

TEMA 14

UTILIZACIÓN DE FICHEROS SEGÚN SU ORGANIZACIÓN

ÍNDICE

1. INTRODUCCIÓN
2. PARÁMETROS DE UTILIZACIÓN DE FICHEROS
3. OPERACIONES REALIZABLES CON FICHEROS
4. FICHEROS CON ORGANIZACIÓN SECUENCIAL
5. FICHEROS CON ORGANIZACIÓN SECUENCIAL ENCADENADA
6. FICHEROS CON ORGANIZACIÓN SECUENCIAL INDEXADA
7. FICHEROS CON ORGANIZACIÓN SECUENCIAL INDEXADA ENCADENADA
8. FICHEROS CON ORGANIZACIÓN DIRECTA
9. BIBLIOGRAFÍA

1. INTRODUCCIÓN

Un **fichero** (también denominado **archivo**) es un conjunto ordenado de datos que tienen entre sí una relación lógica y están almacenados en un soporte de información adecuado para la comunicación con el ordenador. En un fichero se almacena información referente a un mismo tema de una forma estructurada con el fin de manipular los datos de manera individual. Un fichero está compuesto por estructuras de datos más simples denominadas **registros**. Todos los registros de un fichero son del mismo tipo, es decir, un fichero está formado por un conjunto de registros homogéneos. Cada registro está formado por **campos** que contienen información referente a un elemento o característica en particular dentro del fichero.

Se denomina **registro lógico** al conjunto de información identificable acerca de uno de los elementos a que hace referencia el fichero. Se llama **registro físico** o **bloque** al conjunto de información que, según las posibilidades de cada máquina, puede ser escrito o leído de una sola vez. Los registros físicos están almacenados en el dispositivo o soporte de información, siendo el sistema operativo el encargado de escribir y leer los datos que componen el fichero. El sistema operativo transporta, cada vez que accede al dispositivo (para leer o escribir), una cantidad fija de información (bloque o registro físico) que depende de las características hardware o físicas de éste.

La **dirección lógica** de un registro es la posición relativa que ocupa en el fichero, mientras que la **dirección física** es la posición real o efectiva donde se encuentra dicho registro en el soporte de información. En el fichero, los registros aparecen al usuario en secuencia lógica, es decir, ordenados por su dirección lógica. No obstante, el orden físico de los registros de un fichero en el disco puede no tener ninguna relación con la información que contiene.

Los ficheros se guardan en dispositivos de memoria masiva, estando limitado su tamaño por el de los dispositivos que los albergan. Los dispositivos o soportes de memoria auxiliar pueden ser de dos tipos: **secuencial** o **no direccionables**, y de **acceso directo** o **direccionables**. En los soportes no direccionables, si se quiere acceder al registro n hay que leer los $n-1$ registros anteriores. En los soportes direccionables, por el contrario, se puede acceder directamente a un registro físico sin más que dar su dirección física, sin necesidad de recorrer otros registros.

Dentro de un determinado fichero, los registros van a ser identificados por un campo o conjunto de campos, al que se le denomina **llave** o **identificativo**. Este identificativo sirve para discriminar cada registro de los demás, así como para la localización rápida de los registros dentro de ficheros con determinadas organizaciones. Un fichero puede tener una, varias o ninguna llave.

Independientemente de la organización con la que se diseñe el fichero, sobre sus registros se pueden realizar diferentes operaciones. De ellos, se extrae la información para que pueda ser procesada. A esta acción se le denomina **acceso**.

Básicamente, se puede decir que el acceso a ficheros, al igual que su organización, se puede realizar de dos modos: Secuencial y directo.

El **acceso secuencial** permite un acceso rápido a los registros situados en posiciones físicamente contiguas dentro del fichero. Generalmente, este tipo de acceso se utiliza cuando queremos consultar todos los registros de un fichero, o un número bastante grande de ellos.

Cuando utilizamos el **acceso directo**, hay que considerar que los registros no tienen que estar grabados siguiendo ninguna secuencia de clasificación, sino que solamente se sitúan en determinadas posiciones físicas del fichero. Estas posiciones se obtienen directamente del identificativo o de una transformación del mismo. Este será el tipo de acceso deseable cuando queremos realizar consultas esporádicas a los registros del fichero, o cuando el número total de registros a consultar sea bajo.

2. PARÁMETROS DE UTILIZACIÓN DE FICHEROS

Al crear un fichero es necesario especificar qué organización tendrá. La elección de una determinada organización implica tener en cuenta el tipo de operaciones a que van a ser sometidos los ficheros. Con objeto de determinar la forma de organización más adecuada y eficiente, es preciso estudiar una serie de características de utilización. Estos parámetros de utilización son los siguientes:

Volumen. Es el número de caracteres que ocupa (o lo que puede llegar a ocupar en situaciones límite) el fichero en el soporte de almacenamiento. Antes de la creación del fichero, se puede evaluar el espacio que ocupará. Para ello, se hace una estimación del número de registros que contendrá el fichero y se multiplica por la longitud media de los registros. Si se estima que el sistema trabajará con ficheros de gran volumen, debe elegirse una organización que aproveche bien el espacio.

Crecimiento. Especifica el aumento o disminución del tamaño del fichero. Utiliza como parámetro la tasa de crecimiento, que es el número de altas y bajas que se procesan en cada tratamiento del fichero.

Actividad. Se refiere al número de procesos de consulta y modificación que sufrirá el fichero a lo largo de su vida. Si se estima que el fichero tendrá una gran actividad, se elegirá una organización que permita rapidez en las búsquedas. Para medir la actividad de un fichero se utilizan dos parámetros:

- *Tasa de consulta o modificación.* Es el número de registros tratados (consultados o modificados) en cada procesamiento del fichero.
- *Frecuencia de consulta o modificación.* Es el número de veces que se accede al fichero para consultar o modificar por unidad de tiempo (diario, mensual, anual, etc.)

Volatilidad. Se refiere al número de procesos de inserción y borrado en el tratamiento del fichero. Si se estima que el fichero tendrá una volatilidad grande, habrá que elegir una organización que sea flexible. Se utilizan dos parámetros para medir la volatilidad:

- *Tasa de renovación.* Es el número de registro insertados o borrados en el tratamiento del fichero.
- *Frecuencia de renovación.* Es el número de veces que se accede al fichero para insertar o borrar por unidad de tiempo (diario, mensual, anual, etc.).

3. OPERACIONES REALIZABLES CON FICHEROS

El usuario tiene que acceder a los ficheros y realizar un conjunto de operaciones sobre ellos. Los tipos de operaciones más habituales sobre el conjunto de registros que componen un fichero son las siguientes:

Creación. Operación previa a cualquier otra, que supone una descripción de las características de los datos (diseño del fichero). La vida del fichero comienza con la creación del mismo.

Consulta o recuperación. Esta operación se realiza a nivel de registro para conocer la información contenida en el fichero

Mantenimiento o actualización. Una vez que el fichero está creado, puede surgir la necesidad de modificar la información que contiene. Esto supone tres tipos de operaciones distintas a nivel de registro:

- *Inserción* de un registro cuando aparezcan nuevas entidades (admisión de un nuevo alumno en el Instituto).
- *Modificación* de un registro para el que se han producido cambios. Consiste en el cambio de uno o varios campos del registro (modificar la dirección de un alumno que ha cambiado de vivienda).
- *Eliminación o borrado* de un registro porque hayan desaparecido las correspondientes entidades (alumnos que se han dado de baja por cambio de Instituto o por haber terminado sus estudios). La supresión de un registro se puede realizar de dos modos:
 - a) Eliminación por marca o borrado lógico, que consiste en la modificación del valor de un campo mediante el que los programas de aplicación detectan que el contenido del registro no tiene validez.

- b) Eliminación real, que consiste en hacer que ese registro sea inaccesible, o bien ocupar su espacio con otros registros de ese fichero.

Borrado o destrucción. Elimina la información del fichero, así como su estructura. La destrucción del fichero supone el fin de su existencia.

La mayor parte de las operaciones de recuperación y actualización implican realizar una localización o búsqueda de un registro concreto dentro del fichero, para luego actuar sobre él (lectura, escritura, modificación o borrado).

Otras operaciones menos comunes sobre ficheros son su ordenación según algún criterio, la fusión (unir dos o más ficheros para que queden integrados en uno sólo), la división (operación contraria a la fusión), la clasificación, impresión, etc.

Los sistemas informáticos incluyen en su software propio una serie de programas de utilidad para efectuar operaciones básicas con ficheros. Dichos programas se utilizan mediante un lenguaje de control. Para explotar mejor las características de los ficheros, existen paquetes de programas específicos, denominados genéricamente **sistemas de gestión de ficheros**, que permiten diseñar ficheros con determinadas estructuras y realizar recuperaciones y actualizaciones de una forma cómoda y eficaz.

4. FICHEROS CON ORGANIZACIÓN SECUENCIAL

En este tipo de organización, los registros se encuentran almacenados de forma contigua, siguiendo la secuencia lógica del fichero. Todas las operaciones que se realizan sobre el fichero se hacen según esta secuencia. Esta es la única organización de ficheros susceptible de ser gestionada en un dispositivo no direccionable (soportes secuenciales, tales como una cinta magnética), lo que no es óbice para que la organización secuencial sea utilizada también en soportes de acceso directo. Los registros se encuentran ordenados según un indicativo que se toma como base de clasificación. Este indicativo está contenido en alguno de los campos del registro.

Las distintas *operaciones* que se pueden realizar sobre ficheros con esta organización son:

Añadir. Sólo es posible escribir al final del fichero. Los registros se van almacenando uno a continuación del otro, en el orden en que se desea que estén en el fichero.

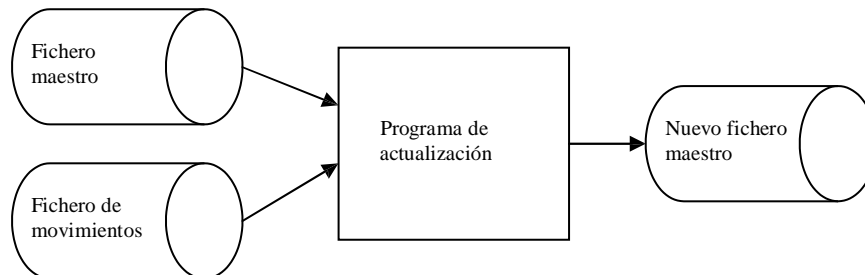
Consulta o recuperación. Se realiza en orden secuencial, es decir, para leer el registro que ocupa la posición n es necesario leer previamente los $n-1$ anteriores. Si se parte de un fichero de movimientos, en el que estén las operaciones a realizar sobre el fichero, en primer lugar se leerá un registro de dicho fichero de movimientos para saber cuál es el identificativo del registro que queremos localizar. Después se lee un registro del fichero maestro para comparar. Entonces:

- a) **Si el identificativo del fichero maestro es mayor que el identificativo del fichero de movimientos**, se produce un error en la localización, ya que, se intenta localizar un registro que no existe en el fichero maestro. En este caso, sólo es necesario volver a leer otro registro del fichero de movimientos y proceder de nuevo a su comparación.
- b) **Si el identificativo del fichero maestro es menor que el identificativo del fichero de movimientos**, indica que aún no se ha localizado el registro deseado, por lo que se leerá un nuevo registro del fichero maestro para volver a comparar.
- c) **Si ambos identificativos son iguales**, se ha encontrado en el fichero el registro deseado. A continuación se leerá un nuevo registro del fichero de movimientos y del fichero maestro para volver a comparar.

El final del proceso se produce cuando se termine cualquiera de los dos ficheros.

Operaciones de actualización: inserción, modificación y eliminación. Estas operaciones no son fáciles de realizar sobre un fichero secuencial. La actualización de un fichero secuencial obliga a escribirlo de nuevo totalmente, es decir, cada operación de actualización implica crear de nuevo el fichero. La operación se realiza mediante un programa que

utilizará como entrada la versión a modificar del fichero y un fichero intermedio denominado **fichero de movimientos**, y devolverá como salida el nuevo fichero, que incluye las modificaciones. El fichero de movimientos también es secuencial y almacenará las modificaciones a realizar sobre el fichero. Los registros del fichero de movimientos tienen la misma estructura que los del fichero a actualizar, más un campo, que normalmente se coloca al principio del registro, indicando el código de la operación a realizar con el registro: modificar, eliminar o insertar.



- **Insertión.** No se pueden insertar registros en cualquier posición del fichero. La única solución es añadirlos al final de los que ya existen. Esto implica que el fichero quedará completamente desordenado y será necesario reordenarlo. Además, si el soporte utilizado para almacenar el fichero es de tipo secuencial, el proceso de inserción de registro se hace prácticamente imposible.

Para subsanar estos problemas, podemos utilizar una técnica de inserción de registros a modo de intercalación. Este proceso se realiza partiendo de un fichero maestro y de un fichero de movimientos, en el que tenemos los registros que queremos insertar, y da lugar a un fichero nuevo, al que irán a parar los resultados de la operación.

Se comienza leyendo un registro del fichero de movimientos e inmediatamente después otro del fichero maestro. El proceso continúa con la comparación de sus identificativos, pudiendo suceder:

- Identificativo del fichero maestro mayor que el identificativo del fichero de movimientos.** Estamos dando de alta a un registro situado en el fichero de movimientos. Se graba un registro en el fichero nuevo, con los datos del registro del fichero de movimientos. La siguiente operación a realizar es la lectura de un nuevo registro del fichero de movimientos.
 - Identificativo del fichero maestro menor que el identificativo del fichero de movimientos.** No ocurre nada de excepción. Lo único que se hace es grabar un registro en el fichero nuevo, con los datos del registro leído del fichero maestro. A continuación se lee un nuevo registro del fichero maestro y se vuelven a comparar los identificativos.
 - Ambos identificativos son iguales.** Al estar en un proceso de inserción, no podemos tener dos registros con el mismo identificativo (el registro a insertar ya existe), por lo que se considera un error. En este caso, grabaremos el registro del fichero maestro en el fichero nuevo. A continuación leeremos un nuevo registro de cada fichero y volvemos a comparar.
- **Eliminación.** Necesitaremos, como en la inserción, utilizar un fichero auxiliar. Éste fichero nuevo se utilizará para copiar todos los registros del fichero maestro excepto aquéllos que queramos eliminar. También se necesitará el fichero de movimientos donde estén contenidos los identificativos de los registros que queremos eliminar del fichero maestro.

- **Modificación.** El proceso consiste, básicamente, en localizar los registros que se quieren modificar. Una vez localizados, se graban con sus modificaciones en el fichero nuevo. El resto de los registros del fichero que no se modifican, se grabarán tal y como están. Una vez terminado el proceso, tendremos que recuperar la información del fichero nuevo sobre el fichero maestro.

Si se utilizan dispositivos direccionables con la organización secuencial, es posible realizar algunas actualizaciones sobre el fichero sin necesidad de crear otro fichero maestro:

Consulta. Únicamente en el caso de que los registros sean de longitud fija, se puede determinar la dirección de comienzo de un registro a partir de su posición relativa en el fichero. Esto es, si los registros tienen una longitud de k caracteres cada uno, el registro número n comenzará en la posición $k*(n-1)$ y finalizará en la posición $k*n$. Por tanto, podemos utilizar el acceso directo para localizar un registro, con lo que obtendremos mayor rapidez en las consultas.

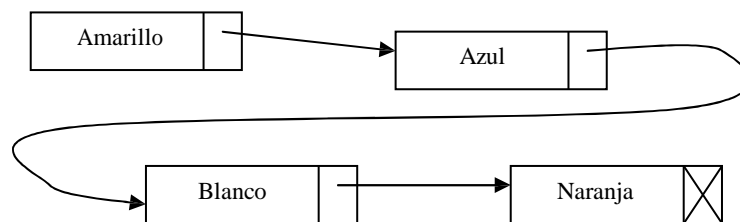
Modificación. Una vez localizado un registro, se puede reescribir en él dentro del propio fichero, siempre que la modificación no suponga un aumento de la longitud del registro.

Borrado. No es posible realizar un borrado real de un registro del fichero. Sin embargo, podemos hacer lo que se denomina **borrado lógico**, que consiste en marcar el registro de forma que al leerlo se identifique como no válido.

La organización secuencial se suele utilizar con ficheros en los que en cada proceso se debe acceder a la mayor parte de los registros del mismo. Presenta la ventaja de aprovechar bien el espacio, es sencilla de utilizar y se puede emplear en dispositivos secuenciales (son los más baratos). Su principal inconveniente es la falta de flexibilidad y la velocidad de acceso, por lo que esta organización no es adecuada para ficheros utilizados por procesos en los que se establecen intercambios de información en tiempo real.

5. FICHEROS CON ORGANIZACIÓN SECUENCIAL ENCADENADA

Los registros que componen un fichero con organización secuencial encadenada almacenan, además de su propia información, un puntero (tipo de dato que almacena una dirección de memoria) con la dirección del registro siguiente, según el orden lógico del fichero. Desde el punto de vista lógico, el fichero está ordenado según el valor de alguna llave. Los registros se encuentran colocados en direcciones físicas totalmente arbitrarias, pero los punteros permiten asegurar la secuencia lógica del fichero.



Con esta organización, se pueden realizar las siguientes operaciones:

Añadir. Para añadir un registro al final del fichero, se localiza la posición del último registro del fichero. El puntero de este último registro contendrá una dirección nula. Una vez localizado el final del fichero, el nuevo registro se escribirá en una zona libre, colocando en su campo de puntero una dirección nula. Finalmente, se modifica el último registro para actualizar el valor de su puntero, de forma que contenga la dirección del nuevo registro.

Recuperación o consulta. La consulta es secuencial. Para leer un determinado registro, se accede al primero en la lista y se comprueba si es el registro buscado. Si no es así, se lee la

dirección del siguiente (gracias al puntero) y se accede a dicha dirección. El proceso continúa hasta que se localiza el registro deseado o se llega al final del fichero.

Inserción. Para insertar un registro, es necesario localizar la posición en que se desea insertar, es decir, entre qué dos registros se quiere situar (registro anterior y posterior). Físicamente, el registro se escribe en una dirección de memoria arbitraria que se encuentre disponible, colocando en su campo de puntero la dirección del registro al cual va a preceder. Luego se modifica el registro anterior para actualizar el valor de su puntero, de forma que contenga la dirección del nuevo registro. El añadir un registro es un caso particular de la inserción, en el que se desea insertar un registro al final del fichero.

Modificación. Si la modificación no implica un aumento en la longitud ni una alteración del valor del campo llave, se localizará el registro y simplemente habrá que sobrescribir en la misma posición. En caso contrario, primero se insertará un nuevo registro que incluya la modificación, y posteriormente se borrará el registro con la información desactualizada.

Borrado. Para eliminar un registro del fichero, sólo hay que destruir su dirección del puntero del registro anterior, copiando en dicho puntero la dirección del registro siguiente al que se desea borrar, es decir, la dirección que contiene el puntero del registro a eliminar.

La velocidad de acceso al fichero dependerá del número de registros lógicos que se puedan leer en un bloque. Si el fichero está compuesto de registros con una longitud mucho menor que el tamaño del bloque, se puede mejorar la velocidad si se leen varios registros en cada acceso al dispositivo, es decir, si en cada bloque incluimos varios registros. En este caso, es necesario encadenar los bloques en lugar de los registros.

La organización secuencial encadenada es útil en ficheros que utilizan procesos interactivos, donde la actualización del fichero es frecuente pero en cada tratamiento se ven afectados pocos registros. La organización secuencial pura es mejor opción cuando el número de registros a actualizar en cada acceso al fichero sea mayor. La principal ventaja de la organización encadenada es la **flexibilidad** (se pueden realizar todo tipo de operaciones). Sin embargo, comparte con la organización secuencial pura el inconveniente de permitir únicamente consulta secuencial: sólo se puede acceder de forma directa al primer registro y al siguiente registro del último procesado; por tanto, para acceder a cualquier registro hay que recorrer todos los que le anteceden.

6. FICHEROS CON ORGANIZACIÓN SECUENCIAL INDEXADA

Un fichero con organización secuencial indexada se considera compuesto por dos estructuras o zonas:

Zona de registros. Contiene todos los registros del fichero, ordenados según el valor de una llave. Su estructura es semejante a la de un fichero con organización aleatoria y direccionamiento directo o a un fichero secuencial puro en el que se pueda direccionar a nivel de registro.

Zona de índices. Se procesa sólo de forma secuencial. Su estructura es la de un fichero secuencial puro en que cada registro contiene dos campos: un campo *llave* (que contendrá algunos valores de la llave del fichero) y un campo *dirección* (que contendrá la dirección de un registro del fichero).

La zona de registros está dividida en tramos lógicos. Cada tramo está formado por una serie de registros consecutivos. Por cada tramo de la zona de registros hay un registro en la zona de índices. En este registro se encuentra el valor mayor de llave del tramo (valor de llave del último registro del tramo) y la dirección del primer registro del tramo. Ambas zonas pueden o no estar en un mismo fichero del sistema. En cualquier caso, la gestión de la estructura la realiza el sistema operativo o un paquete software especial. Por tanto, el usuario de esta estructura no necesita conocer la existencia de ambas zonas, pudiendo contemplarlas como un todo.

Llave	Dirección
Cáceres	1
Logroño	4
Oviedo	8
Zaragoza	10

Zona de índices

1	Almería
2	Álava
3	Cáceres
4	Cuenca
5	Huesca
6	Jaén
7	Logroño
8	Madrid
9	Oviedo
10	Zaragoza

Zona de registros

Con ficheros secuenciales indexados se pueden realizar las siguientes operaciones a nivel de registro:

Consulta. Se pueden realizar consultas secuenciales, como en los ficheros con organización secuencial. Además, se pueden realizar consultas por llave (localizar un registro conocida su llave) sin necesidad de leer los registros que le anteceden en el fichero. Para realizar una consulta por llave se utiliza el siguiente proceso:

- 1). Leer secuencialmente las llaves en la zona de índices hasta encontrar una mayor o igual a la del registro buscado. Obtener la dirección de comienzo del tramo donde se encuentra el registro.
- 2). Leer secuencialmente en la zona de registros, a partir de la dirección obtenida en la zona de índices hasta encontrar el registro buscado o uno con valor de llave mayor que el buscado. En este último caso el registro no se encuentra en el fichero.

Inserción. Dado que ambas zonas son secuenciales, no es posible insertar un registro en ficheros con esta organización. En algunos casos, se permite la escritura de nuevos registros al final del fichero. Estos registros, como es lógico, no podrán ser consultados por llave con el procedimiento antes descrito.

Eliminación. No es posible borrar un registro, aunque sí se permite un borrado lógico (marcándolo).

Modificación. Las modificaciones son posibles tan solo si el registro no aumenta de longitud al modificarlo ni supone una variación del valor de la llave del registro.

Esta organización resulta especialmente útil cuando se deben combinar consultas a registros concretos y el procesamiento secuencial de todo el fichero. Su principal inconveniente es la imposibilidad de introducir nuevos registros o actualizaciones en los registros ya existentes en el fichero sin volver a realizar una *reorganización* del mismo con las modificaciones efectuadas.

Para permitir actualizaciones es necesario incluir en la estructura una **zona de desbordamientos** (o zona de *overflow*). En la zona de desbordamientos los registros están desordenados, cada nuevo registro se escribe al final de ésta sin tener en cuenta el orden de la llave. Esto complica la búsqueda por llave, pues, si el registro no es encontrado en la zona de registros, es necesario buscarlo secuencialmente en la zona de desbordamientos. Además, se imposibilita la consulta secuencial del fichero, ya que, los registros no aparecen ordenados por llave.

7. FICHEROS CON ORGANIZACIÓN SECUENCIAL INDEXADA ENCADENADA

Otra mejora de la estructura anterior es incluir punteros entre los registros, de forma que éstos mantengan el orden lógico de los registros. A esta organización se le llama **secuencial indexada encadenada**. La estructura de un fichero será similar a la de un fichero secuencial indexado, salvo que se ha previsto un campo en cada registro para albergar un puntero.

Para **insertar** un nuevo registro, en este caso, es necesario encontrar el que le sigue en la zona de registros. Se escribe el nuevo registro en la zona de desbordamientos y se reescribe el registro encontrado como siguiente según el orden lógico, para incluir el puntero al registro recién grabado.

La **eliminación** de un registro debe realizarse por marcas. Esto no implica en ningún caso la realización de modificaciones en el índice, pero degrada el fichero. La **consulta** es semejante a la realizada para el caso no encadenado, se averigua la posición de entrada al fichero utilizando la tabla de índices, y se sigue el encadenamiento de los punteros para localizar el registro deseado.

8. FICHEROS CON ORGANIZACIÓN DIRECTA

No tiene porqué existir ninguna relación lógica entre los registros y su colocación física, por eso también se le denomina **organización aleatoria**. Cada registro se sitúa en una dirección de memoria que se calcula para cada uno de ellos aplicando una fórmula o algoritmo matemático.

En la mayoría de los casos, no resulta adecuado realizar una **consulta secuencial** en un fichero con organización directa, ya que los registros no serían leídos según el orden de la llave. Se suele realizar en procesos en los que se tiene que modificar o añadir el mismo campo en todos los registros.

Lo más usual en este tipo de ficheros es utilizar un **acceso directo**. Para ello, se empleará la *llave o identificativo* de los registros, que tiene una doble función: como identificativo de orden y como identificativo de búsqueda.

Existen muchos métodos para generar las direcciones de los registros. Todos ellos tomarán el valor de un campo del registro, aplicarán la transformación y obtendrán la dirección. Por tanto, dependiendo del fichero concreto, se elegirá el método que asegure que las direcciones estarán dentro del rango permitido intentando distribuir las de modo que existan pocos **sinónimos** (registros con diferentes llaves pero con una misma dirección física). Cuando se detecta que la dirección asignada a un registro está ya ocupada por otro, se puede optar por dos alternativas:

- Buscar una nueva dirección libre en el mismo espacio de fichero hasta encontrar una posición libre donde almacenar la información.
- Reservar una zona de desbordamiento e ir almacenando los sinónimos en esta zona de manera consecutiva a medida que van apareciendo.

El problema principal de esta organización es la elección de la función de transformación o **método de direccionamiento** que se ha de usar para cada tipo de fichero. Existen distintos algoritmos para determinar la dirección a partir de una llave:

Direccionamiento directo. Este método consiste en asignar a cada llave una dirección lógica, es decir, la transformación asigna como dirección el valor de la llave. El problema principal radica en que tiene que haber tantas direcciones como valores de la llave, y que las llaves han de ser numéricas. La ventaja es que no se producen sinónimos y el fichero queda ordenado.

Direccionamiento asociado. Normalmente se utiliza cuando las llaves son alfanuméricas, y consiste en asociar a cada llave una dirección lógica por medio de una tabla. Se construye una tabla con todas las llaves del fichero y la dirección física de cada registro. Mediante una búsqueda secuencial en la tabla de llaves, obtenemos la dirección lógica de almacenamiento. Este tipo de almacenamiento no produce sinónimos, pero necesita espacio adicional para la tabla. Se puede hacer también que las llaves se encuentren ordenadas en la tabla, con lo que la búsqueda sería más rápida.

Direccionamiento calculado (hashing). En este tipo, la dirección de cada registro se obtiene mediante una serie de cálculos matemáticos realizados con la llave. El principal problema que plantea es la existencia de sinónimos, pues distintos cálculos pueden llevar al mismo resultado. Existen varios sistemas de cálculo, entre los que podemos citar:

- *División por un número primo.* Si N es el número máximo estimado de registros en el fichero y P es el número primo más próximo a N, se divide la llave por P y se utiliza como dirección el resto de la operación.
- *Cuadrado de la llave.* Si la llave no es un número muy alto, se eleva al cuadrado y se extraen como dirección unas cuantas cifras intermedias.

Ejemplo:

Llave = 856

$856 * 856 = 732736 \Rightarrow \text{Dirección} = 3273$

- *Truncamiento o extracción.* Se trunca la llave, quedándonos con sus 4 o 5 últimas cifras, y a partir de ellas con una serie de cálculos se obtiene la dirección.

Ejemplo:

Llave = 125763213 \Rightarrow Se extraen las 5 últimas cifras y se divide por 57 \Rightarrow

Dirección = $63213 / 57 = 1109$

- *Plegamiento.* Se utiliza cuando la llave toma valores numéricos muy grandes. Consiste en recortar la llave en trozos y después sumar dichos trozos. Al resultado de la suma se le puede volver a aplicar cualquier método de los descritos.

Ejemplo:

Llave = 145627290124 \Rightarrow

Dirección = $145 + 627 + 290 + 124 = 1186$

Las *operaciones básicas* sobre ficheros con esta organización son las siguientes:

Consulta. Se realiza siempre por llave. Para leer un determinado registro, únicamente se ha de aplicar el algoritmo de transformación a la llave. Si el registro buscado no se encuentra en la dirección resultante, se aplicarán los algoritmos de resolución de sinónimos.

Inserción. Para insertar un nuevo registro, se aplica a la llave el algoritmo de transformación elegido. Si la dirección que resulta de aplicar el método de direccionamiento ya está ocupada por otro registro, se emplea un algoritmo de resolución de sinónimos.

Modificación. Siempre se puede realizar esta operación. Para localizar el registro, se aplica la transformación correspondiente a la llave. Una vez que se tiene la dirección, se modifica la información del registro.

Borrado. Se realiza un borrado lógico. Se localiza el registro mediante los algoritmos descritos y se marca con un valor determinado para indicar que no es un registro válido. Normalmente, el marcado consiste en introducir determinado carácter al principio del registro. El espacio del registro eliminado se puede reutilizar (verificando que el nuevo registro tampoco esté en la zona de sinónimos). El inconveniente es que el fichero se va deteriorando paulatinamente. Cuando después de un proceso completo de eliminación queda el fichero más o menos degenerado, se hace imprescindible una reorganización del mismo.

La gran flexibilidad y la rapidez de consulta son algunas ventajas de esta organización. Entre los inconvenientes que presenta, destacan el mal aprovechamiento del espacio, la necesidad de utilizar soportes direccionables y la exigencia de una planificación previa para utilizar el método de direccionamiento más apropiado.

9. BIBLIOGRAFÍA

Alberto Prieto

Introducción a la Informática

Mc Graw-Hill, 2ª edición, 1997

Alfonso Ureña López

Fundamentos de Informática

Ra-ma, 1997