDFS, los sectores son agrupados también en bloques que reciben el nombre especial de clusters. Un bloque (o cluster en algunos sistemas) es también el espacio de disco mínimo que puede reservarse para almacenar un fichero. Este concepto hace que un fichero pueda ocupar un tamaño en disco diferente dependiendo del sistema de ficheros utilizado.

El sistema operativo puede acceder directamente a los datos almacenados en un disco, pero esto supondría que las transferencias entre memoria y el disco serían muy frecuentes y haría que el manejo de ficheros tuviera un bajo desempeño debido al coste computacional de una operación de acceso al disco. En lugar de operar de esta forma, y para minimizar así la frecuencia de acceso al disco, el sistema operativo mantiene una zona en la memoria para alojar los bloques del disco de más reciente uso. Esta zona de memoria recibe el nombre de caché de disco.

Cuando un programa requiere leer o escribir datos en el disco el sistema operativo accede en primer lugar a la caché de disco. Si el dato buscado se encuentra, en la caché, el sistema operativo no tiene que acceder al disco. Si el dato buscado no se encuentra en la caché, el sistema operativo accede al disco y mueve a la caché el bloque, consiguiendo así que los datos del bloque requerido se tomen siempre de la caché y que se mantengan en la caché los datos más frecuentemente utilizados. Del mismo modo, cuando el sistema debe reemplazar el bloque que en ese momento se encuentra en la caché, se escribe previamente el bloque de la caché al disco antes de reemplazarlo.

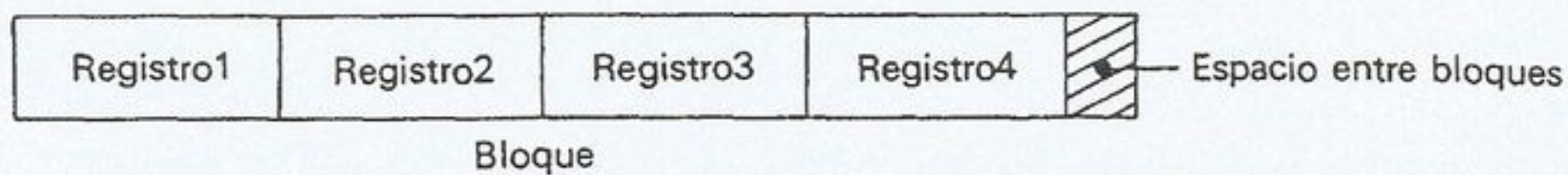
2.3 Rendimiento de la transferencia de la información.

El número de registros lógicos que puede contener un registro físico se denomina factor de bloqueo. Se pueden dar las siguientes situaciones:

- Registro lógico > Registro físico. En un bloque se contienen varios registros físicos por bloque; se denominan registros expandidos.
- Registro lógico = Registro físico. El factor de bloqueo es 1 y se dice que los registros no están bloqueados.

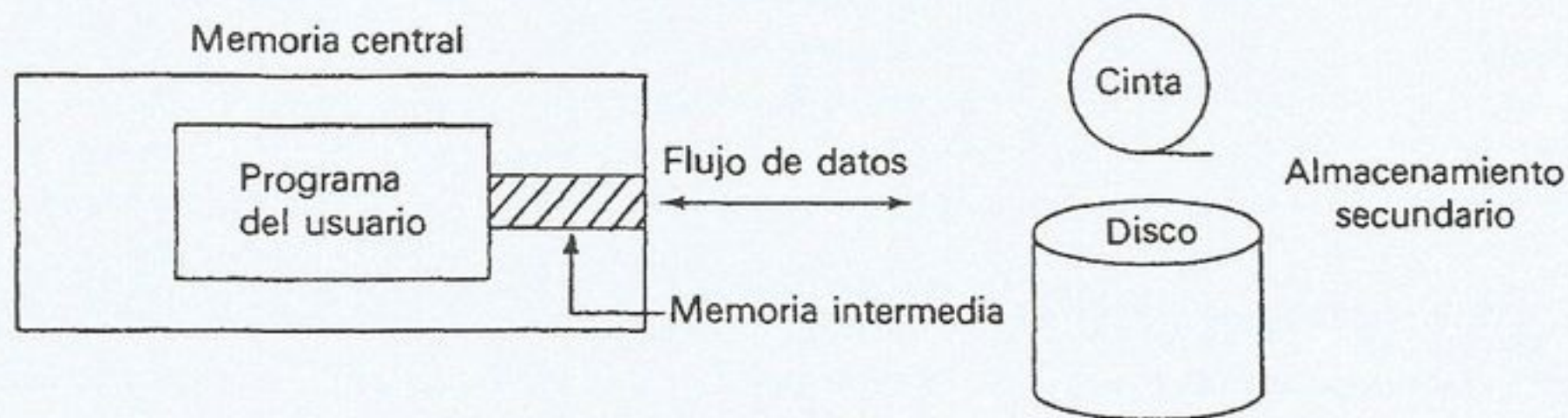


- Registro lógico < Registro físico. El factor de bloqueo es mayor que 1 y los registros están bloqueados.



La importancia del factor de bloqueo se puede apreciar mejor con un ejemplo. Supongamos que se tienen dos ficheros. Uno de ellos tiene un factor de bloqueo de 1 (un registro en cada bloque). El otro fichero tiene un factor de bloqueo de 10 (10 registros/bloque). Si cada fichero contiene un millón de registros, el segundo fichero requerirá 900.000 operaciones de entrada/salida menos para leer todos los registros.

Un factor de bloqueo mayor que 1 siempre mejora el rendimiento; entonces, ¿por qué no incluir todos los registros en un solo bloque? La razón reside en que el aumento del bloque implica el aumento de la memoria intermedia y por consiguiente se reduce el tamaño de la memoria central.



3 Tipos de archivos.

En una aplicación informática se pueden utilizar archivos para realizar funciones diversas. Conocer la función que va a desempeñar un archivo concreto es fundamental a la hora de decidir como se debe organizar éste. Podemos clasificar los archivos como archivos permanentes y archivos temporales.

3.1 Archivos permanentes.

Un archivo permanente contiene información relevante para una aplicación, es decir, los datos necesarios para el funcionamiento de la misma. Su vida es larga (al menos comparable a la de la aplicación para la que ha sido creado) y generalmente no puede generarse de una forma inmediata a partir de otros archivos.

Los archivos permanentes se pueden clasificar en:

- Archivos maestros. Un archivo maestro contiene el estado actual de los datos susceptibles de ser modificados en la aplicación. Es el núcleo central de la aplicación. Todos los procesos están, en general, orientados a actualizar el archivo maestro o a obtener resultados de él. Un ejemplo de este tipo de archivo es el archivo de clientes de un banco, en el que los registros contienen información de identificación de clientes, su saldo en cuenta, etc.
- Archivos constantes. Un archivo constante es aquel que contiene datos fijos para la aplicación. En él las modificaciones son infrecuentes, normalmente se accede a él sólo para consultar. Serán archivos constantes los que contengan los intereses para distintos tipos de cuentas bancarias, la ubicación de estantes en una biblioteca, la capacidad de las aulas de un centro, una tabla de números primos, etc.
- Archivos históricos. Un archivo histórico es aquel que contiene datos que fueron actuales en tiempos anteriores. Se conservan para poder reconstruir la situación actual o situaciones anteriores. En algunos casos puede estar formado simplemente por los registros borrados del archivo maestro. Un archivo histórico puede contener, por ejemplo, los clientes que se han dado de baja en una entidad bancaria.

3.2 Archivos temporales.

Un archivo temporal contiene información relevante para un determinado proceso o programa, pero no para el conjunto de la aplicación. Se genera a partir de los datos de archivos permanentes o para actualizar éstos, y su vida es generalmente muy corta.

Los archivos temporales se pueden clasificar en:

- Archivos intermedios. Se utilizan para almacenar resultados de un programa que han de ser utilizados por otro, dentro de una misma tarea.
- Archivos de maniobras. Se utilizan para almacenar datos propios de un programa que éste no puede conservar en memoria principal, por falta de espacio en ésta. Se encuentran normalmente en programas de cálculo numérico, compiladores y editores. Su vida es siempre menor que el tiempo de ejecución del programa.
- Archivos de resultados. Se utilizan para almacenar datos elaborados que van a ser transferidos a un dispositivo de salida, como por ejemplo un archivo de impresión, que contiene datos que van a ser transferidos a una impresora.

4 Organización de archivos.

La organización de un fichero determina el modo de estructurar los registros pertenecientes a un fichero en almacenamiento secundario. Los métodos de acceso nos permitirán localizar dichos registros. La elección de una determinada organización permitirá realizar distintos tipos de acceso a los registros del fichero y, viceversa, si se desea acceder a un registro o conjunto de registros de una determinada forma, se deberá elegir una organización adecuada para ello.

La organización de los ficheros y los métodos de acceso repercuten fuertemente en los tiempos de respuesta. Los métodos básicos de acceso son los siguientes:

- Acceso secuencial. Para localizar un registro se necesita haber accedido al registro anterior. Puede resultar el modo de acceso más eficiente cuando se tiene que procesar el fichero completo (o un porcentaje del mismo).
- Acceso directo. Se localiza un registro por su dirección, obtenida a partir del valor de una clave de direccionamiento o por la posición relativa que ocupa el registro en el fichero.

La elección del tipo de organización depende de tres factores:

1. Del tipo de proceso: Más concretamente del tipo de acceso requerido (secuencial o directo) y de si las operaciones que se realizan en el fichero son de recuperación o de actualización.
2. De los factores que se desean optimizar: Estos factores son diversos y están muy relacionados entre sí, pero podemos enumerar los siguientes: espacio de almacenamiento, tiempos de respuesta, accesos a disco, etc.
3. De las características del fichero: Es decir, si el fichero tiene gran volumen o no, si su crecimiento será rápido, si se eliminarán muchos registros, etc. En definitiva, tamaño, volatilidad y crecimiento, principalmente.

4.1 Organizaciones consecutivas.

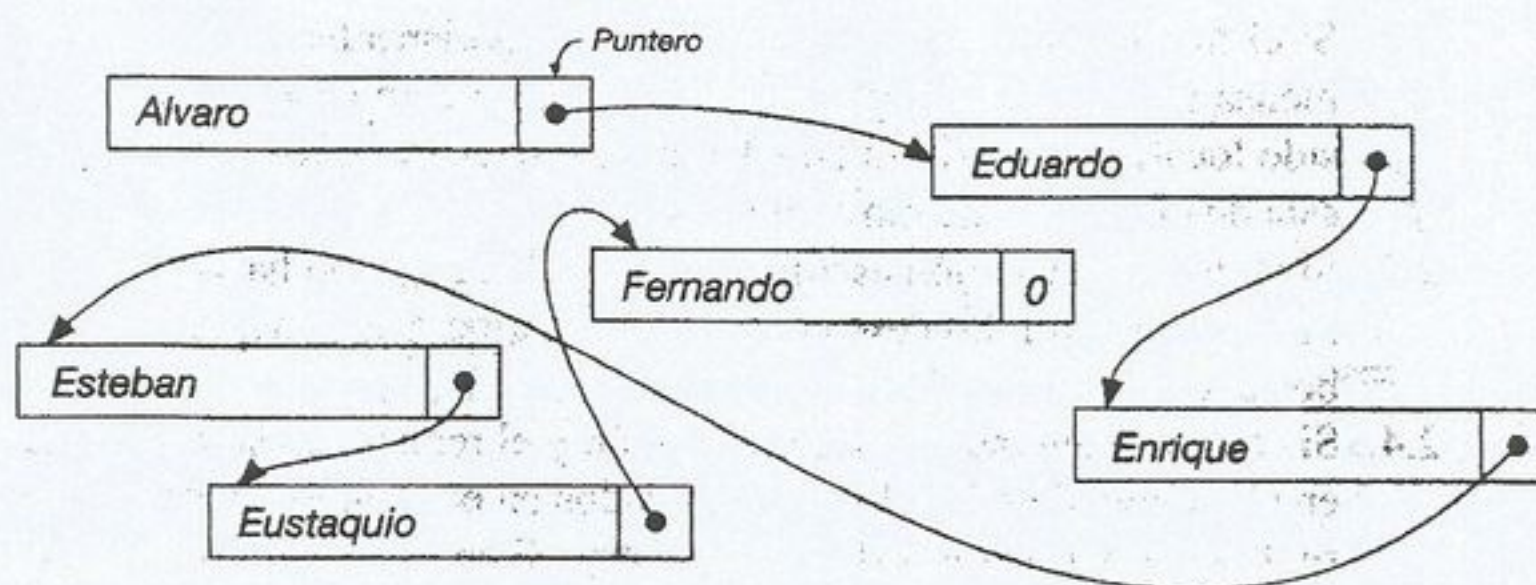
Las organizaciones consecutivas sitúan los registros físicamente uno a continuación de otro. Si la colocación de los registros no tiene un orden lógico, el tipo de organización se denomina consecutiva serial; si, por el contrario, tienen un orden, según una clave de ordenación, se denomina consecutiva secuencial.

El método de acceso que se utiliza para localizar los registros almacenados es el secuencial si la organización es serial; si es secuencial puede utilizarse, además, un acceso directo por posición relativa del registro (búsqueda binaria o dicotómica).

Es adecuada para recuperar ficheros en procesos por lotes, pues optimiza al máximo el espacio de almacenamiento, es muy fácil de programar, puede utilizar registros de longitud variable y heterogéneos, y no está limitada al tipo de soporte directo. Es conveniente utilizar este tipo de organización cuando existe una carga masiva de datos, las tablas son pequeñas, o cuando, en el proceso del fichero se accede normalmente a casi todas las filas.

Esta es la única organización de archivos susceptible de ser gestionada en un dispositivo no direccionable (soportes secuenciales tales como cinta magnética, cinta de papel o tarjeta perforada). La secuencia en que aparecen los registros en el archivo puede estar determinada por el valor de algún campo o ser simplemente temporal.

Si el archivo se encuentra sobre un soporte direccionable (disco por ejemplo) es posible almacenar junto a cada registro un puntero con la dirección del registro siguiente, según el orden lógico del archivo. Estructuralmente un archivo secuencial encadenado es equivalente a una lista lineal de registros.



Si la longitud de los registros es mucho menor que el tamaño del bloque del dispositivo (factor de bloqueo mayor que uno), en cada acceso a disco se podrán leer o escribir varios registros. Si en cada nodo de la cadena hay un solo registro se ralentiza el acceso, ya que la lectura de cada registro implicará un acceso al dispositivo. Para reducir el tiempo de acceso se deben almacenar en cada bloque tantos registros como sea posible, encadenando los bloques, en lugar de encadenar los registros. Los registros contenidos en cada bloque serán consecutivos según el orden lógico del archivo.

4.2 Organizaciones direccionadas.

Este tipo de organización almacena los registros según una relación establecida entre el valor de la clave de direccionamiento y la dirección física del registro, siendo posible realizar un acceso directo a un registro si se conoce la clave de direccionamiento, aunque puede impedir (o al menos dificultar) el acceso secuencial.

La ubicación de un registro se determinará aplicando un algoritmo de transformación a la clave de direccionamiento, lo que dará como resultado la dirección base donde se debería almacenar dicho registro. La dirección base estará comprendida dentro de un rango de valores, que representan las direcciones relativas, denominado espacio de direccionamiento, el cual se fija previamente en función del volumen de datos que hay que almacenar. La organización direccionada resulta muy adecuada en procesos de tipo selectivo donde hay que acceder a un registro por el valor de la clave de direccionamiento o por un rango de la misma.

4.2.1 Métodos de direccionamiento.

Dentro de este tipo de organización se puede distinguir entre organizaciones direccionadas directas y dispersas.

Direccionadas directas.

La función de transformación en esta organización es 1:1, lo que implica que no puede dar direcciones base repetidas: además, la clave de direccionamiento y el algoritmo de transformación deben ser tales que las direcciones obtenidas han de estar comprendidas dentro de un espacio de direccionamiento fijado de antemano. La clave de direccionamiento no podrá tener, por tanto, valores repetidos.

Las ventajas de utilizar este tipo de organización es que los registros siempre están ordenados por la clave de direccionamiento, estando los registros situados en direcciones contiguas, por lo que el acceso puede ser secuencial ordenado, además de directo. Todo registro ocupa su dirección base, lo que permite que, mediante un solo acceso a memoria secundaria, localicemos un registro. El inconveniente que tiene la utilización de este tipo de organización es que las exigencias de la clave de

direccionamiento son muy restrictivas, por lo que suele ser muy difícil encontrar una clave con estas características.

Direccionadas dispersas.

Las organizaciones direccionadas dispersas utilizan una función de transformación $n:1$, lo que significa que para distintas claves de direccionamiento podemos tener la misma dirección base, produciéndose entonces una colisión. Estos registros se denominan sinónimos. Para este tipo de organizaciones debemos tomar las siguientes decisiones:

1. El tamaño del bloque o registro físico en el espacio de direccionamiento. Si el tamaño de bloque es uno el direccionamiento es a registro y si por el contrario es mayor que uno, el direccionamiento es a cubo.
2. Fijar un espacio de direcciones, considerando que no puede haber una densidad de registros mayor del 80%. Donde la densidad se define como n° registros del fichero / espacio de direccionamiento
3. Elegir un algoritmo de transformación que obtenga las direcciones base dentro del espacio de direccionamiento que disperse los registros de forma uniforme, a fin de minimizar los sinónimos (si el direccionamiento es a registro) u optimizar su distribución en el caso de direccionamiento a cubo.
4. Utilizar alguna técnica de gestión de desbordamientos. Un desbordamiento se produce cuando la dirección base de un registro o el cubo al que estaba asignado están ocupados; por tanto, debemos pensar en la nueva ubicación de este registro excedentario. Hay dos formas de resolver el problema de los sinónimos, siempre a costa de complicar la estructura del archivo:
 - Cuando se asocia a una clave una dirección ya ocupada por un registro distinto (esto es, por un sinónimo de esta llave), se busca en el archivo hasta encontrar una posición libre donde escribir el registro.
 - Se reserva una zona de desbordamientos donde se escribirán los registros que no se pueden escribir en la posición que les corresponde según la transformación, a la que denominaremos zona principal.

Si se utiliza el primer procedimiento, la búsqueda de una posición libre puede realizarse secuencialmente a partir de la posición asignada, o aplicando a la dirección obtenida un segundo método de direccionamiento. En cualquier caso, ambos procedimientos son lentos y degradan considerablemente el archivo.

Si se utiliza zona de desbordamientos se puede gestionar ésta, bien secuencialmente o bien encadenada a la zona principal. Esto último presenta el inconveniente de tener que reservar un puntero en cada registro, pero permite un acceso más rápido a los sinónimos, lo que en general será más importante que el espacio.

4.2.2 Algoritmos de direccionamiento.

Hay tres métodos básicos de direccionamiento para los archivos de organización directa:

- Directo. Se utiliza como dirección la propia llave.

- Asociado. A cada llave se le asocia una dirección mediante una tabla.
- Calculado o por transformación de llave. La dirección se obtiene realizando operaciones y transformaciones con la llave.

A continuación se describen brevemente estos tres tipos de direccionamientos para archivos de organización directa.

Direccionamiento directo.

El direccionamiento directo sólo es factible cuando la clave es numérica y su rango de valores no es mayor que el rango de direcciones en el archivo. Se utiliza como dirección la propia clave.

En algunos casos pueden quedar lagunas de direcciones sin utilizar, en lugares conocidos de antemano. En este caso se pueden ocupar dichas direcciones desplazando las direcciones superiores. Un archivo aleatorio con direccionamiento directo está siempre ordenado.

Direccionamiento asociado.

El direccionamiento asociado se puede utilizar para cualquier tipo de clave. Si se utiliza este método debe construirse una tabla en la que figurarán todas las claves y la dirección donde se encuentra el registro correspondiente.

Al añadir nuevos registros las claves se colocan al final de la tabla. Esta se encontrará normalmente desordenada. Por tanto, habrá que localizar las claves en ella por lectura secuencial, lo que ralentiza el acceso, al menos que la tabla esté residiendo en la memoria principal, se indexe o se ordene y se busque en ella la clave con un procedimiento rápido, tal como el de búsqueda dicotómica.

Direccionamiento calculado (hashing).

En el direccionamiento calculado (*hashing*) la dirección de cada registro se obtiene, o bien evaluando una expresión que utiliza como dato la llave, o bien realizando una transformación sobre dicha clave. Se utiliza cuando la llave no es numérica o, siendo numérica, cuando puede tomar valores dentro de un conjunto demasiado grande como para poder ser utilizada directamente como dirección. Cualquiera que sea la transformación usada es normal que aparezcan sinónimos, si bien la frecuencia con que éstos aparecerán dependerá en cada caso concreto del algoritmo de transformación utilizado.

Si la clave no es numérica se utilizará una conversión previa para obtener un número a partir de ella.

Existen gran cantidad de métodos de cálculo de dirección. Aquí sólo citaremos los más relevantes.

- División. La dirección es el resto de dividir la llave por una constante. Normalmente esta constante es el mayor de los números primos menores que el número de posiciones del archivo.
- Extracción. Consiste en utilizar como dirección un grupo de cifras de la llave, contiguas o no.

- Elevación al cuadrado. Se utiliza cuando la representación numérica de la llave no es muy larga. Consiste en elevar ésta al cuadrado y tomar los dígitos centrales.
- Plegamiento. Se utiliza para llaves muy largas. Se descompone la llave en trozos de cifras contiguas del mismo tamaño y se suman éstos. Sobre el número generado se puede aplicar otro método si es necesario.

Siempre que deba accederse a una posición, ya sea para introducir por primera vez un registro o para su consulta o modificación, debe aplicarse a la llave el algoritmo de transformación.

4.3 Organizaciones indexadas.

Sobre las organizaciones básicas (consecutivas y directas) es posible superponer nuevas estructuras que consiguen mejorar la eficiencia en el acceso a los registros que cumplan ciertas condiciones de búsqueda; se trata de las organizaciones indexadas. Puede ocurrir también que la estructura indexada esté unida a los datos, de forma que éstos no tengan una organización básica previa a la construcción del índice.

Una estructura de índice se define sobre uno o más campos de un fichero, llamados clave de indexación; de forma que cada uno de los valores de la clave de indexación se asocia a la dirección del registro al que le corresponde dicho valor.

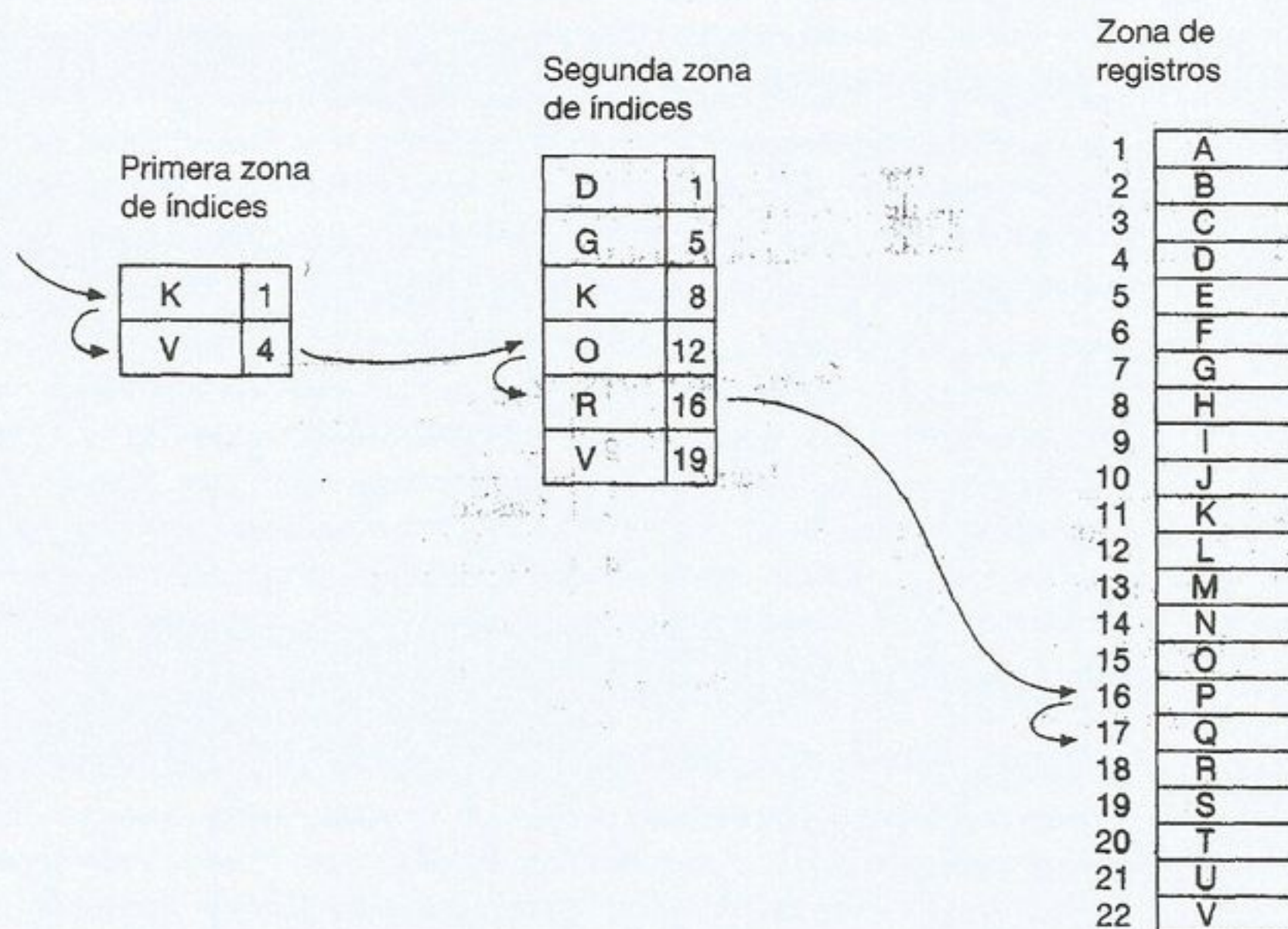


Un valor de la clave de direccionamiento con la dirección asociada constituye una entrada al índice. El acceso al registro de datos se hace por el valor de la clave de indexación, siendo dicha clave el criterio de búsqueda en consultas selectivas. La clave de indexación puede ser un identificador del fichero de datos (índice primario), o un conjunto de campos no identificadores (índice secundario).

Sobre un mismo fichero de datos se pueden definir varios índices, cada uno sobre una clave de indexación. Cualquier búsqueda selectiva de un registro en una organización indexada obliga a una consulta previa al índice (la cual proporciona la dirección del registro) y un posterior acceso directo al fichero de datos por medio de la dirección obtenida. Por tanto, para optimizar la eficiencia en los accesos, será conveniente mantener el índice, siempre que sea posible, en memoria principal.

Este tipo de organización admite que no exista una entrada al índice por cada registro del fichero (índice no denso), en cuyo caso el fichero de datos tiene que estar ordenado por la clave de direccionamiento. Si existe una entrada por cada registro

(índice denso) el fichero no necesita estar ordenado. Los índices no densos admiten la creación de índices sobre los índices, teniendo así estructuras indexadas a varios niveles (índices multinivel).



De entre todas las organizaciones basadas en índices destacan las secuencias indexadas (ISAM), que añaden a las ventajas de las secuenciales (rapidez en el acceso secuencial al fichero), la posibilidad de acceder directamente a ciertos registros (acceso directo) apoyándose en el índice. En este tipo de organizaciones se penalizan las actuaciones respecto a las de tipo secuencial, por la necesidad de actualizar el índice; además, se deterioran cuando se producen muchas actualizaciones.

Las técnicas de indexación difieren a causa del tipo de índice (primario/secundario, denso/no denso), la estructura del índice (ordenado/desordenado, mononivel/multinivel), la organización del fichero de datos (serial/secuencial/disperso), y/o la forma de tratar los desbordamientos

Los índices multinivel, pueden no tener un número fijo de niveles a fin de evitar desbordamientos y/o reorganizaciones del índice, así como de los datos, cuando se insertan registros; en este caso se fija previamente el número de entradas por cada nivel, y se va incrementando el número de niveles, cuando las inserciones obligan a ello. Surgen así distintos tipos de índices (binarios, equilibrados, etc.), basados, en general, en estructuras de tipo árbol. Entre todas las estructuras en árbol destacan la familia de árboles B.

Los nodos del árbol son bloques del espacio de almacenamiento que contienen los valores de las claves de búsqueda y punteros a los nodos hijos; además han de contener, bien los datos, bien los punteros a los mismos. Se asegura una determinada ocupación de los bloques no inferior al 50%, que se estabiliza en un promedio del 69%.

El árbol B es un árbol especial de búsqueda con restricciones adicionales, que garantizan que siempre estará equilibrado y que su densidad es siempre superior a un determinado valor (el espacio de almacenamiento que se "desperdicia" nunca será

excesivo). Existen diversas variedades de árboles B (B^+ , B^* , con la información asociada o separada de la estructura del árbol, etc.), siendo estas estructuras el fundamento de las organizaciones físicas en los sistemas relacionales. Los árboles B^+ soportan, al igual que las estructuras ISAM, tanto acceso secuencial como directo, con la ventaja adicional de que crecen dinámicamente cuando lo hace la tabla, manteniendo además el orden de la clave de acceso.

La utilización de índices mejora los tiempos de respuesta ante consultas que impliquen a los atributos indexados, pero disminuye el rendimiento de la base de datos, ya que se debe actualizar el índice cuando se actualizan los atributos sobre los que está definido, además de aumentar el espacio de almacenamiento.

Por estas razones suele ser conveniente indexar la clave primaria (mediante un índice único) en el caso de que el producto no lo haga, las claves alternativas que se utilicen frecuentemente (también mediante un índice único), y aquellas claves ajenas que se utilicen en combinaciones con otras tablas. Sin embargo, en tablas pequeñas, o en aquellas en que prácticamente se recuperan todas las filas, no suele ser conveniente, ya que es mejor una búsqueda secuencial. También se deberá tener en cuenta a la hora de indexar el tipo de datos de los atributos afectados, ya que no es conveniente indexar datos de tipo carácter muy largos.

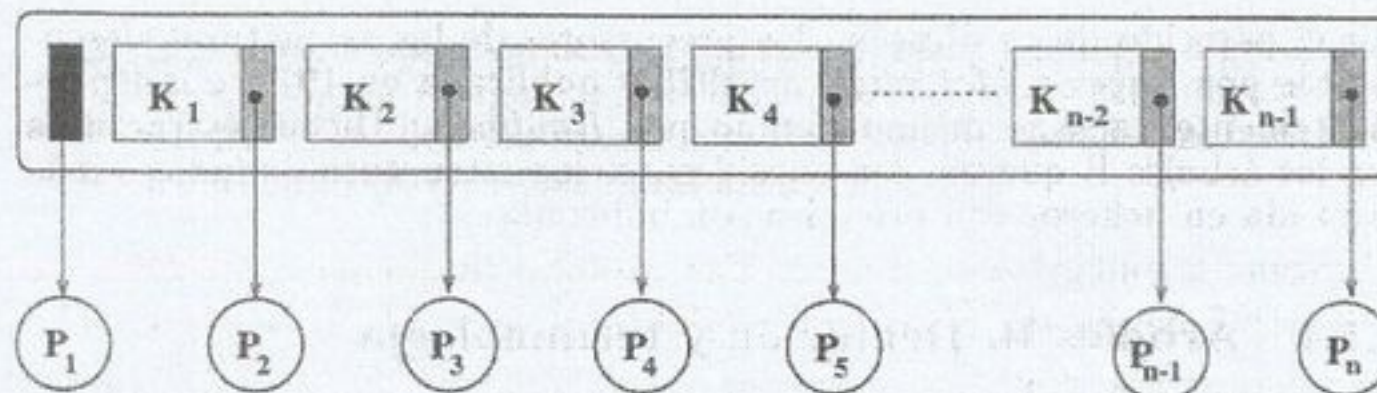
4.3.1 Archivos de índices de árbol B^+

La estructura de índice de árbol B^+ es la más extendida de las estructuras de índices que mantienen su eficiencia a pesar de la inserción y borrado de datos. Un índice de árbol B^+ toma la forma de un árbol equilibrado donde los caminos de la raíz a cada hoja del árbol son de la misma longitud. Cada nodo que no es una hoja tiene entre $\lceil n/2 \rceil$ y n hijos, donde n está fijo para cada árbol en particular.

La estructura de árbol B^+ implica una degradación del rendimiento al insertar y al borrar, además de un espacio extra. Este tiempo adicional es aceptable incluso en archivos con altas frecuencias de modificación, ya que se evita el coste de reorganizar el archivo. Además, puesto que los nodos podrían estar a lo sumo medio llenos (si tienen el mínimo número de hijos), hay algo de espacio desperdiciado. Este gasto de espacio adicional también es aceptable, dados los beneficios en el rendimiento aportados por las estructura de árbol B^+ .

Estructura de árbol B^+

Un índice de árbol B^+ es un índice multinivel. Puede contener hasta $n-1$ claves de búsqueda K_1, K_2, \dots, K_{n-1} , y n punteros P_1, P_2, \dots, P_n . Los valores de la clave de búsqueda de un nodo se mantienen ordenados; así, si $i < j$, entonces $K_i < K_j$.



Consideraremos primero la estructura de los nodos hoja. Para $i = 1, 2, \dots, n-1$, el puntero P_i apunta o bien a un registro del archivo con valor de la clave de búsqueda K_i ,

o bien a un cajón de punteros, cada uno de los cuales apunta a un registro del archivo con valor de la clave de búsqueda K_i . La estructura cajón se usa solamente si la clave de búsqueda no forma una clave primaria y si el archivo no está ordenado según la clave de búsqueda. El puntero P_n tiene un propósito especial que discutiremos más adelante.

Ahora que hemos visto la estructura de un nodo hoja, consideraremos cómo los valores de la clave de búsqueda se asignan a nodos concretos. Cada hoja puede guardar hasta $n-1$ valores. Está permitido que los nodos hoja contengan al menos $\lceil (n-1)/2 \rceil$ valores. Los rangos de los valores en cada hoja no se solapan. Así, si L_i y L_j son nodos hoja e $i < j$, entonces cada valor de la clave de búsqueda en L_i es menor que cada valor de la clave en L_j . Si el índice de árbol B^+ es un índice denso, cada valor de la clave de búsqueda debe aparecer en algún nodo hoja.

Ahora se puede explicar el uso del puntero P_n . Dado que existe un orden lineal en las hojas basado en los valores de la clave de búsqueda que contienen, se usa P_n para encadenar juntos los nodos hoja en el orden de la clave de búsqueda. Esta ordenación permite un procesamiento secuencial eficaz del archivo.

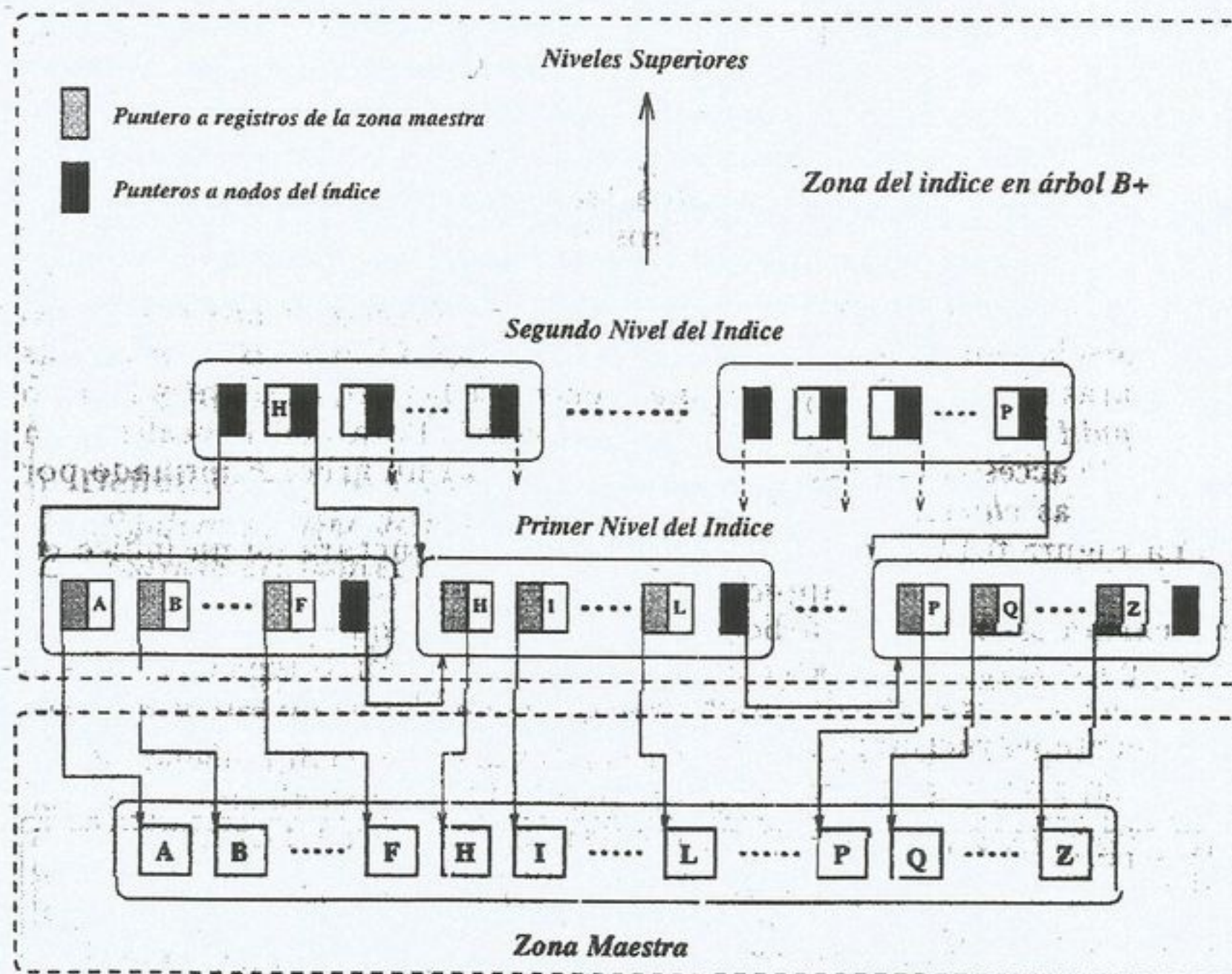
Los nodos internos del árbol B^+ forman un índice multinivel (disperso) sobre los nodos hoja. La estructura de los nodos internos es la misma que la de los nodos hoja, excepto que todos los punteros son punteros a nodos del árbol. Un nodo interno podría guardar hasta n punteros y debe guardar al menos $\lceil n/2 \rceil$ punteros. El número de punteros de un nodo se llama grado de salida del nodo.

Consideremos un nodo que contiene m punteros. Para $i = 2, 3, m-1$, el puntero P_i apunta al subárbol que contiene los valores de la clave de búsqueda menores que K_i y mayor o igual que K_{i-1} . El puntero P_m apunta a la parte del subárbol que contiene los valores mayores o iguales que K_{m-1} , y el puntero P_1 , apunta a la parte del subárbol que contiene los valores de la clave menores que K_1 .

El requisito de que cada nodo interno mantenga al menos $\lceil n/2 \rceil$ punteros se impone en todos los niveles del árbol, excepto en la raíz.

Organización de archivos con árboles B^+

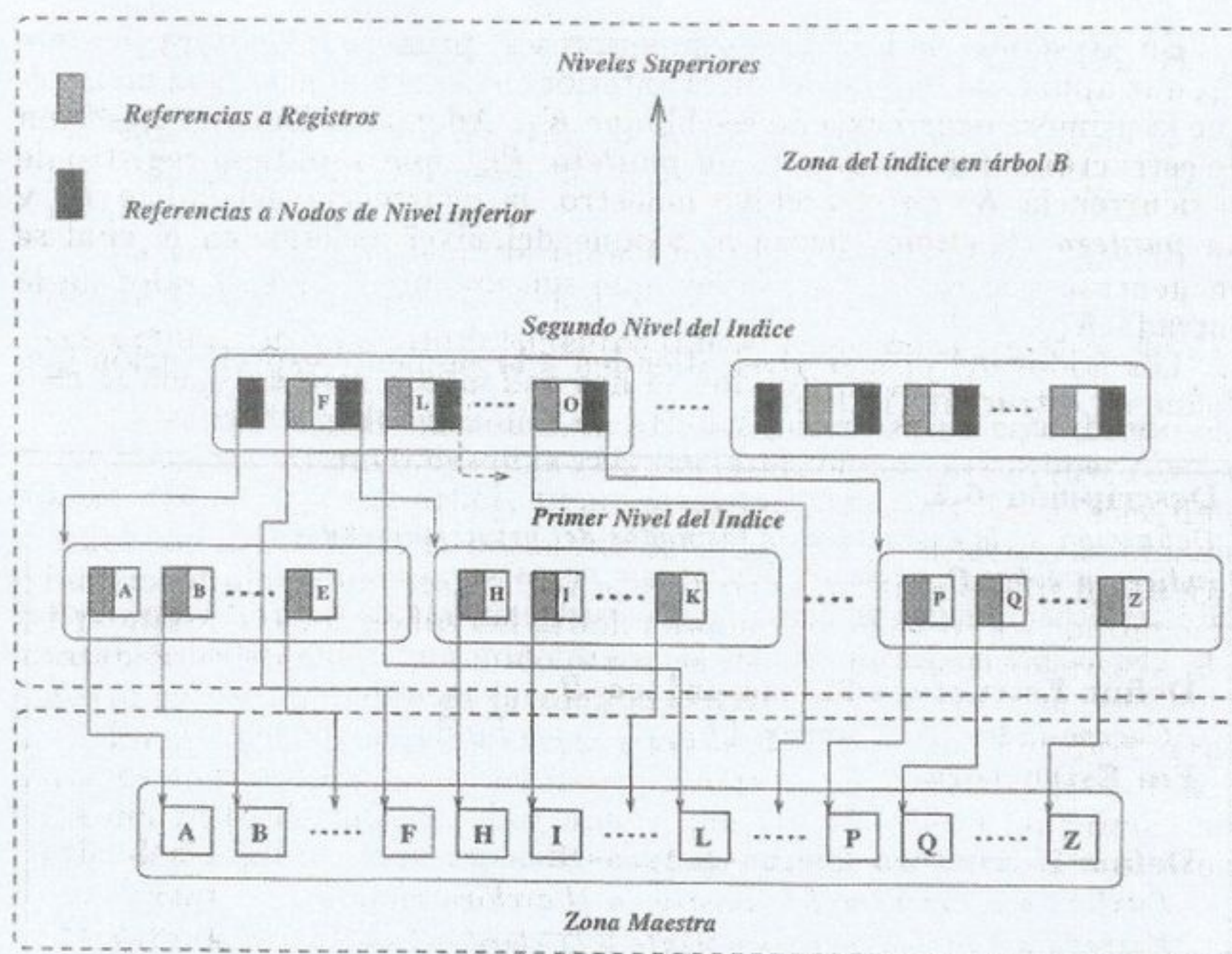
En estas estructuras, la estructura del árbol B^+ se usa no sólo como un índice, sino también como un organizador de los registros dentro del archivo. En la organización de archivo con árboles B^+ , los nodos hoja del árbol almacenan registros, en lugar de almacenar punteros a registros. Ya que los registros son normalmente más grandes que los punteros, el número máximo de registros que se pueden almacenar en un nodo hoja es menor que el número de punteros en un nodo interno. Sin embargo, todavía se requiere que los nodos hoja estén llenos al menos hasta la mitad.



4.3.2 Archivos de índices de árbol B

Los índices de árbol B son similares a los índices de árbol B⁺. La diferencia principal entre las dos aproximaciones es que un árbol B elimina el almacenamiento redundante de los valores de la clave búsqueda puesto que un árbol B permite que los valores de la clave de búsqueda aparezcan solamente una vez.

Ya que las claves de búsqueda no están repetidas en el árbol B, sería posible almacenar el índice empleando menos nodos del árbol que con el correspondiente índice de árbol B⁺. Sin embargo, puesto que las claves de búsqueda que aparecen en los nodos internos no aparecen en ninguna otra parte del árbol B, nos vemos obligados a incluir un campo adicional para un puntero por cada clave de búsqueda de un nodo interno. Estos punteros adicionales apuntan a registros del archivo o a los cajones de la clave de búsqueda asociada.



5 Conclusiones.

Los datos que se encuentran en memoria masiva suelen organizarse en archivos. Un fichero es un conjunto de información sobre un mismo tema, tratada como una unidad de almacenamiento y organizada de forma estructurada para la búsqueda de un dato individual. Un fichero está compuesto de registros homogéneos que contienen información sobre el tema.

El sistema operativo utiliza como unidad de transferencia de datos al disco el bloque. Un bloque es el espacio de disco mínimo que puede reservarse para almacenar un fichero. Para minimizar así la frecuencia de acceso al disco, el sistema operativo mantiene una caché en memoria para alojar los bloques del disco de más reciente uso.

El número de registros lógicos que puede contener un registro físico se denomina factor de bloqueo. Este valor es importante porque influye en el rendimiento del sistema.

Podemos clasificar los archivos como archivos permanentes y archivos temporales. Los archivos permanentes se pueden clasificar en archivos maestros, constantes, o históricos. Los archivos temporales se pueden clasificar en intermedios, de maniobras, o de resultados.

La organización de un fichero determina el modo de estructurar los registros pertenecientes a un fichero en almacenamiento secundario.

Las organizaciones consecutivas sitúan los registros físicamente uno a continuación de otro. Esta es la única organización de archivos susceptible de ser gestionada en un dispositivo no direccionable. La secuencia en que aparecen los registros en el archivo puede estar determinada por el valor de algún campo o ser

simplemente temporal. Si el archivo se encuentra sobre un soporte direccionable es posible almacenar junto a cada registro un puntero con la dirección del registro siguiente.

Las organizaciones direccionadas almacenan los registros según una relación establecida entre el valor de la clave de direccionamiento y la dirección física del registro, siendo posible realizar un acceso directo a un registro si se conoce la clave de direccionamiento. Podemos distinguir entre organizaciones direccionadas directas y dispersas.

Sobre las organizaciones consecutivas y directas es posible superponer estructuras que mejoran la eficiencia en el acceso a los registros que cumplan ciertas condiciones de búsqueda; se trata de las organizaciones indexadas. Una índice se define sobre uno o más campos de un fichero, de forma que cada uno de los valores de la clave de indexación se asocia a la dirección del registro al que le corresponde dicho valor.

Este tipo de organización admite que no exista una entrada al índice por cada registro del fichero (índice no denso), o que exista una entrada por cada registro (índice denso). Los índices no densos admiten la creación de índices sobre los índices, teniendo así estructuras indexadas a varios niveles (índices multinivel).