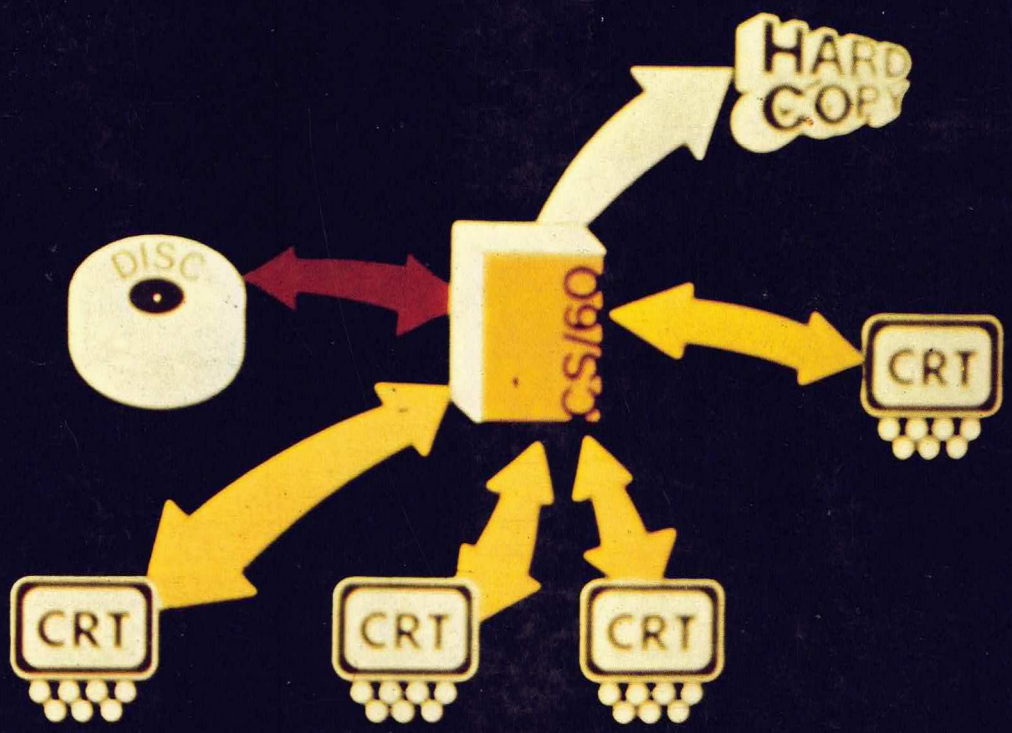
PROGRAMACION DE ORDENADORES

ESTRUCTURA BASICA DEL PROCESO DE DATOS



EDITORIAL tesys

TOMO 5

El presente tratado de **Programación de Ordenadores** está compuesto por un conjunto de temas que constituyen la estructura imprescindible para todo aquel que desee un amplio conocimiento del campo de la Programación.

Dichos temas vienen englobados en volúmenes diversos que a continuación se detallan; constituyéndose por separado como unidades monográficas susceptibles de un estudio particular, al margen del conocimiento conjunto y global de todos ellos, como compendio completo para el logro final de programar.

La agrupación por tomos se desarrolla de la siguiente forma:

Tomo 1. **Estructura Básica del Proceso de datos.**

Tomo 2. **Estructura Básica del Proceso de datos.**

Tomo 3. **Estructura Básica del Proceso de datos.**

Tomo 4. **Estructura Básica del Proceso de datos.**

Tomo 5. **Estructura Básica del Proceso de datos.**

Tomo 6. **Lenguaje Basic**

Tomo 7. **Lenguaje Cobol**

Tomo 8. **Lenguaje Cobol**

Tomo 9. **Leyes de Construcción de programas.**

Tomo 10. **Diccionario de Proceso de datos**

El tratamiento de los temas comprende tanto los aspectos teóricos como las aplicaciones al diseño de estructuras básicas de posible uso inmediato, reagrupándose en los siguientes conjuntos específicos que por sí mismos pasan a ser unidades independientes.

Son los siguientes:

Grupo I: Tomos 1, 2, 3, 4 y 5. **(Estructura Básica del Proceso de Datos).**

Grupo II: Tomo 6. **(Lenguaje Basic).**

Grupo III: Tomos 7 y 8. **(Lenguaje Cobol).**

Grupo IV: Tomo 9. **(Leyes de Construcción de Programas).**

PROGRAMACION DE ORDENADORES

ESTRUCTURA BASICA DEL PROCESO DE DATOS

PROGRAMACION DE ORDENADORES

ESTRUCTURA BASICA DEL PROCESO DE DATOS

AUTOR

JOSE LUIS MORALES PASCUAL
Licenciado en Informática

COLABORADORES

CARLOS ANTONIO MARTINEZ TREVIN
Ingeniero de Telecomunicación

ANDRES J. ERNESTO RUANA GIRONES
Ingeniero Industrial

BARCELONA (ESPAÑA)

© E.C.C. - Barcelona, 1981

Editado por:
EDITORIAL TESYS, Bruch 20, Barcelona-10. (España), 1981

Primera edición: Enero, 1981

Depósito legal: B.: 25.869-1980 (V)

I.S.B.N. 84-85838-00-9 (Obra completa)
I.S.B.N. 84-85838-05-X (Tomo 5)

Impreso por:
GRAFOS, Arte sobre papel.
Paseo Carlos I, 157 - Barcelona (13).

Impreso en España - Printed in Spain

CONTENIDO

Introducción 9

1. Ficheros 11

2. Ficheros de acceso secuencial (I). Tarjetas perforadas 41

3. Ficheros de acceso secuencial (II). Cintas magnéticas 59

4. Ficheros de acceso directo: Discos magnéticos 77

5. Manejo de datos (I). Tablas 95

6. Manejo de datos (II). Listas lineales 117

7. Manejo de datos (III). Arboles 137

Soluciones de los ejercicios 157

Indice general 160

INTRODUCCION

El ordenador maneja los datos según las órdenes que le proporciona el usuario a través de los programas.

Para el tratamiento de estos datos son indispensables los soportes de almacenamiento. En este tomo se estudian, en especial, tres de los soportes de información más utilizados (tarjeta perforada, cinta magnética y disco magnético).

Los últimos capítulos del texto están destinados a la exposición de las distintas estructuras que pueden adoptar los datos en los diferentes medios de almacenamiento, así como el manejo de dichas estructuras de datos.

1.

FICHEROS

CUADRO RESUMEN DE LA LECCIÓN

- Ficheros
  - — Definición
  - — Gestión de ficheros
    - — Identificación
    - — Creación
    - — Actualización
    - — Interrogación
  - — Organización de los ficheros
    - — Secuencial
      - — Sin clave
      - — Indexado
    - — Aleatoria
      - — Indexado
      - — Mapeado
  - — Instrucciones de manejo de ficheros
    - — De identificación del fichero
    - — De control del dispositivo
    - — De lectura y escritura
  - — Funciones para el cálculo de la dirección
    - — Simplificación de un campo
    - — Rutinas de aleatoriedad
  - — Criterios de elección de ficheros

1.1. DEFINICIÓN

Un *fichero* es un conjunto de registros que tienen características comunes, es decir, que están relacionados entre sí.

Los registros que constituyen el fichero pueden ser de diferentes tipos según su función:

— *Registro de datos:* conteniendo los datos del fichero.

— *Registro de definición del fichero:* en donde se guarda la información relativa a la estructura del fichero, a su organización, longitud, etc.

— *Registro índice:* conteniendo la información relativa al lugar que ocupan algunos o todos los registros dentro del fichero.

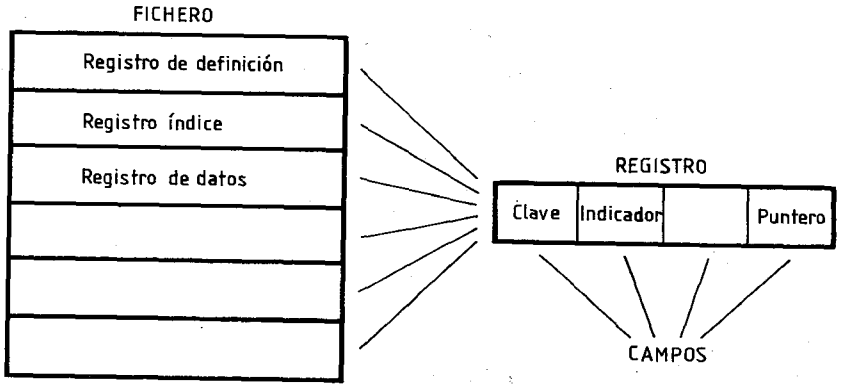


Fig. 1.1. Ficheros, Registros y Campos.

A su vez, un registro está formado por *campos.* En cada uno de los campos se guarda distinta información perteneciente al mismo registro.

Un registro puede tener diferentes tipos de campos:

— *Campo clave:* contiene información útil para la identificación del registro. Esta *clave* puede usarse para realizar una clasificación de todos los registros del fichero.
Algunos registros no tienen clave, otros tienen uno o varios campos clave.

— *Campo puntero:* contiene información de la posición de otros registros del fichero. En un registro pueden colocarse más de un puntero según se quiera enlazar con uno solo, o con varios otros registros. Algunos registros no poseen punteros (por ejemplo, en ficheros de acceso secuencial).

— *Campo indicador:* es un campo especial que señala alguna situación particular del registro donde está situado. Por ejemplo, el indicador puede encontrarse en un estado en el que indica que no se graba nada en el registro, mientras que cuando se encuentra en otro estado, entonces puede grabarse.

Ejemplo

Se quiere organizar todos los libros de una biblioteca en un fichero. Posteriormente, se añadirán ficheros de lectores y préstamos con lo que se completarán los elementos para la mecanización de la biblioteca.

El fichero de libros está formado por registros. En cada uno de los registros se guarda la información relativa a un libro. Los campos de que consta este registro son los siguientes:

— título del libro
— autor primero
— autor segundo
— año de edición
— editor
— número de volumen o revista
— puntero 1
— puntero 2
— puntero 3
— signatura o código de libro.

En cada uno de los punteros se guardará información de los registros de fichero de lectores y préstamos, por ejemplo, el lugar donde está el registro que contiene la información del lector al que se le ha prestado el libro.

1.2. GESTION DE FICHEROS

La descripción lógica trata de todo lo referente al tratamiento de ficheros por el ordenador al conjunto de los programas que los manejan.

Las cuestiones que tienen que resolver los distintos programas son las siguientes:

1.2.1. Definición de la estructura del fichero

La primera operación que debe realizar un programa de tratamiento de ficheros es la identificación de éste.

Esta identificación o definición del fichero sirve al programa para conocer la forma del fichero, cómo son cada uno de los registros, qué tipo de organización se usa, cuál es la longitud del fichero, etc.

1.2.2. Creación del fichero

Una vez definida la estructura que se le va a dar se procede a llenar la memoria auxiliar con todos los datos que deba poseer el fichero. Estos datos deben estar organizados en registros y estos registros en campos.

Los problemas existentes en la creación de un fichero se exponen a continuación:

Primero. Se debe conocer el formato de información de un registro. Es decir, cómo está estructurado el registro en sus diversos campos.

Segundo. Validación de la información. Cuando un registro tiene que incorporarse a un fichero se debe comprobar que éste corresponda al formato de cada uno de los registros.

Esta validación debe realizarse en dos aspectos:

— Validación sintáctica, que la información corresponda al formato.

— Validación semántica, que el contenido sea el esperado en cada uno de los campos.

Por ejemplo, se esperan registros que tengan dos campos, uno alfanumérico de 30 caracteres y el otro numérico de 12 dígitos. Si en ese momento llega un registro con ambos campos numéricos, este registro debe ser rechazado por el programa ya que tiene la información errónea. Si llega un registro con un campo alfanumérico que tenga 40 caracteres seguido de un campo numérico de 12 dígitos, también este debe ser rechazado por no cumplirse las condiciones semánticas.

Tercero. Posición del registro dentro del fichero. Después se verá que es en el caso de acceso aleatorio en el que ésta cuestión tiene importancia.

1.2.3. Actualización del fichero

Esta operación de puesta al día del fichero se efectúa cuando han habido cambios en los datos del fichero. Los cambios pueden ser de tres tipos:

— Inserción de nuevos elementos.
— Supresión de elementos existentes en el fichero.
— Modificación de registros del fichero.

El modo de realización de cada uno de estos es distinto para cada organización de fichero que se tenga.

1.2.4. Interrogación del fichero

Este punto trata del uso del fichero. En algún momento la información que ha sido almacenada en él debe ser usada, pero quizá no toda sino sólo la de algunos registros. La interrogación busca algún método de conseguir esos registros. Al igual que las otras operaciones, dependerá de cómo se haya estructurado el fichero.

1.3. ORGANIZACIÓN DE LOS FICHEROS

Los datos almacenados en los distintos ficheros deben cumplir ciertas características. Para que el proceso de estos datos sea efectivo deben estar correctamente organizados.

El tipo de memorias auxiliares condiciona la organización de los datos. El tiempo de acceso a los datos viene condicionado por la estructura del registro lógico, por el tipo de organización y por el tipo de direccionamiento.

Existen dos tipos de ficheros según la organización o el modo de acceder a los datos:

— Ficheros de acceso secuencial
— Ficheros de acceso directo.

1.3.1. Ficheros de acceso secuencial

El *acceso secuencial* es un proceso que implica la lectura o escritura de datos en serie.

Un fichero de acceso secuencial es un dispositivo de memoria estructurado de tal manera que el acceso (escritura o lectura) a las posiciones de almacenamiento es en orden secuencial.

Los datos almacenados en un fichero de acceso secuencial se encuentran clasificados por un número de identificación siguiendo un orden (creciente o decreciente).

Los datos de un fichero secuencial deben leerse en el mismo orden (o en orden inverso, en algunos casos) en que fueron almacenados. No es posible el acceso a una posición (registro o bloque) determinada previamente.

Ejemplos de ficheros de acceso secuencial son las fichas perforadas, las cintas perforadas y las cintas magnéticas. En estos soportes, los datos sólo pueden ser procesados secuencialmente.

El acceso a un dato almacenado en una cinta magnética se efectúa mediante el movimiento de la cinta hacia adelante o hacia atrás de forma que todos los datos van pasando por la cabeza de lectura/grabación de la cinta.

Pueden distinguirse dos tipos de ficheros secuenciales:

— Secuenciales sin clave.
— Secuenciales indexados.

Secuenciales sin clave

Estos ficheros son los propiamente secuenciales. Su característica es que un registro sucede al otro en un orden determinado. Este orden puede ser

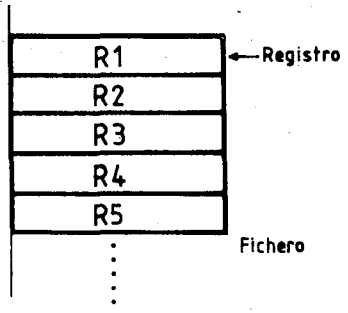


Fig. 1.2. Organización secuencial sin clave.

meramente cronológico de creación del fichero, o puede darse por la clasificación de alguno de los campos de que consta el fichero (clasificación numérica o alfabética).

Características

La frecuencia de los accesos a estos ficheros puede ser muy elevada pero siempre que se acceda a todos los registros. No puede accederse a un registro particular sin haber revisado todos los anteriores. Esto es debido a que los registros no tienen ninguna dirección para ser identificados.

Las modificaciones en los ficheros secuenciales es muy costosa por lo que no debe realizarse con demasiada frecuencia.

Es conveniente que cuando se diseña un fichero de este tipo se deje espacio, al final del fichero, para las inserciones. Las inserciones en el interior del fichero no son posibles, excepto en el caso en que se vuelva a copiar todo el fichero de nuevo, o por lo menos a partir del punto en que se realiza la inserción.

Las eliminaciones también requieren recopia si se hacen en el interior del fichero, puesto que los registros que forman el fichero, deben aparecer uno a continuación del otro. Se deben recopiar todos los registros a partir de donde se ha hecho la eliminación para que queden nuevamente de forma correcta.

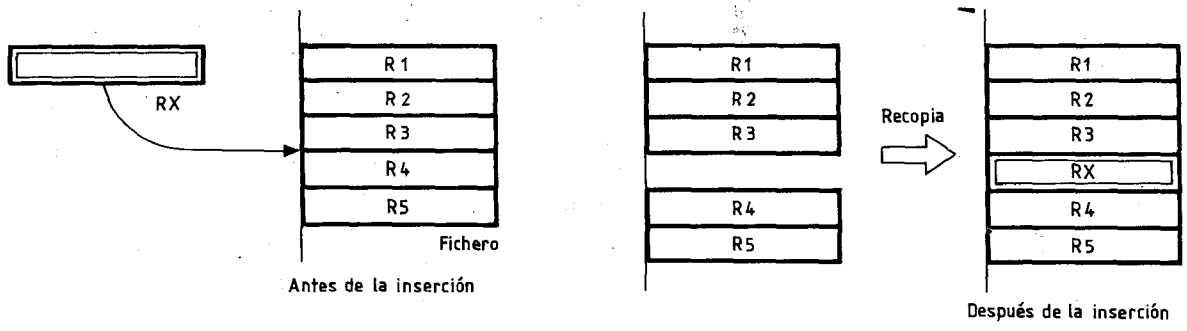


Fig. 1.3. Inserción de un registro en un fichero secuencial sin clave.

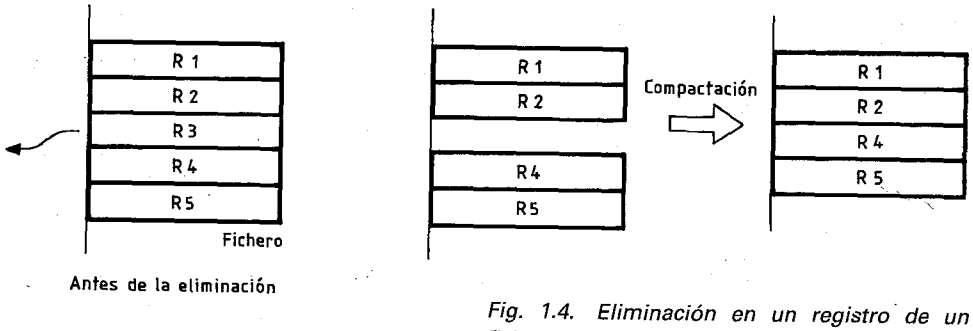


Fig. 1.4. Eliminación en un registro de un fichero secuencial sin clave.

Los cambios no pueden realizarse a menos que el registro nuevo ocupe la misma longitud de palabras que el antiguo, en cuyo caso el nuevo ocupa completamente el lugar antiguo. En los otros casos es necesaria una recopia de todo el resto del fichero.

Operaciones

La creación de un fichero secuencial sin clave consiste en la grabación sucesiva de todos los registros del fichero. Cada registro que se debe grabar se añade a continuación del ya grabado, y así sucesivamente hasta que se acaben todos los registros.

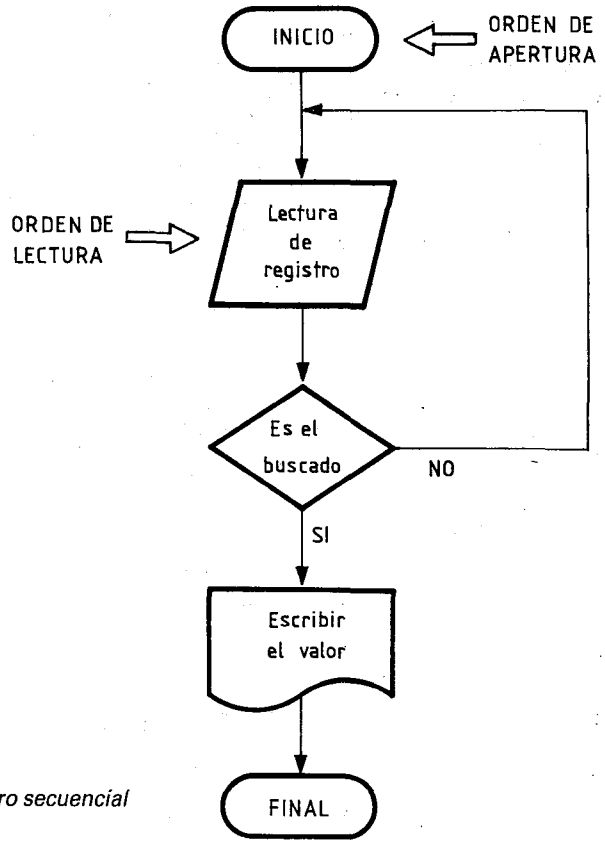


Fig. 1.5. Interrogación de un fichero secuencial sin clave.

La puesta al día de un fichero requiere una recopia de la información en un fichero, que se llamará *actualizado,* a partir del fichero base y del fichero de modificaciones.

La interrogación del fichero debe hacerse leyendo cada uno de los registros. En la Fig. 1.5. se presenta en un organigrama que resume la acción.

Ejemplo

Se considera un fichero en el que se va a guardar información referente a la población forestal. En cada uno de los registros se coloca la información referida una especie arbórea diferente.

Los campos de que consta cada uno de los registros son:

— Nombre científico de la especie de árbol.
— Superficie arbolada sin aprovechamiento comercial después del incendio (miles de hectáreas).
— Superficie arbolada con aprovechamiento comercial después del incendio (miles de hectáreas).
— Superficie arbolada total (miles de hectáreas).
— Volumen de madera (m^3).
— Pérdidas en madera (miles de pesetas).
— Gastos de extinción (miles de pesetas).

Organización secuencial indexada

La organización del fichero sigue siendo secuencial pero dispone de un mecanismo de identificación de los registros almacenados.

Tienen en primer lugar un registro índice que contiene las claves de los últimos registros de cada bloque físico del fichero. La ventaja que se obtiene con el método es que no se tienen que revisar todos y cada uno de los registros de que consta el fichero, sino sólo los registros índice. Estos, que se encuentran en primer lugar del fichero, llevan directamente al bloque donde está grabada la información buscada.

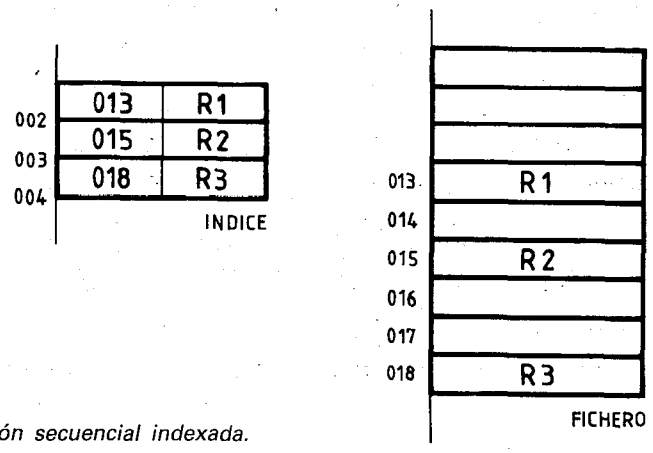


Fig. 1.6. Organización secuencial indexada.

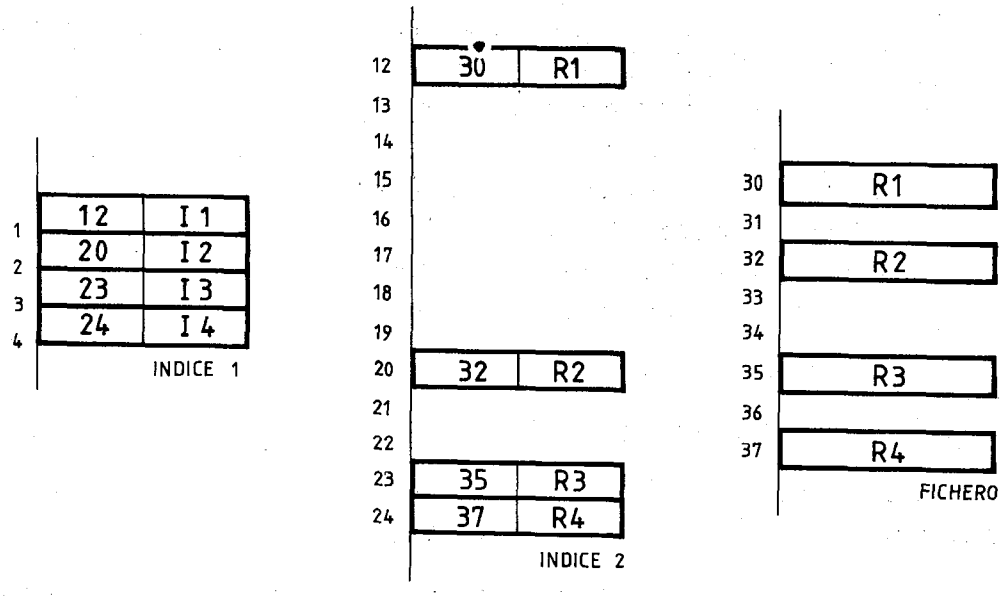


Fig. 1.7. Niveles de indexación.

Características

La frecuencia de los accesos a los registros particulares puede ser mayor que en el caso anterior. Ahora sólo se debe buscar secuencialmente en un conjunto de registros índice (a veces llamado *índice del fichero*) y, con el dato encontrado en éste, incidir directamente en el registro buscado.

Para que la búsqueda sea más rápida, se puede crear otro índice conteniendo la información sobre la posición de los registros que forman el índice del fichero (se hace un índice del índice). La realización de varios índices encadenados se denomina creación de *niveles de indexación.*

Las modificaciones son más fáciles de realizar. Las inserciones y eliminaciones requieren una modificación del índice único o de todos los niveles de indexación.

Las inserciones se realizan en el mismo bloque si hay lugar para ellas. Si no hay lugar se llevan a un *bloque de desbordamiento* donde se almacenan como continuación lógica del bloque donde se debían haber colocado. Se debe

conocer las características del bloque de desbordamiento. Detalles como lugares libres, driecciones de esos lugares, espacio que aún queda libre, etc. Deben estar contenidos en variables. Estas deben ser exploradas antes de efectuar las inserciones. El conjunto de estas variables se conoce con el nombre de *registro de control.*

La eliminación de algún registro se realiza recopiando el nuevo bloque que queda después de la eliminación. Otro método que no precisa recopia sería la colocación de un *indicador de registro eliminado.* El lugar de este registro puede usarse para la inserción de un nuevo registro.

Se ve que por el modo de realizar las actualizaciones, la frecuencia de las modificaciones puede ser mayor que en el caso secuencial sin clave. En el caso indexado se tarda menos tiempo en realizar cada una de ellas (sólo se tiene que modificar un bloque en lugar de todo el fichero).

Por ser un fichero secuencial, sin hacer caso del índice, el acceso en un orden predeterminado es posible y fácil de realizarse.

Los soportes que se utilizan para esta organización son los que permiten acceso directo, por ejemplo, los discos magnéticos. Un soporte de acceso secuencial no puede ser usado ya que no dispone de direcciones para sus posiciones de almacenamiento. Las direcciones son necesarias aquí porque el registro índice las contiene.

Operaciones

La creación de fichero tiene una doble actividad. Por un lado se debe crear el índice. Con las direcciones que contiene este índice se accede a las posiciones de almacenamiento del soporte y se graba el bloque (el bloque es un registro formado por un conjunto de registros lógicos). Las dos actividades son pues:

— grabación del fichero
— grabación del índice.

La puesta al día se comentó en el apartado anterior.

La interrogación se efectúa en dos pasos. En primer lugar se busca en el índice la dirección donde está grabado el registro. Esta búsqueda se efectúa secuencialmente, de modo que sigue el esquema presentado en el apartado de ficheros secuenciales sin clave.

En segundo lugar se busca el registro donde está almacenada la información buscada y se lee. Esta búsqueda no es tal búsqueda sino un acceso directo, puesto que gracias al índice ya se conoce la dirección donde se puede encontrar el registro.

Ejemplo

Se va a crear un fichero para la mecanización de los historiales clínicos en un hospital. Cada uno de los registros tiene información sobre un enfermo.

Los campos contienen:

— Código de entrada.
— Nombre del enfermo.
— Dirección.
— Antecedentes familiares.
— Exploración.
— Diagnóstico.
— Enfermedad actual.
— Tratamiento.

El *código de entrada* será el número que se le dé al enfermo y servirá a la vez de índice.

El índice estará compuesto por todos los registros que contendrán esos códigos de entrada.

1.3.2. Ficheros de acceso aleatorio

En los ficheros de acceso directo o *aleatorio* se accede a cualquier registro del fichero directamente, mediante una clave. Por esta razón, en ningún caso pueden ser colocados en soportes secuenciales.

Existen dos tipos de ficheros aleatorios:

— Acceso aleatorio indexado.
— Acceso aleatorio mapeado.

Ficheros de acceso aleatorio indexado

La particularidad de estos ficheros es que tienen un índice con una clave que indica dónde se puede encontrar el registro buscado. En estos ficheros, los

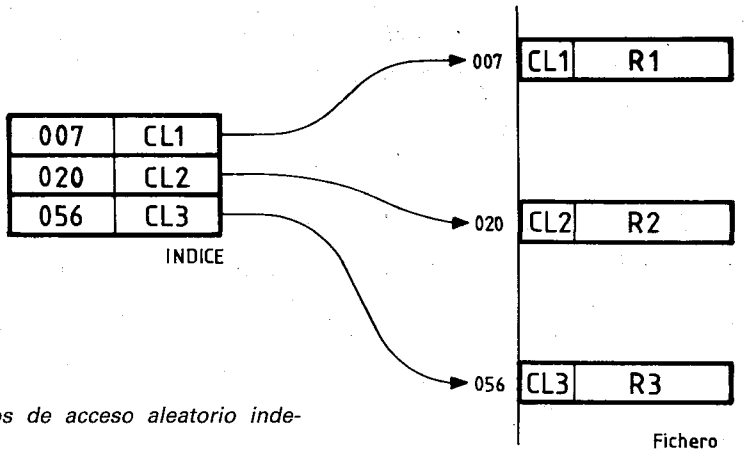


Fig. 1.8. Ficheros de acceso aleatorio indexado.

registros no están ordenados respecto de clave alguna. A consecuencia de esto, el índice debe contener la dirección de todos los registros, y no sólo los de final del bloque como en el caso de los ficheros secuenciales indexados.

Pueden crearse varios índices, cada uno de ellos con una clave diferente. Gracias a esto se pueden acceder de diferentes modos a los registros del fichero.

Características

La frecuencia de los accesos puede ser elevada, ya que el acceso consiste en buscar en dos tablas directamente. Primero en la del índice y después en el fichero de datos. Este acceso es rápido.

Las inserciones se realizan al final del fichero o donde haya lugar para ello, ya que no existe ninguna ordenación para los registros de que se compone el fichero. La eliminación y modificación se realiza con la misma facilidad.

Los accesos en orden predeterminado son posibles cuando se crea una variable a la que se va haciendo variar de valor. Los distintos valores que contienen serán, sucesivamente, los de las claves del índice. La utilidad mayor que tiene esta organización es cuando el acceso se hace directamente sobre un registro particular o un grupo de registros, y no cuando se tiene que explorar el

fichero entero. En este último caso, una organización secuencial tiene más ventajas.

El único soporte que se puede utilizar es el que permita un acceso directo (discos magnéticos, tambores magnéticos).

Operaciones

La creación de un fichero de este tipo no requiere mucha complicación.

Se van grabando uno por uno los registros en las posiciones de memoria que se indiquen en la clave. Además, se debe ir creando el índice que almacenará las posiciones donde se han ido grabando los registros.

La interrogación del fichero se efectúa en dos partes. En primer lugar se realiza la búsqueda de la clave en el índice. Este valor se usa en la segunda parte para posicionar el sistema de lectura de la unidad de almacenamiento delante del área donde se va a leer. Es siempre un acceso directo porque se conocen las direcciones.

Fichero de acceso aleatorio mapeado

Esta es otra organización de ficheros de acceso directo. La diferencia con el anterior es que el valor de la dirección donde está el registro se calcula mediante una expresión matemática, teniendo como dato el valor de la clave.

Características

La frecuencia de los accesos puede ser muy grande y los accesos muy rápidos cuando se trata de obtener un solo elemento. Para la revisión de todos los registros del fichero se necesita calcular la dirección de cada uno de ellos y en este caso el fichero secuencial es más conveniente.

Las modificaciones se pueden realizar con facilidad. Para las inserciones es necesario poseer los datos acerca de un *registro de desbordamiento* si los registros que se van a insertar no caben en la posición de memoria que se les da por el cálculo. Cuando se tiene que insertar un registro nuevo se calcula la dirección por medio de la fórmula. Se mira si esta dirección está ocupada por algún registro. En caso de que esté ocupada se busca el registro de desbordamiento de este bloque y se coloca la información en él, con el

conveniente puntero que enlace la dirección donde debiera ir con la dirección donde realmente se coloca.

Cuando el lugar calculado por la fórmula no está ocupado, el registro se coloca en esa posición.

Las eliminaciones se realizan de la misma manera. Se localiza, en primer lugar, la posición donde está el registro que se va a eliminar. En esa posición estará el registro buscado o estará otro. Este otro tendrá un puntero indicando a la dirección donde se encuentra el bloque de desbordamiento. Allí se encontrará el registro para eliminar, acción que se hará a continuación. Se deberá poner un indicador en esta posición para que a partir de ahora se tome por una posición libre.

Las modificaciones se hacen según los mismos principios que las inserciones y eliminaciones.

Los accesos son siempre de tipo directo, con facilidad para acceder por clave. Va a existir un campo, clave, por el que se identificará al registro y por el que puede reconocerse.

Los soportes físicos deben ser siempre soportes con acceso directo, es decir, que dispongan de direcciones de almacenamiento. Estas direcciones serán siempre resultados de los cálculos para encontrar una posición de memoria para un registro.

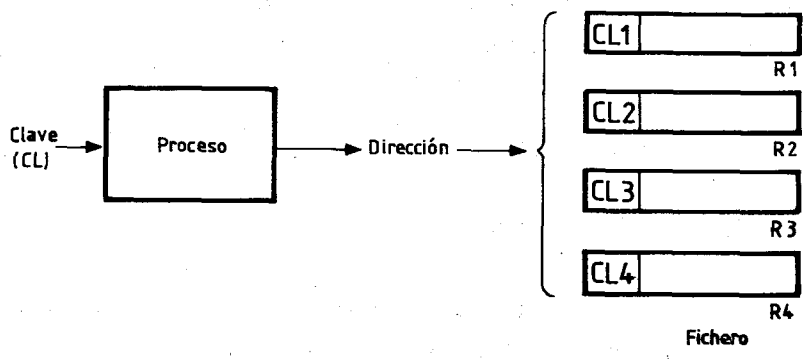


Fig. 1.9. Fichero de acceso aleatorio mapeado.

Operaciones

La creación del fichero exige sólo el cálculo de la posición para cada uno de los registros que lo componen. En caso de colisión (registros que deban ir en la misma posición según el cálculo efectuado), se colocan los nuevos registros en otras posiciones de memoria. Estas posiciones constituyen *bloques de desbordamiento.* Es pues necesario colocar en los registros campos en los que se indique el lugar vacío que queda en el bloque y la posición de los bloques de desbordamiento que se usarán para ese bloque.

La puesta al día, que incluye las operaciones de inserción, eliminación y cambio de registros se comentó en el apartado anterior.

1.4. INSTRUCCIONES DE MANEJO DE LOS FICHEROS

Para la utilización de un fichero en un programa se deben usar las siguientes instrucciones.

— Instrucciones de identificación del fichero.
— Instrucciones de control del dispositivo.
— Instrucciones de entrada y salida para el dispositivo que controla el soporte.

1.4.1. Las instrucciones de identificación del fichero

Contienen información de cómo está formado el fichero, qué clase de acceso va a usar, qué tipo de soporte se usará, cuál es la longitud del fichero, si existe o no bloqueo y qué factor de bloqueo se usa.

Un ejemplo de instrucción de identificación de fichero es:

```
FILES 8-MT (RI = 20, BL = 204)
```

El lenguaje usado para la codificación ha sido el FORTRAN. La descripción de cada uno de los contenidos es la siguiente:

FILES : Código que indica que se trata de un archivo en acceso secuencial.

8 : Es el número que se da al periférico utilizado. Siempre que el programa se deba referir a ese periférico para escritura, lectura o cualquier otra actividad, lo hará con ese número.

MT : Indica que el soporte usado es una cinta magnética (Magnetic Tape).

RL = 20 : Significa que la longitud de los registros lógicos del fichero es de 20 bytes. Esta longitud se puede calcular del formato de los datos que se van a colocar en el registro.

BL = 204: Indica que se ha hecho un bloqueo con 204 bytes. El factor de bloqueo es 10(10 × 20 = 200), de modo que se cogen bloques de 10 registros. La colocación de una longitud del bloque (número de palabras que contiene) de 204 en lugar de 200, como corresponde a los cálculos, significa que el sistema se reserva 4 bytes para la formación del bloque, esos bytes sirven para control interno del manejo del fichero.

Una instrucción de definición de fichero (identificación) para un caso de acceso directo es la siguiente:

```
DEFINE FILE 8 (25,38,E,IPOS)
```

que significa que en el periférico 8 se crea un fichero de 25 registros, de una longitud de 38 bytes. El fichero se formará con un formato y el índice estará designado por la variable IPOS.

Usualmente es necesario un conjunto de fichas de control del sistema.

1.4.2. Las instrucciones de control del dispositivo

Son las que manejan el controlador del soporte. Tienen más importancia en soportes secuenciales que en los de acceso directo, porque en aquellos el programa no puede controlar el fichero, ya que es un sistema secuencial y va tratando registro por registro.

Instrucciones de este tipo para las cintas pueden ser:

— Rebobinar la cinta.
— Colocar el puntero del controlador en el registro físico tratado en último lugar.
— Poner una marca de fin de fichero.

En ficheros de acceso directo existe una instrucción que significa:

Buscar la posición donde se ha grabado el registro que tiene como clave XX (XX se debe interpretar como un número de los válidos para esa clave). La actuación del mecanismo es el posicionamiento de su cabeza (lectura y escritura) encima de la posición donde está colocado ese registro.

1.4.3. Las instrucciones de lectura y escritura

Son las que graban o extraen los contenidos de las posiciones de memoria auxiliar. En estas instrucciones se debe incluir la información sobre el periférico con el que se va a trabajar, las variables que corresponden a cada uno de los campos del registro y los formatos de escritura o lectura para cada una de las variables. También parte de control del programa como: qué debe hacer el programa en caso de error en la lectura, qué debe hacer cuando acabe de leer todo el fichero, etc.

Ejemplo de estas instrucciones es:

```
READ (8,540,ERR = 30, END = 40) A,M,B,C
```

que significa que se lea un fichero grabado en el periférico número 8. Cuando exista un error en la lectura la siguiente instrucción a ejecutar es la que tiene la etiqueta 30. Cuando se acabe el fichero el flujo del programa pasa a la instrucción con etiqueta 40. Cada uno de los registros consta de cuatro campos nombrados respectivamente por las variables A,M,B,C. Cada uno de los campos del registro ocupa una zona de grabación que es la indicada por el formato que está en la instrucción con etiqueta 540.

1.5. FUNCIONES PARA CALCULO DE LA DIRECCION

La interrogación del fichero necesita del cálculo de la posición en memoria del registro que está con esa clave.

Existen diversas funciones para el cálculo de la dirección.

La más natural que aparece es usar el mismo valor de la clave como posición de almacenamiento. Ese es el caso de los registros de acceso aleatorio indexado.

Por ejemplo se hace un fichero de artículos para el inventario de un almacén. A cada artículo se le puede dar un valor numérico que lo identifique, esa sería su clave. Además este valor numérico correspondería con la posición en el disco (si se toma ese soporte) en la que se va a almacenar.

La única condición que se debe cumplir para que esta asignación de posiciones de memoria sea válida es que no haya más elementos en el fichero que posiciones de almacenamiento en el disco.

Si en el stock (cantidad de piezas almacenadas) de un almacén se han numerado cada una de las piezas desde 1 hasta 300, se necesitará un conjunto de discos con, por lo menos, 300 posiciones de almacenamiento.

El fichero consta de registros. En cada uno de ellos se guarda información sobre un artículo del stock en particular. Los campos de que constaría el registro son los siguientes:

— Número del producto (que constituye la clave del registro).
— Precio unitario.
— Cantidad en almacén.
— Límite mínimo de la cantidad almacenada.
— Descripción del producto.

Un ejemplo de uno de los registros puede ser: (se separan los distintos campos por un espacio en blanco).

1 6.00 100 30 mesas desmontables.

Aquí el primer campo de este registro, que contiene un 1 sirve como clave, indicando que se guardará en la primera posición de almacenamiento del conjunto de discos magnéticos. En este caso como existen 300 diferentes, cada uno con una clave diferente, y 300 posiciones de almacenamiento no existe ninguna colisión.

1.5.1. Simplificación de un campo para formar la clave

Puede utilizarse cualquier función que tenga como resultado números menores que las posiciones de almacenamiento.

Se presenta otro ejemplo en el que se hace uso de otra función: Se quiere crear un fichero de datos a partir de documentos de identidad de 100 personas. Cada uno de los registros dispone de los datos de una de las 100 personas. La longitud de este fichero es de 100 registros.

Se va a considerar un documento de identidad que conste de 4 dígitos. La numeración de este documento puede variar de 0000 a 9999, lo que significa que pueden existir 10000 documentos de identidad diferentes.

Puesto que el fichero que se va a realizar consta sólo de 100 registros, no se va a usar la clave: *número de carnet de identidad* como posición de memoria auxiliar donde se guarda el registro. La razón es que el sistema debería disponer de una memoria auxiliar capaz de disponer de 10000 posiciones de almacenamiento, de las que sólo se usarían 100. Con esa disposición quedarían 9900 posiciones sin usarse, esto es una utilización de, tan sólo, el 1 % de la memoria.

El siguiente paso es encontrar alguna modificación al número de carnet de identidad para que sea utilizable como clave y dirección de memoria.

Uno de los modos es utilizar las dos últimas cifras del número del documento de identidad. Con estas dos cifras tenemos como máximo 100 situaciones diferentes del 00 al 99. El inconveniente que se le ve a esta clave es que habrán personas diferentes con las mismas dos últimas cifras del documento de identidad.

Esta coincidencia originará colisiones que tendrán que resolverse con bloques de desbordamiento.

Los campos de que consta un registro para este fichero serán los siguientes:

— Número de carnet de identidad.
— Nombre.
— Dirección.
— Fecha de expedición.
— Lugar de expedición.
— Profesión.
— Observaciones.

Supóngase que vamos a almacenar los siguientes 4 registros del conjunto de los 100.

1924 Juan Montornés .....
1808 Pedro Garrido .....
5624 Ana López .....
3509 Laura Díaz .....

El registro con clave 1924 se coloca en la posición de memoria 24 (correspondiente a las dos últimas cifras).

El registro 1808 en la posición 08.

El 5624 ya no se puede colocar en la 24 como le correspondería, porque está ocupada. Se coloca en la 99 y además se coloca un puntero que indique la posición en la que debería ir, la 24. En el registro de la 24 se coloca un puntero que señala a la 99, donde está el siguiente elemento del bloque.

El registro de número 3509 se coloca en la posición 09. La disposición de los registros se muestra en la figura 1.10.

Con la incorporación de estos punteros los campos que tiene cada registro son:

— Clave: número de documento de identidad.
— Todos los datos anteriores de la persona.
— Puntero 1: indica la posición correcta en que tendrá que ir el registro.
— Puntero 2: indica la posición del siguiente elemento con colisión (la posición donde se almacena el siguiente elemento que tiene que ir en esa posición).

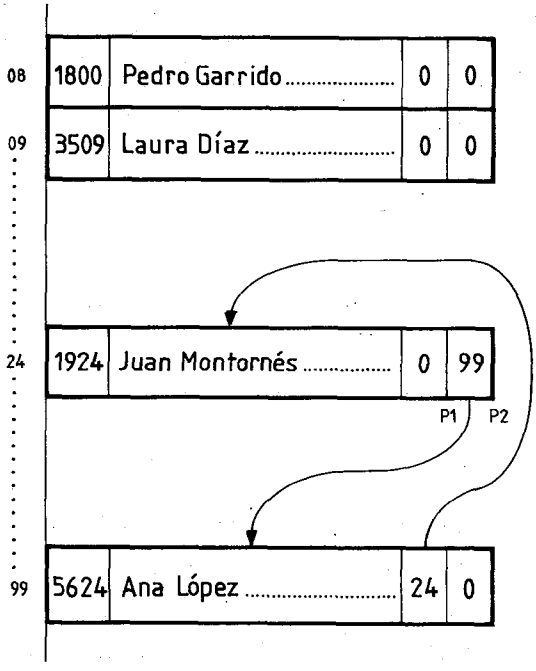


Fig. 1.10. Elección de la clave en un fichero de acceso aleatorio mapeado.

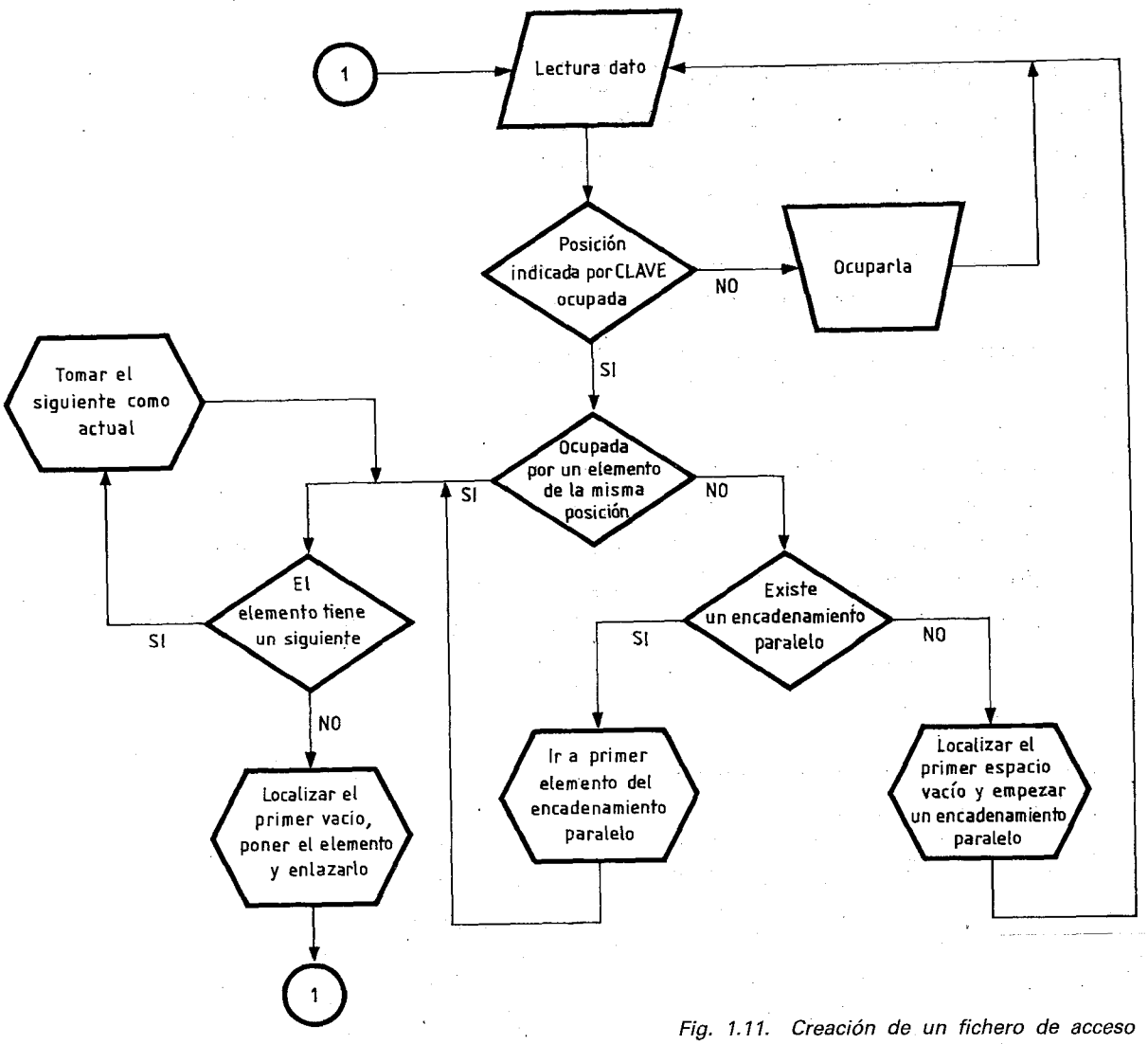


Fig. 1.11. Creación de un fichero de acceso aleatorio mapeado.

Un algoritmo para la creación de un fichero de este tipo es el de la fig. 1.11. Las decisiones de si existe una cadena paralela (un bloque de desbordamiento) o de si el elemento pertenece a la misma posición se hacen mediante los correspondientes punteros. Uno de ellos indica la posición en la que debería colocarse el elemento y el otro indica el elemento siguiente del encadenamiento.

1.5.2. Rutinas de aleatoriedad

Otras funciones muy usadas para el cálculo de la dirección son las *rutinas de aleatoriedad.* Estas funciones generan números comprendidos entre los límites de posiciones de almacenamiento, a partir de los números que constituyen la clave, que puedan tener cualquier margen de variación.

Método del resto de la división

Ejemplo:

Se quiere grabar un conjunto de 9600 registros. Para ello se han necesitado 25 pistas de un conjunto (cilíndrico) de discos, es decir, en total 25 * 20 = 500 pistas en el dispack.

La clave consiste en un número que puede variar de 100000 a 9000000.

Se diseñará una función que calculará sucesivamente: la pista en la que se va a grabar, y el número de cabeza de grabación que debe usarse.

La función usa un método llamado de «resto de la división». Con estos 25 conjuntos se tiene un total de 20 * 25 = 500 pistas, que se numerarán de la 000 a la 499.

El número clave se divide por el número primo (aquel que sólo puede dividirse por uno y por sí mismo con resto cero) más cercano e inferior al número total de pistas. En este caso, siendo el total de pistas 500, el número primo más cercano e inferior es 499. La división por este número siempre proporciona un resto comprendido entre 000 y 498. Este resto designará el número de pista que va a usarse.

Estas pistas están distribuidas en 25 cilindros, que vamos a numerar del 00 al 24.

El número de la cabeza de escritura que va a usarse para la grabación se calcula dividiendo este número de pista que ha quedado por 20, las cabezas que hay por cada conjunto. El resto de la división variará siempre entre 00 y 19. Para calcular el conjunto de discos que va a ser utilizado se suma el cociente número de cabeza el número del primer disco que va autilizarse. Para este ejemplo los 25 conjuntos se han distribuido desde el 60 al 84. Por lo tanto se añadirá 60 a este número de cabeza que ha salido del cálculo.

Ejemplos del uso de este método para el cálculo de la posición de memoria en este caso particular son:

Grabación de 254932 (clave). El resto de la división por 499 es 442. El resto de la división de 442 entre 20 es 2. De estos resultados se tiene que el conjunto usado es (60 + 22), 22 es cociente 442/20.

Otros valores de la clave darían como direcciones:

Clave	*Pista*	*Número de conjunto de cilindro*	*Número de cabeza*
9000000	36	61	16
1048342	442	82	2
794210	301	75	1
254932	442	82	2
100000	200	70	0

De los resultados obtenidos en la tabla se ve que existe colisión. Pues hay dos números claves, el 1048342 y el 254932 que llevan a la misma dirección de pista. Si en esa pista no hubiera lugar para los dos registros se colocaría el sobrante en una zona de desbordamiento.

1.6. CRITERIOS DE ELECCION DE FICHEROS

Los ficheros se clasifican en dos tipos, según el modo de acceder a los datos. La elección de uno u otro se realiza en base a las cuestiones que se expresan a continuación.

Frecuencia de los accesos

Por *frecuencia de acceso* a un fichero se entiende el número de veces que se tiene que usar la información almacenada en este fichero. Si este número es muy grande se dice que la frecuencia de acceso es alta y se utiliza un tipo de fichero mejor que el otro.

Frecuencia de las modificaciones

Por modificación de un fichero se entiende la realización de una o varias de las tres operaciones siguientes:

— inserción de registros nuevos
— eliminación de registros existentes
— modificación de la información contenida en alguno de los registros.

El modo de realizar las modificaciones sigue el método expresado a continuación.

Se necesita el *fichero base* o fichero maestro (el antiguo sin las modificaciones) y el *fichero de transacciones* o *movimientos,* donde se han grabado las modificaciones que se deben realizar. Estos dos ficheros se combinan en uno solo que es llamado el *fichero actualizado.* Este fichero contiene toda la información del fichero base convenientemente actualizada con las modificaciones del fichero de movimientos.

Realización de los accesos en un orden predeterminado

El modo exacto en que se van explorando los registros del fichero puede incidir en la elección de un tipo u otro. No es lo mismo explorar algún registro aleatoriamente, sin conocer el orden en que va a aparecer, que explorar todos y cada uno de los registros, uno detrás del otro.

Características del soporte físico

Se debe conocer el soporte que se va a usar para el almacenamiento. No todos los soportes físicos permiten almacenar todos los tipos de ficheros que se pueden diseñar.

Existen dos tipos de soportes. Aquellos en los que puede accederse independientemente a cada una de las disposiciones de almacenamiento y que está indicada por una dirección. El otro tipo es el que ninguna de las posiciones de almacenamiento es identificable. Se debe acceder secuencialmente a cada una de ellas. Cuando se ha grabado un registro se graba, a continuación el siguiente, y así sucesivamente.

Ejemplo del primer tipo (ficheros de acceso directo) son los discos magnéticos. Un soporte del segundo tipo (ficheros de acceso secuencial) es la cinta magnética.

EJERCICIOS

1. La interrogación de un fichero es:

   a) Preguntarle su nombre.
   b) La supresión de elementos existentes en el fichero.
   c) Buscar algún método de conseguir un registro determinado.

2. En un registro de definición de fichero se guarda:

   a) La información relativa al lugar que ocupan otros registros.
   b) La información que debe contener el fichero, es decir, los datos.
   c) Información relativa a la constitución del fichero.

3. Un registro índice contiene:

   a) Información relativa a la longitud de los registros.
   b) Información relativa al lugar que ocupan los registros en la memoria auxiliar.
   c) El índice de todos los ficheros existentes en la cinta.

4. Un campo clave:

   a) Nunca puede contener un número.
   b) Se caracteriza por su pequeña longitud.
   c) Identifica a un registro.

5. Un puntero:

   a) Es el campo que se coloca en primer lugar.
   b) Es un campo que tiene información sobre la posición de otros registros.
   c) Es un registro que tiene información sobre la posición de otros registros.

6. Un indicador:

   a) Es un registro que indica donde están los datos.
   b) Es un campo que señala el estado de un registro.
   c) Es un campo que indica que a partir de ahí empiezan a colocarse datos.

7. Una diferencia entre los ficheros de acceso secuencial y los de acceso aleatorio es:

   a) Los de acceso directo pueden contener muchos más registros.
   b) Los de acceso aleatorio permiten el acceso a un registro particular con más facilidad.
   c) Los de acceso secuencial disponen de una clave que indica la secuencia de todos los ficheros.

8. Señalar cuál de las dos frases es correcta:

   a) Los ficheros de acceso secuencial sin clave tienen menos longitud que los de acceso secuencial indexados.
   b) Un fichero de acceso secuencial dispone de punteros en sus registros.

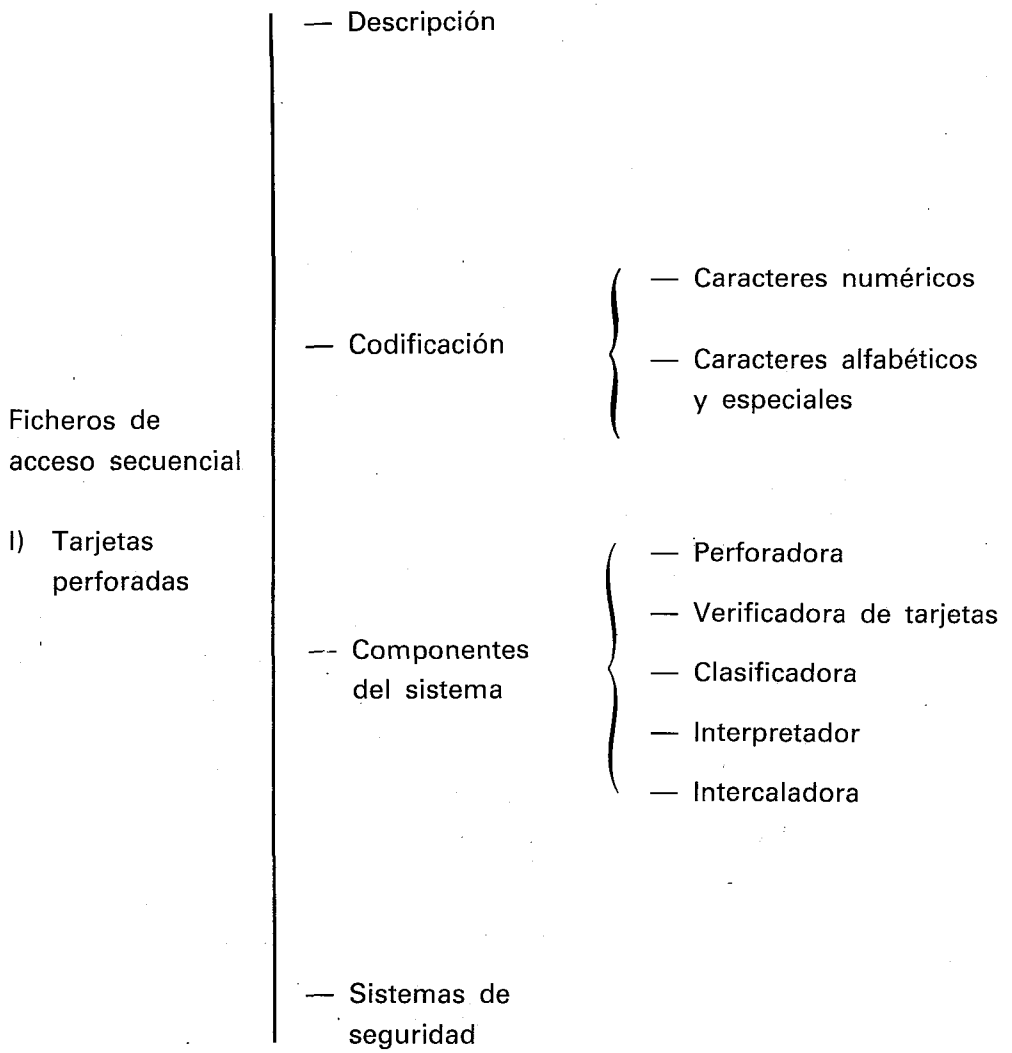
9. Los ficheros de acceso aleatorio indexado y los mapeados se diferencian en:

   a) El indexado permite una más fácil modificación del fichero.
   b) Los indexados tienen un registro con las claves cortas y otro con las claves largas.
   c) Los mapeados precisan de un cálculo para conocer la posición de un registro.

2.
FICHEROS DE ACCESO SECUENCIAL

Tarjetas perforadas

CUADRO RESUMEN DE LA LECCIÓN



2.1. DESCRIPCIÓN FÍSICA

La aplicación de la tarjeta como memoria es prácticamente nula, quedando como principal actividad la de medio de entrada de información al ordenador. La creación de cualquier almacén pasa a menudo por una creación anterior en tarjetas perforadas.

La tarjeta perforada es una cartulina rectangular con una esquina cortada oblicuamente (Fig. 2.1). Posee, en el caso más habitual, 80 columnas y doce filas de perforación. En cada confluencia de línea y columna puede o no existir una perforación, dependiendo del dato que se haya perforado.

Las tarjetas pueden ser de varios colores para facilitar su identificación por el operador que las maneja. Este hecho no tiene nada que ver con el procesamiento de la tarjeta por el ordenador al que sólo le interesa la existencia o no de las perforaciones y el lugar en que están realizadas.

Una aplicación de los diferentes colores es, por ejemplo, el designar un color para el fichero de clientes, otro color para el fichero de artículos, etc.

La esquina cortada también tiene como misión el manejo de la tarjeta por el ordenador. De este modo, colocando siempre todas las tarjetas con la esquina cortada en la posición izquierda, no da lugar a confusión. Como el ordenador toma como primera columna la primera que le llega, si colocáramos la tarjeta boca abajo, la máquina interpretaría la columna 80 como la columna 1, constituyendo un error en el proceso. Colocando las tarjetas con la esquina a la izquierda siempre nos aseguraremos que la primera columna que se va a interpretar correctamente es la columna 1.

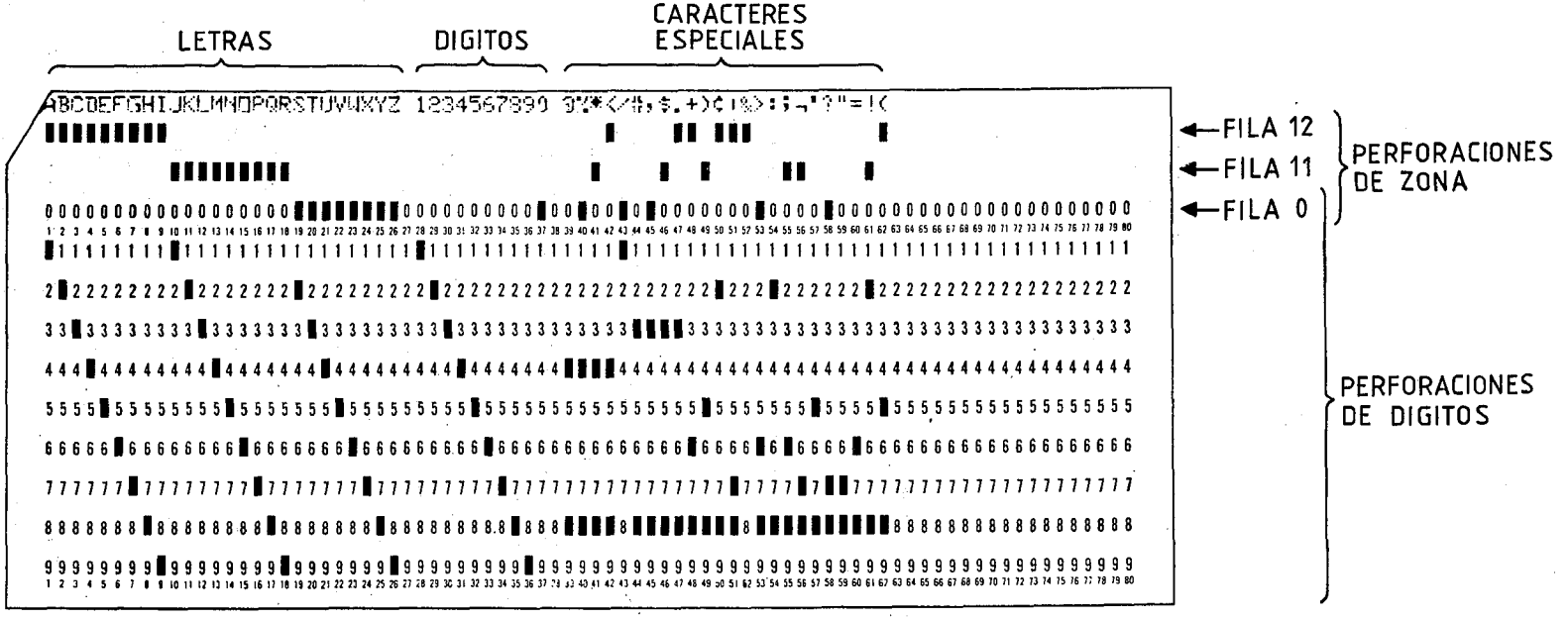

LETRAS
DIGITOS
CARACTERES ESPECIALES
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 ¤'¢*<>#,$.+)¢↑%>:;¬"?"=!(
FILA 12
FILA 11
FILA 0
PERFORACIONES DE ZONA
PERFORACIONES DE DIGITOS

Fig. 2.1

Se desarrollará todo lo siguiente haciendo referencia a este tipo de tarjeta, aunque ya se ha dicho que existen otras formas de tarjetas, por ejemplo las que utilizan 90 o 96 columnas.

Las columnas se dividen en dos áreas (véase la figura 2.1).

— el área de perforación de zona
— el área de perforación de dígitos.

La primera está constituida por las filas 12, 11 y 0 que no tienen impreso en la tarjeta un número que las identifique. Estas filas se localizan en la parte superior de la tarjeta. La fila superior a la 0 es la fila 11 y la que le sigue es la 12.

El área segunda está constituida por las filas numeradas de la 0 a la 9.

2.2. CODIFICACIÓN

La tarjeta puede representar caracteres según la codificación inventada originariamente por Hollerit.

Un caracter se representa por el conjunto de perforaciones existentes en una columna.

2.2.1. Caracteres numéricos

Un dígito se codifica por una perforación en el área de dígitos de la tarjeta. La perforación se hace en la fila correspondiente al dígito que se va a representar (ver figura 2.1).

Cuando se precisen almacenar datos numéricos con más de un dígito se van perforando las columnas siguientes, en las líneas apropiadas. El conjunto de todas las columnas que representan al dato numérico se denomina campo de ese dato.

Si el campo es llamado en el programa por una variable, el ordenador, mediante su dispositivo de entrada, explora cada una de las columnas que corresponden a ese campo. El valor numérico se almacena internamente en el ordenador en una cierta posición de memoria. Si la siguiente tarjeta pertenece al mismo conjunto de datos se explorará de la misma forma, asignando a la variable el dato contenido en el mismo número de columnas.

Una vez acabado de explorar el campo se pasa al siguiente que estará constituido por las siguientes columnas hasta un número determinado definido por el programa. Todo ese conjunto de columnas son interpretadas por el ordenador como el siguiente dato numérico. El proceso se hace subjetivamente hasta cubrir, si es necesario, las 80 columnas de la tarjeta.

2.2.2. Caracteres alfabéticos y especiales

Los dígitos del cero al nueve también pueden representarse de la forma anterior y considerarse como caracteres en lugar de datos numéricos. La única diferencia es que la variable que hace referencia a ese campo sea una variable alfanumérica.

El problema que se considera aquí es el de la representación del resto de caracteres alfanuméricos: las letras y los caracteres especiales.

La representación de estos caracteres se hace igualmente en una columna, colocando dos perforaciones en la misma. Poniendo dichas perforaciones en las diferentes líneas se tienen los diferentes caracteres (Fig. 2.1).

Para representar el alfabeto se hace: una perforación en el área de zona y otra en el área de dígitos para las letras A a la R y una perforación en la fila cero y otra en el área de dígitos para las letras S a Z.

El resto de los caracteres especiales precisan de una, dos o tres perforaciones repartidas en las dos áreas.

Al igual que en el caso de datos numéricos se divide la tarjeta en campos y se tienen variables que representan el valor de estos campos en el ordenador. Cada uno de los campos consta de un conjunto de una o varias columnas.

Como ejemplo de la distribución en campos de una ficha perforada se presenta el siguiente:

Columnas 1-12: nombre empresa
Columnas 13-23: comentario de su producción
Columnas 24-29: año de su fundación
Columnas 30-35: número de empleados
Columnas 36-76: observaciones.

En el ejemplo no se han utilizado todas las columnas. Una vez que se le ha contado al ordenador la disposición y longitud de los campos (número de columnas), emplea este formato para revisar la tarjeta, prescindiendo de las zonas que no han sido definidas.

El ordenador tampoco va a hacer caso de un campo que no esté completamente lleno de caracteres alfanuméricos, el resto de las columnas las considerará como caracter blanco y así las almacenará en el ordenador. Con los datos numéricos no ocurre lo mismo. El campo debe estar completo ya que la mayoría de los equipos interpretan un blanco en un campo numérico como un cero. Este hecho puede producir errores al tratar un campo que guarda un número entero, ya que los ceros añadidos por la derecha por el ordenador causan la confusión del número. Como regla se rellenarán siempre los campos numéricos sin dejar hueco en la derecha.

Naturalmente se considera en el ejemplo que las variables que van a atender a cada uno de los campos son capaces para almacenar los contenidos de dichos campos.

Ejemplo:

Teniendo un campo de 5 columnas, almacenar el número 12. La forma de colocar este número es:

columna 1,2,3: perforar un cero en cada una de ellas
columna 4 : perforar un 1
columna 5 : perforar un 2.

Una forma alternativa sería el colocar un blanco en las columnas 1 a 3 en lugar del cero (esto sería válido para los sistemas que interpretaran los blancos como los ceros).

2.3. COMPONENTES DEL SISTEMA DE FICHAS PERFORADAS

Un sistema de proceso de datos que tenga como forma de entrada de información la ficha perforada debe poseer como mínimo los siguientes elementos:

— La unidad central de proceso.
— Una impresora u otro periférico de salida.
— La perforadora de tarjetas.
— La verificadora de tarjetas.
— La clasificadora de tarjetas.
— La interpretadora.
— La intercaladora.
— La lectora de tarjetas.

Se describen a continuación las máquinas relacionadas directamente con el tratamiento de las tarjetas perforadas.

2.3.1. La perforadora

La perforadora es la máquina encargada de efectuar las perforaciones en la tarjeta dependiendo de los caracteres que quieran registrarse.

Las partes de que consta son:

— Teclado.
— Punzones de perforación.
— Depósito de tarjetas.
— Tambor con tarjeta de programa.
— Depósito receptor de tarjetas.
— Escobillas de lectura.

Este teclado es análogo al de las máquinas de escribir convencionales, además de las teclas para perforar los distintos caracteres especiales que pueden ser grabados en la tarjeta así como los interruptores de control de la máquina. Este teclado controla a los punzones de perforación.

Existen dos depósitos: el de almacén de tarjetas antes de ser grabadas y el que recoge las tarjetas una vez que se han grabado.

Un dispositivo de interés es el tambor con tarjeta de programa. Este dispositivo permite realizar automáticamente el salto y la duplicación de una tarjeta.

Se coloca en el tambor una tarjeta de control donde está codificado el formato de las tarjetas que se van a escribir. Los saltos de un campo a otro se realizan comparando la tarjeta que se está perforando actualmente con la que está colocada en el tambor, una vez se ha actuado una tecla de control. En esa tarjeta de control se pueden codificar las columnas que deben ser copiadas.

La duplicación de una tarjeta consiste en copiar, en una tarjeta virgen, la misma información que se perforó en la tarjeta que se quiere duplicar. Existe también una tecla adecuada para realizar esta función. La tarjeta que está enfrente de las escobillas de lectura va pasando por delante de ellas leyéndose la información que contiene, a la vez, se va perforando la información leída en la tarjeta que se tiene en la zona de perforación (el lugar donde están los punzones de perforación).

La información que se escribe en las tarjetas se obtiene de una serie de documentos, donde están escritos todos los datos que se van a necesitar para la creación del fichero. Se llama *documento base* al conjunto de esos papeles de donde se obtiene la información. Por ejemplo si se quiere hacer una estadística de las ventas de una empresa cada mes, el documento base podría estar constituido por las copias de las facturas que se mandaron a los distintos clientes a lo largo de ese mes.

2.3.2. Verificadora de tarjetas

El paso siguiente al proceso de perforación es llamado la *verificación.*

Su objetivo es detectar los errores que puedan existir en el proceso de perforación, porque se haya tecleado una tecla en lugar de otra.

La máquina que realiza esta acción se denomina *verificadora.* Su apariencia es la de una perforadora, pero no perfora. El operador de la máquina pulsa toda la información de nuevo, a partir del documento base, en el teclado de la máquina. La verificadora va comparando las perforaciones existentes en la tarjeta con las que deberían aparecer según la tecla que se acaba de pulsar. En

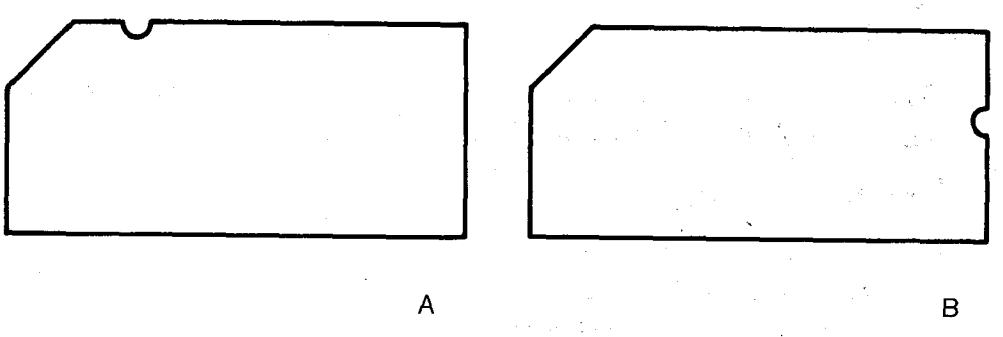


Fig. 2.2. En A) La verificadora hace una muesca en el borde superior de la tarjeta cuando en una columna hay una perforación incorrecta.
En B) se muestra la muesca lateral que hace la verificadora cuando todas las perforaciones son correctas.

caso de que ambas coincidan el proceso sigue adelante. Cuando la máquina se encuentra con una falta de coincidencia, avisa al operador de que hay un error. El operador vuelve a pulsar el caracter y así hasta tres veces consecutivas si se sigue repitiendo el error. En caso de que aún no coincidan las perforaciones y la tecla pulsada, la verificadora marca esa columna con una muesca, queriendo indicar que en esa columna existe un error. La razón de que el operador de la máquina tenga que pulsar otras dos veces el caracter es para evitar un error en la verificación, por ejemplo que el operador entienda algún caracter del documento base erróneamente y que en la perforación estuviese bien perforado.

Este proceso se repite para todas las columnas de la tarjeta. Si en la tarjeta no ha existido error queda marcada por la verificadora con una muesca en la parte derecha.

Los errores que no pueden detectarse son aquellos en los que exista coincidencia. Es el caso, por ejemplo, de que tanto el operador de la perforadora como el de la verificadora hubieran entendido un 5 del documento base como una S, y hubieran perforado y verificado ambos una S.

Para evitar estas situaciones es conveniente que la persona que perfore y verifique no sean la misma. De este modo es difícil que caigan en los mismos errores.

2.3.3. Clasificadora

El dispositivo llamado clasificadora se utiliza para obtener una ordenación de las tarjetas distinta a la secuencia en que se perforaron.

La máquina posee:

— Una estación de entrada.
— Una lectora de columnas.
— Varias estaciones de salida o receptores.

Las tarjetas se colocan en la estación de entrada y se van disponiendo sucesivamente delante de la lectora. La lectora reconoce el caracter que está escrito en esa columna y manda la tarjeta a una de las estaciones receptoras, dependiendo del valor o tipo del caracter considerado.

Las tarjetas se clasifican por el valor de una columna. Si se quieren clasificar por el valor de un campo se deben hacer tantas pasadas por la

clasificadora como columnas tenga ese campo, puesto que en cada una de las pasadas se clasifica una sola columna.

La utilización de la clasificadora es necesaria cuando se quiere tratar por ordenador el conjunto de tarjetas pero ordenadas según un campo diferente del que se ordenaron originariamente.

Ejemplo:
El formato de una tarjeta está formado por los siguientes campos:

Campo 1: Nombre empresa (10 columnas).
Campo 2: Tipo de artículos que produce (20 columnas).
Campo 3: Año de fundación de la empresa (4 columnas).
Campo 4: Número de empleados (5 columnas).

Si en una aplicación se quiere conocer las empresas que aparecieron en los diferentes años, será conveniente una clasificación por años y no la actual clasificación por empresas que existe, en el conjunto de fichas. Naturalmente puesto que el campo donde se indica el año tiene 4 columnas se necesitarán 4 pasadas de las fichas por la clasificadora.

En la primera pasada las tarjetas se clasificarán según la primera columna del campo de año. Suponiendo que todas las empresas se han formado a partir del año 1000 se tendrán las tarjetas clasificadas en el mismo receptor, es decir, tal como estaban, de modo que esta operación puede ahorrarse.

Seguidamente se pasa a la siguiente columna, donde se investigará la segunda cifra del año, separándose las tarjetas en como máximo 10 receptores de la estación de salida (la segunda cifra del año no puede tener más que 10 valores diferentes, los dígitos del cero al nueve). Naturalmente también será raro el caso de las empresas que se hayan formado antes del 1700, lo que limita a 3 el número de receptores de salida utilizados.

Para cada uno de los tres conjuntos diferentes se hace una pasada por la clasificadora, comprobando ahora el valor que tiene la tercera columna. El proceso acabaría al pasar cada uno de los grupos que nos han salido de la verificación de la tercera columna.

2.3.4. La interpretadora

La misión que se le encomienda a la máquina conocida como *interpretadora* es que lea una tarjeta perforada y que escriba en algún lugar de la misma el caracter que se ha representado en cada una de las columnas perforadas.

La mayoría de las perforadoras poseen esta función de intérprete, de modo que a la vez que se va perforando la tarjeta se va escribiendo su contenido en la parte superior.

La diferencia entre la interpretadora y la impresión de los caracteres por la perforadora es que en aquella la legibilidad de lo escrito es mayor, puede escribirse en cualquier lugar de la tarjeta y se tienen control para no escribir sobre algunas columnas.

2.3.5. La intercaladora

La función de la máquina intercaladora es similar a la de la clasificadora pero más potente.

La intercaladora posee dos dispositivos de entrada de tarjetas y varios dispositivos de salida de tarjetas.

Las funciones principales de las intercaladoras son:

— Fusionar dos conjuntos en uno solo.
— Emparejamiento.
— Ambas operaciones, fusión y emparejamiento a la vez.

La fusión de dos conjuntos en uno solo consiste en conseguir un conjunto único a partir de dos en una secuencia, alfabética o numérica, definida. Esta secuencia es controlada por uno de los campos de la tarjeta.

El emparejamiento consiste en colocar juntas las tarjetas de ambos dispositivos de entrada que tengan el mismo dato en el campo que se utiliza para el control de la máquina.

Ejemplo:
El proceso de fusión se ilustra con el ejemplo siguiente:

Se tienen dos conjuntos de tarjetas. El número que tiene el campo por el que se va a clasificar es:

Primer conjunto	*Segundo conjunto*
1153	1057
1153	1153
1259	1257
1351	1257
1576	1351

El conjunto formado después de pasar por la intercaladora es:

1057 (del segundo conjunto)
1153 (del primero)
1153 (del primero)
1153 (del segundo)
1257 (del segundo)
1257 (del segundo)
1259 (del primero)
1351 (del primero)
1351 (del segundo)
1576 (del primero)

Un ejemplo de la función de emparejamiento es:

Se tienen dos conjuntos que se quieren emparejar. Las tarjetas emparejadas se colocarán en la salida 1, las que no queden emparejadas del primer conjunto de entrada se colocarán en la salida 2 y las que no sean emparejadas de la segunda entrada se colocarán en la salida 3.

Los conjuntos usados son los mismos del problema anterior.

Las salidas 1, 2 y 3 dan respectivamente:

Salida 1:
1153 (del primer conjunto)
1153 (del primer conjunto)
1153 (del segundo conjunto)
1351 (del primer conjunto)
1351 (del segundo conjunto)

Salida 2:
1259 (del primer conjunto)
1576 (del primer conjunto)

Salida 3:
1057 (del segundo conjunto)
1257 (del segundo conjunto)
1257 (del segundo conjunto)

2.4. SISTEMAS DE SEGURIDAD

Las informaciones grabadas, en cualquier sistema de almacenamiento deben preservarse de cualquier eventualidad que las destruyera. En caso de que esto sucediera se debería poder recuperar la información almacenada de un modo u otro.

Se llama sistema de seguridad al modo de hacer posible esta recuperación de información.

En las tarjetas, el único sistema de seguridad es la copia de las tarjetas perforadas. Las tarjetas se gastan con el tiempo de uso y el ordenador puede tener fallos en la interpretación de una tarjeta gastada, además, volver a copiar todo el conjunto de tarjetas permite tener otro fichero alternativo que debería ser guardado en un lugar diferente al que se guarda el fichero original, disminuyendo así la probabilidad de que ocurra algún percance en los dos lugares a la vez.

Ejemplo en un sistema de tarjetas:

Se desean diseñar los recibos de los pedidos servidos a los clientes mediante un sistema computador por tarjetas perforadas.

Se procede a perforar y verificar un fichero de pedidos, siendo el documento base los pedidos de los clientes.

Un registro, correspondiente a un pedido y representado por una tarjeta, tendrá los siguientes campos:

— Número de pedido.
— Número de código del cliente que lo pide.
— Cantidad pedida.
— Cantidad entregada (rellenado por el almacén).
— Número de código del artículo pedido.
— Descripción del artículo.
— Precio unitario.

Además de este fichero con los pedidos, el ordenador precisará, para rellenar la cabecera del recibo, los datos del cliente. Estos datos se sacarán del fichero de clientes. Este fichero tiene el formato siguiente:

— Número de código del cliente.
— Nombre del cliente.
— Dirección del cliente (englobando en sucesivos campos: teléfono, télex etc.).

Los dos ficheros mencionados constituyen los datos de entrada al ordenador. Se supone en la solución del ejemplo que este fichero de clientes está constituido con anterioridad y que está clasificado por número de cliente. De este modo no es necesario perforarlo ni clasificarlo.

El siguiente proceso a realizar es la clasificación del fichero de pedidos creado. El campo de control para dicha clasificación será el de número de código del cliente. Ese es el campo que debe ser revisado por la clasificadora.

Después de la clasificación se pasa a la intercalación del fichero de clientes y el de pedidos. El campo de control vuelve a ser el código del cliente.

Al finalizar este proceso se tiene un solo conjunto de fichas por grupos.

La primera ficha del grupo corresponde a los datos del cliente y las siguientes a todos los pedidos realizados por él. Con este conjunto de fichas el ordenador ya puede trabajar.

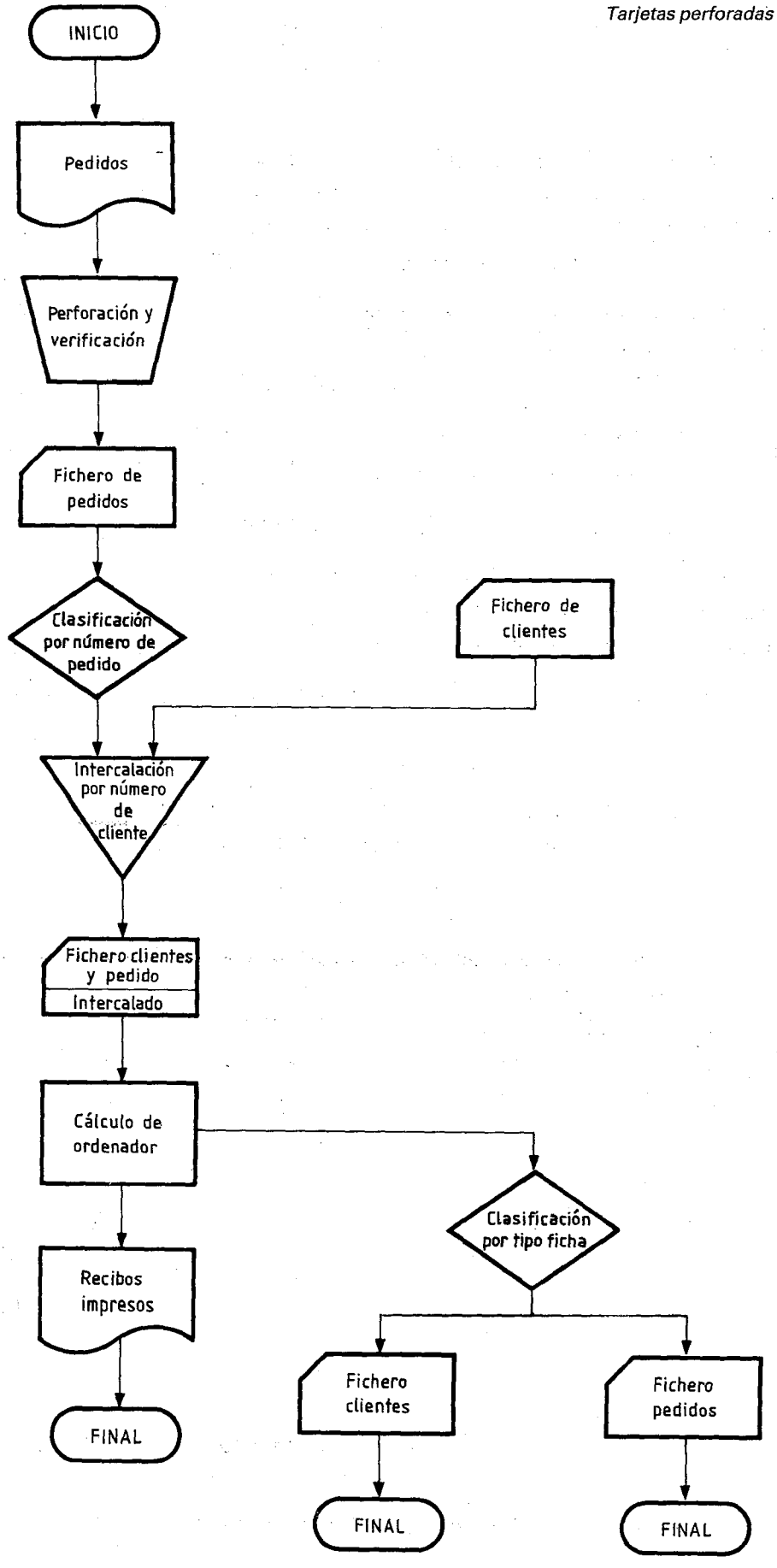
Los recibos obtenidos por el ordenador tendrán el siguiente formato:

— Cabecera: nombre y dirección del cliente.
— En cada una de las líneas de impresión del recibo: cantidad pedida, cantidad entregada, nombre del artículo, precio unitario, precio total.
— Como última línea del recibo: cantidad total a pagar.

Al finalizar la operación se deberá hacer uso de la clasificacadora para separar los dos ficheros: el de clientes y el de pedidos. El campo de control será uno que identifique el tipo de tarjeta (por ejemplo 01 significaría tarjeta de cliente, 02 tarjeta de pedidos) y que no se mencionó anteriormente.

El proceso se muestra en la figura 2.3.

Fig. 2.3. Diagrama de flujo del ejemplo anterior.


INICIO
Pedidos
Perforación y verificación
Fichero de pedidos
Clasificación por número de pedido
Fichero de clientes
Intercalación por número de cliente
Fichero clientes y pedido
Intercalado
Cálculo de ordenador
Clasificación por tipo ficha
Recibos impresos
Fichero clientes
Fichero pedidos
FINAL
FINAL
FINAL

EJERCICIOS

1. En una tarjeta, un caracter viene representado por:

   a) Una columna.
   b) Varias columnas.
   c) No se puede representar.

2. ¿Cuántos campos podría tener como máximo una tarjeta?

   a) 4 – 79.
   b) 80 – 96.
   c) 5 – 95.

3. Para representar una letra se hacen

   a) Dos perforaciones en dígito.
   b) Una perforación en dígito y otra en zona.
   c) Dos perforaciones de zona y una de dígito.

4. Un documento base es el conjunto de papeles de donde se obtiene la información para la creación de un Fichero.

5. El primer paso que deben sufrir las tarjetas es:

   a) Verificar.
   b) Interpretar.
   c) Perforar.

6. La misión de fusionar dos conjuntos de tarjetas en uno sólo la efectúa la

   a) Clasificadora.
   b) Intercaladora.
   c) Verificadora.

7. Llamamos sistema de seguridad Al modo de hacer posible la recuperación de información

3.

FICHEROS DE ACCESO SECUENCIAL

Cintas magnéticas

CUADRO RESUMEN DE LA LECCIÓN

- Ficheros de acceso secuencial
  - Cintas magnéticas
    - Descripción física
    - Lectura y escritura
    - Descripción lógica
    - Grabación de ficheros
    - Sistemas de seguridad
      - Ciclo del «abuelo»
      - Uso del anillo protector
      - Uso de etiquetas electrónicas
    - Concepto de bloqueo
      - Disminución del tiempo de acceso
      - Aumento de la capacidad
    - Sistemas de proceso de datos con cinta magnética
    - Otros sistemas de almacenamiento
      - Tambores magnéticos
      - Célula de datos

3.1. DESCRIPCIÓN FÍSICA

La cinta magnética es el soporte de almacenamiento de datos con una relación de coste más pequeña.

Su aspecto físico es el de una cinta de plástico recubierta de material magnético y que está enrollada en un carrete para facilitar su manejo. Para hacer uso de la cinta se necesita un dispositivo (bobinador, unidad de cinta) que intercambie la información entre el ordenador y la cinta. Este bobinador tiene como misión efectuar los movimientos de la cinta y escribir o leer los datos, según las órdenes del programa.

El almacenamiento de la información se basa en los mismos principios que para el disco pues ambos son almacenes magnéticos.

Se denomina *pista* a cada una de las zonas longitudinales en que puede considerarse dividido el ancho de la cinta a efectos de grabación.

La grabación de un carácter se hace habitualmente con un código de 9 bits. Cada uno de ellos se graba a lo ancho de la cinta, en cada una de las nueve pistas.

El código comentado de 9 bits dispone de 8 de ellos para la representación del carácter y el noveno restante para la exploración de errores; denominándose *bit de paridad.*

3.2. LECTURA Y ESCRITURA

La cinta es un sistema de almacenamiento reutilizable. La información puede escribirse y borrarse un número ilimitado de veces, pero, a diferencia del disco, no precisa de un control como para el posicionado de las cabezas.

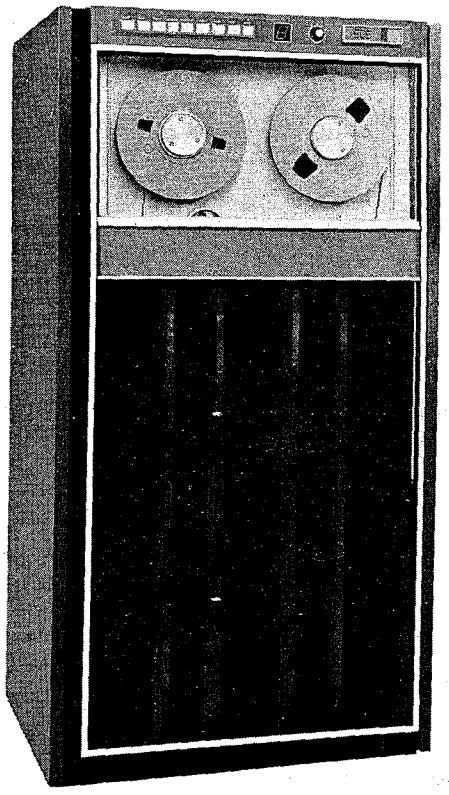
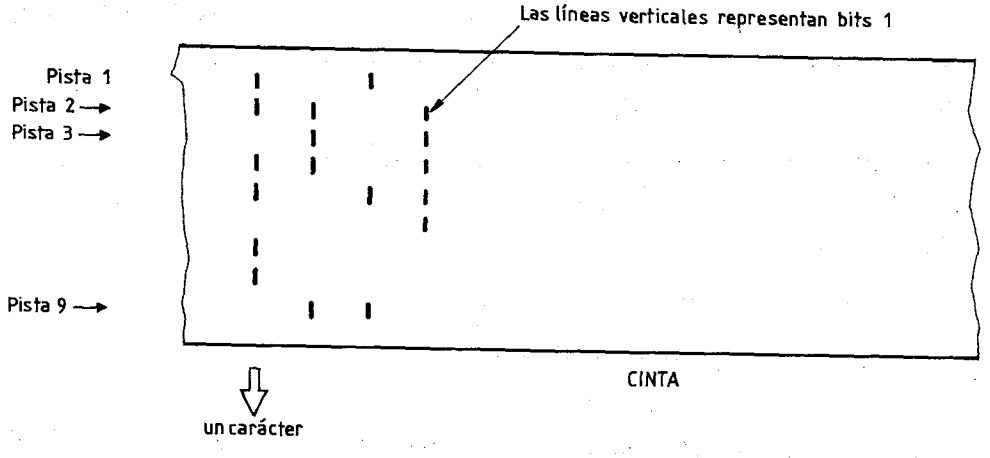


Fig. 3.1. Cinta magnética y unidad de lectura y grabación de cintas magnéticas.



La cinta pasa por delante de la cabeza de lecura o escritura siendo ésta fija. Esta disposición de una información detrás de otra no permite que el sistema vaya directamente al lugar donde tiene que leer o escribir porque no existen zonas representadas por una dirección como en el disco.

Si quiere leer una información en una lista tiene que pasar por todas las anteriores hasta conseguir encontrar la información que se busca.

Por lo tanto no existen tampoco mecanismos de control de las cabezas ni tiempos de acceso, en el sentido dado en el disco.

3.3. DESCRIPCIÓN LÓGICA

La grabación de un conjunto de caracteres constituye un registro. Este registro será dividido en varios campos por el ordenador, sin hacerse ninguna diferencia entre ellos en la grabación. Entre cada uno de los registros se dispone una zona sin grabar para su separación. La denominación habitual de esta zona sin grabación es *gap* en la literatura inglesa.

3.4. GRABACIÓN DE FICHEROS

Cuando interesa grabar un conjunto de registros para formar un fichero, se colocan en la cinta, además de los estrictos datos y zonas sin grabar, unas informaciones llamadas *etiquetas.*

Estas etiquetas tienen como misión informar del contenido del fichero que vendrá a continuación, y algunos datos sobre la cinta.

Los datos que se colocan y la secuencia en que se ponen es:

— ETIQUETA de volumen.
— *Gap*
— Etiqueta de cabecera.
— *Gap*
— Los datos que se han de almacenar.
— *Gap*
— Etiqueta recordatorio y de fin de fichero.

Todas las etiquetas se comparan con las fichas de control del programa para verificar que todo concuerda y que no existen confusiones entre un fichero y otro.

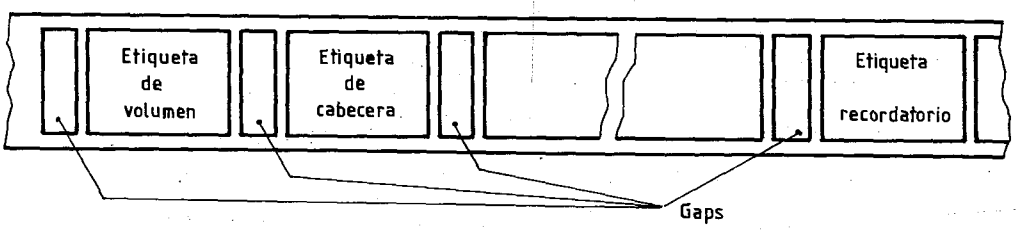


Fig. 3.2. Detalle de una cinta magnética.

La etiqueta de volumen contiene la información sobre el número de carrete que se está usando.

La etiqueta de cabecera contiene datos referentes al fichero. Estos son: el nombre que se le ha dado al fichero, la fecha que se creó el fichero, la fecha límite de conservación del mismo.

En este momento se crea el fichero y se graban todos los registros que contiene separados por convenientes «gaps».

Una vez se han colocado todos los datos, se coloca la etiqueta recordatorio que contiene la misma información de la etiqueta cabecera y de la cantidad de bloques de datos que tiene el fichero.

A continuación de este fichero puede colocarse otro con el mismo esquema.

3.5. SISTEMAS DE SEGURIDAD

Para evitar la destrucción de la información grabada en una cinta se dispone de varios sistemas de seguridad:

— Ciclo del «abuelo».
— Uso del anillo protector.
— Etiquetas electrónicas.

3.5.1. Ciclo del «abuelo»

El método consiste en que siempre se deben mantener el «padre» y el «abuelo» de la cinta actual. Estas palabras nombran a las dos cintas anteriores de las que se obtuvo la actual. La cinta actual se obtiene de la cinta «padre» y de correcciones a esa cinta si los datos del fichero han variado. Esta cinta padre se obtuvo en su día de la cinta «abuelo» y de correcciones a aquella cinta «abuelo». Estas cintas «abuelo» y «padre» deben ser almacenadas en lugares distintos a los que se almacena la cinta actual con el fin de aumentar la seguridad.

3.5.2. Uso del anillo protector

Este anillo consiste en una pieza en forma de aro que se coloca en la cinta que se va a grabar. Si este anillo no está colocado, no se puede escribir en la cinta, porque el dispositivo de grabación está bloqueado. De este modo cuando en una cinta se ha grabado un fichero, se le quita el anillo y queda protegida contra una eventual escritura.

3.5.3. Uso de las etiquetas electrónicas

Las etiquetas electrónicas son todos aquellos registros etiqueta escritos en la cinta que dan información sobre la misma y sobre el fichero. Estos registros no constituyen los datos del fichero.

El modo de evitar errores es la comparación de los contenidos de dichas etiquetas y de las informaciones de control que se incluyen en el programa. Estas informaciones deben de coincidir, pues en caso contrario significaría que el ordenador está considerando un fichero diferente al del programa, lo que evidentemente es un error.

Para verificar la corrección de lo que se ha grabado en la cinta existen diversos mecanismos. Además del bit de paridad para cada caracter grabado, existe un caracter longitudinal de verificación que se coloca al final de la grabación de cada registro.

Con el bit de paridad y con el caracter de verificación se puede comprobar la existencia de un error en cada uno de los bits. El método para calcular ese caracter de verificación suele ser también el obtener la paridad en cada una de las pistas.

En la figura 3.3. se ve cómo el bit de paridad controla verticalmente y el carácter de verificación (o carácter de paridad) controla horizontalmente.

Para mejorar la comprensión se supondrá que existe un sistema de coordenadas para numerar los bits y cuyo origen está en el extremo superior izquierdo.

Se supone que ha habido un error en el bit de la columna 2, fila 3. Este bit se ha cambiado de 1 (valor correcto) a 0 (incorrecto).

El bit de paridad toma un valor tal que con su presencia hace el número total de unos impar (si se ha tomado paridad impar).

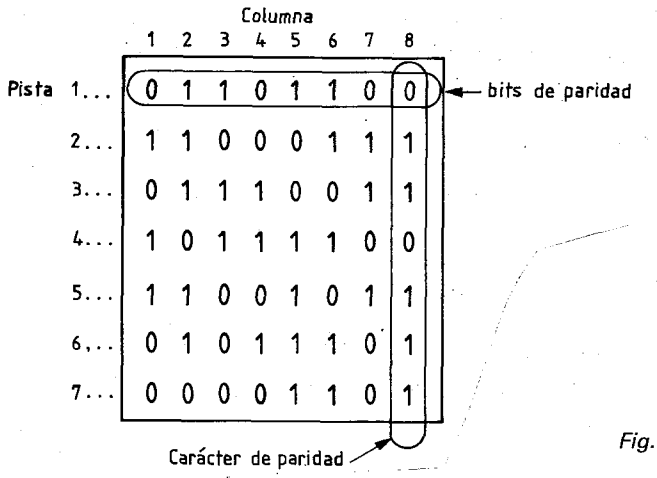


Fig. 3.3. Bit y carácter de paridad.

En el caso presentado el bit de paridad de la columna 2 vale 1 porque (sin errores) el número total de unos existentes en el carácter es cuatro y con el bit de paridad el número de unos pasa a ser cinco, un valor impar como se quería.

Del mismo modo el bit de paridad para la columna 3 vale 1 porque el número total de unos en ella es 2, y con el bit de paridad habrán 3.

Para descubrir el error se comprueban los bits de paridad. Después de producirse el error se ve que la paridad impar no se conserva ni en la fila 3, ni en la columna 2, existiendo respectivamente 5 y 3 unos.

Cualquier sistema de corrección tiene un límite en cuanto al número de errores que puede corregir. Así, si se producen dos errores (o un número par de errores) en una fila el bit del carácter (horizontal) no detectará error, pero sí lo harán los bit de paridad (vertical). El sistema habrá detectado el error pero no podrá conocer dónde se halla (sólo sabe la columna en este caso).

3.6. CONCEPTO DE BLOQUEO

La capacidad de almacenamiento es una característica que siempre se intenta aumentar. En un sistema, colocar más información en un lugar más reducido disminuye el coste del almacenamiento. El concepto de bloqueo ayuda a aumentar esta capacidad.

Se denomina *bloqueo de registros* a la reunión de varios registros lógicos en el espacio de un registro físico. Este concepto puede ser aplicado a cualquier soporte de almacenamiento de los comentados, tanto discos como cintas. Se debe distinguir entre registro físico y registro lógico y cuál de ellos actúa en cada momento, para tener una comprensión completa del concepto.

Si por ejemplo, se colocan 3 registros entre los «gaps» de una cinta, se dice que se ha hecho un bloqueo o que los registros forman un bloque. Al número de registros que se coloca entre los dos «gaps» se le llama *factor de bloqueo.* En este caso el factor de bloqueo sería igual a 3. El bloqueo se hace controlado por el programa. Este envía al soporte los registros en bloques de tres en tres (siguiendo con el ejemplo). Como se ve, existen tres registros lógicos, con informaciones diferentes y que son tratados por el ordenador como registros diferentes.

En cambio el dispositivo de almacenamiento considera un sólo registro (registro físico) que es el que está colocado entre los «gaps» y que será tratado como unidad por el controlador del soporte (en el caso de la cinta será el bobinador).

Las ventajas que presenta el bloqueo son un menor tiempo de acceso y un incremento en la capacidad.

3.6.1. Disminución del tiempo de acceso

Sea un fichero grabado en una cinta que contiene 100.000 registros. Si la grabación se ha efectuado sin bloqueo la cinta contendrá el mismo número de «gaps», es decir, 100.000. Al ir efectuándose el proceso de lectura la cinta se

para en cada gap un corto lapso de tiempo para volver a arrancar y pasar al siguiente registro. Se supone que la velocidad de lectura es de 100.000 bytes/segundo. El tiempo que se tarda en leer este fichero será:

Tiempo de lectura de 100.000 registros + tiempo de parada y comienzo en 100.000 «gaps».

Poniendo resultados particulares:

[(100 bytes/registro × 100.000 registros) / 100.000 bytes/segundo] +
+ 100.000 gaps × 8/1000 segundos/gap

100 segundos (datos) + 800 segundos (gaps)

(se han considerado registros de 100 bytes cada uno)

Si se hubiera hecho un bloqueo con un factor de 10, el tiempo de transferencia de los datos no hubiera variado porque siguen siendo 100.000 × 100 bytes los que hay que leer, pero el tiempo de paro y comienzo en los «gaps» se hubiera reducido 10 veces porque ahora existen 10 veces menos «gaps».

El tiempo total resultaría:

100 segundos + 80 segundos = 180 segundos

Lo que supone sólo el 20 % del tiempo anterior.

3.6.2. Aumento de la capacidad

Puesto que el «gap» ocupa lugar en la cinta, si se disminuye el número de gaps se habrá acortado la longitud usada para éstos y quedando como longitud útil para el almacenamiento de otros datos.

Se graba un fichero de 10.000 registros, cada uno de ellos de 80 bytes, en una cinta con una densidad de grabación de 800 bytes/pulgada. Haciendo la grabación sin bloqueo (o lo que es lo mismo, con bloqueo unidad) la longitud de cinta necesaria sería:

(80 bytes/registro × 10.000 registros) / 800 bytes/pulgada =
10.000 pulgadas para los datos.

Si se supone que cada «gap» ocupa 0.6 pulgadas en cinta, los 10.000 «gaps» ocuparán un espacio en la cinta de.

0.6 pulgada/gap × 10.000 gaps = 6000 pulgadas.

Si ahora se hace la operación con un factor de bloqueo de 10. La longitud de la cinta para la grabación de los datos sigue siendo la misma porque el número de datos no ha variado, pero la longitud de cinta para soportar a los «gaps» queda reducida en 10 veces:

600 pulgadas (6000/10)

La longitud del fichero es ahora tan sólo el 66 % de la anterior.

Las ventajas del bloqueo en los conceptos de tiempo y capacidad han quedado manifiestas, pero no se deben olvidar los inconvenientes. El controlador del soporte va a trabajar con registros físicos. Eso quiere decir que cuando se le mande leer un registro, el controlador va a entregar al dispositivo de entrada al ordenador un bloque de 10 registros (se ha supuesto un factor de bloqueo 10). La condición que se pone es que este bloque pueda ser almacenado por el dispositivo de entrada (análogamente ocurriría con un dispositivo de escritura, el cual esperaría a que tuvieran los 10 registros que le proporciona el ordenador para pasarlos a la cinta). Vista esta cuestión, dependerá de cada sistema de proceso de datos el elegir el factor de bloqueo más conveniente y que no dé errores. La elección se hace, naturalmente, en base a los componentes del sistema.

3.7. SISTEMAS DE P.D. CON CINTA MAGNÉTICA

Los componentes de un sistema de proceso de datos basado en cinta magnética son generalmente:

— Unidad central de proceso.
— Unidades de perforación, verificación y lectura de fichas.
— Varias cintas y unidades de cinta (habitualmente 4).
— Una impresora o terminal de salida.

Otra vez se necesita la presencia de los periféricos de manejo de tarjetas o fichas perforadas puesto que la escritura directa en cinta magnética de un fichero no es posible. Debe seguirse un *proceso tarjeta-cinta* al igual que se

hacía en la creación de ficheros en disco. La diferencia existente entre aquel caso y éste de las cintas, es que en éstas no se debe calcular por programa la dirección en la que se ha de colocar la información, puesto que la cinta no tiene direcciones, sino que la información se escribe secuencialmente.

Por ejemplo, se desea poner al día un fichero que está escrito en cinta. Se debe crear un nuevo fichero, también en cinta, a partir del fichero antiguo y de las correcciones que se hayan producido.

La primera operación que se debe realizar es perforar en tarjetas todas las correcciones que deben modificar el fichero. Una vez perforadas y verificadas, se crea un fichero en cinta con todas las correcciones. Este fichero se obtiene de las tarjetas perforadas.

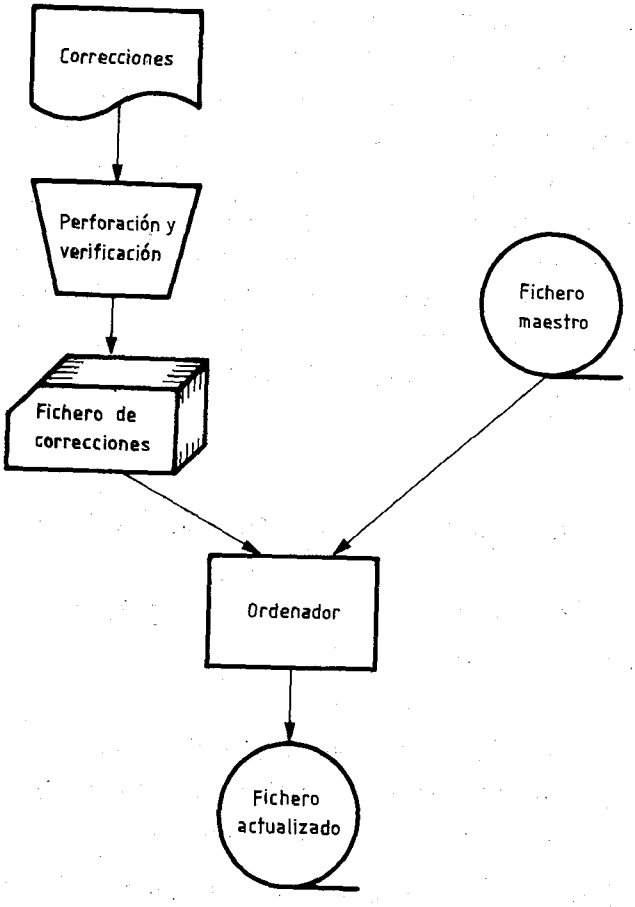


Fig. 3.4. Organigrama del proceso del tratamiento de un problema con cinta magnética.

A continuación se crea un fichero nuevo a partir de éste de correcciones en el que los registros estén ordenados en la misma secuencia que lo están los registros del fichero que se debe modificar.

A partir del fichero antiguo y de las correcciones se obtiene la cinta con el fichero actualizado.

Un organigrama del proceso se muestra en la figura 3.4.

3.8. OTROS SISTEMAS DE ALMACENAMIENTO

Se considerarán dos soportes de almacenamiento no tan ampliamente usados como las cintas magnéticas. Se desarrollan aquí para reforzar la idea de que, aunque siempre se hable de discos o cintas magnéticas, existen otros sistemas.

Se comentan dos tipos:

— Los tambores magnéticos.
— Las células de datos (data cells).

3.8.1. Los tambores magnéticos

El tambor es un soporte de almacenamiento reutilizable. La grabación y lectura de la información se efectúa magnéticamente.

Su forma es cilíndrica y la información se graba en la parte externa.

Esta superficie se considera dividida, a efectos lógicos, en pistas. Cada una de ellas tiene una dirección que la representa. En la unidad de manejo del tambor se colocan un conjunto de cabezas de lectura y escritura igual al número de pistas en el tambor.

La cabeza, fija, va obteniendo la información o escribiéndola por la rotación del cilindro en torno a su eje.

3.8.2. Las células de datos

Este es también un soporte reutilizable que se graba y lee magnéticamente. Consiste en una serie de tiras magnéticas donde se graba la información en pistas. Estas tiras están guardadas en un contenedor de forma cilíndrica. Cuando se quiere usar una de las células se elige el contenedor, una vez elegido

se pone a girar hasta que la célula queda enfrente del mecanismo de lectura/escritura. En ese momento la célula es atrapada por un dispositivo que la levanta y la saca del contenedor.

La coloca forrando un tambor que lleva el mecanismo de lectura o escritura. Se hace la operación (lectura o escritura) y se devuelve a su contenedor.

De la descripción puede deducirse que el sistema es bastante lento de acceso. La ventaja que tiene es la enorme cantidad de datos que puede almacenar y su bajo coste. Fácilmente una célula de datos puede almacenar 400 millones de bits de datos. Cifra que se puede comparar con los 29 millones de datos que puede almacenar un conjunto de discos.

EJERCICIOS

1. La relación de coste de las cintas magnéticas es:

   a) La mayor.
   b) La menor.
   c) Normal.

2. Los caracteres se graban normalmente en las cintas magnéticas en códigos de:

   a) 7 bits.
   b) 8 bits.
   c) 9 bits.

3. Se denomina pista a cada una de las zonas longitudinales en que se puede dividir la cinta a efectos de información.

4. La cabeza de lectura o escritura:

   a) Es fija.
   b) Precisa de control de posicionado.
   c) Debe cambiarse con frecuencia.

5. El concepto de *tiempo de acceso* con cintas magnéticas:

   a) Es más grande que con discos.
   b) Es el tiempo que se tarda en leer un registro.
   c) Este concepto no existe en las cintas magnéticas.

6. La separación de los campos de un registro grabado en cinta magnética:

   a) Se logra con zonas *gap*.
   b) No están diferenciados.
   c) Se logra con etiquetas.

7. La información del contenido de un fichero en cinta magnética se logra:

   a) Mediante una etiqueta.
   b) Mediante la zona *gap*.
   c) Leyendo los registros que lo componen.

8. Las etiquetas se comparan con los fichas de control de programas.

9. En una cinta magnética se puede:

   a) Grabar sólo un fichero.
   b) Grabar varios ficheros.
   c) Acceder directamente a cualquier fichero.

10. El anillo protector se coloca:

   a) En una cinta magnética a leer.
   b) A continuación de las etiquetas.
   c) En una cinta magnética a grabar.

11. El bloqueo permite:

   a) Ordenar los registros.
   b) Reducir el coste de almacenamiento.
   c) Comprimir cada registro.

12. El factor de bloqueo es:

   a) El número de registros entre dos gaps.
   b) Un control de la cinta magnética.
   c) El número de bloques que contiene un fichero.

13. El tambor magnético es un soporte de almacenamiento:

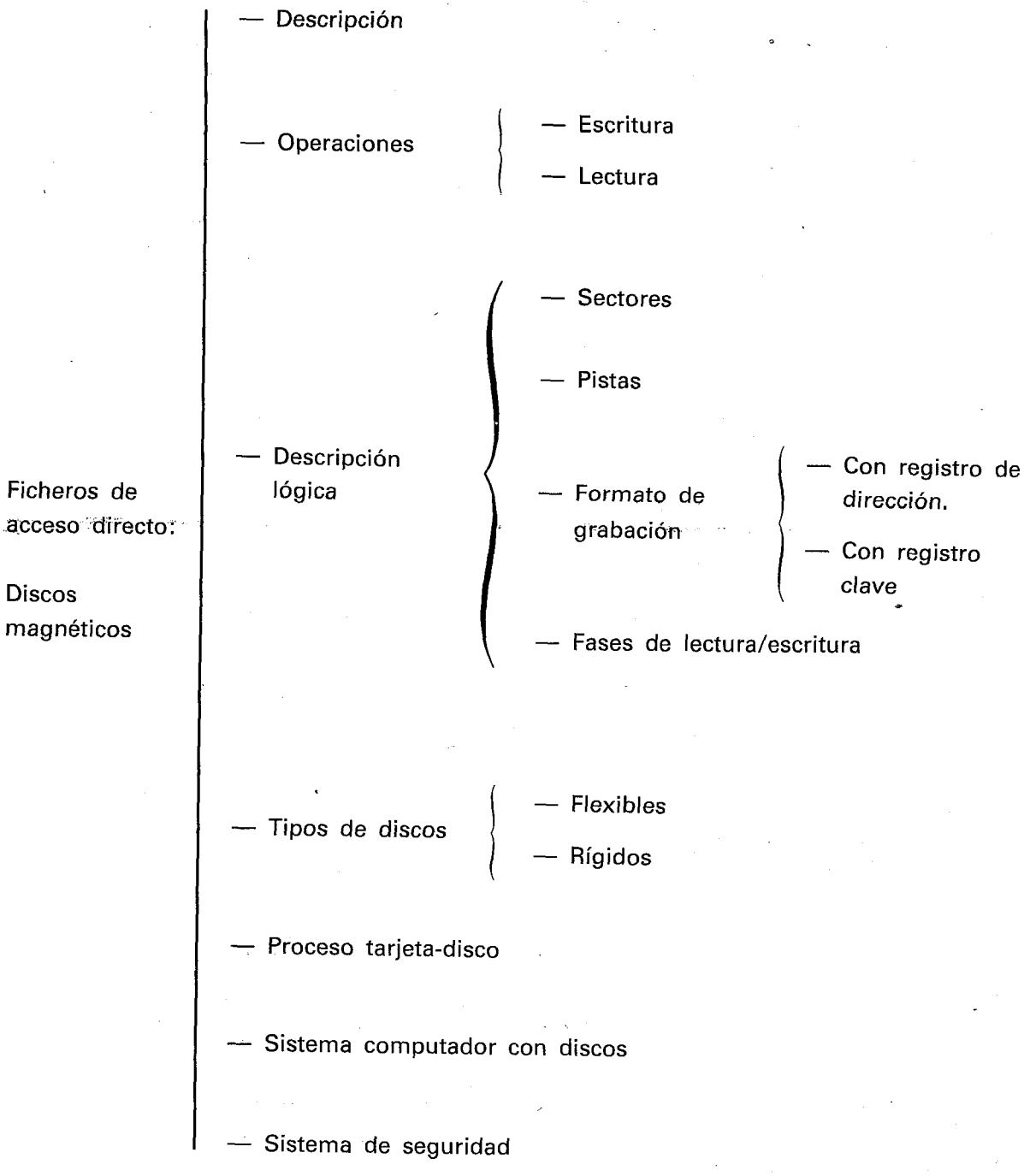
   a) Con una sola cabeza de lectura y escritura.
   b) De acceso secuencial.
   c) Reutilizable.

4.

FICHEROS DE ACCESO DIRECTO

Discos magnéticos

CUADRO RESUMEN DE LA LECCIÓN



4.1. DESCRIPCIÓN

El disco magnético es un dispositivo de almacenamiento de información. La forma usual de utilización de los discos es en bloques de 11 montados sobre un soporte único (dispack).

Un disco es una placa circular de plástico, más o menos rígido según el tipo de plástico y que se ha recubierto de una sustancia con propiedades magnéticas. Esta sustancia está formada por partículas alargadas, en forma de agujas, de una aleación de hierro.

En la disposición habitual de 11 discos son aprovechables para la grabación 20 caras (la primera superior y la última inferior del conjunto no se aprovechan). A cada una de estas caras le corresponde una cabeza de lectura y otra de escritura. Así, en total, existen 20 conjuntos de cabezales.

Estos conjuntos de cabezas están colocados en un dispositivo general denominado *unidad de discos* que contiene, a su vez, los motores y soportes para el giro del conjunto de discos.

El almacenamiento de la información es posible gracias a las propiedades magnéticas de las partículas depositadas en la superficie del disco. Cada una de esas partículas alargadas forma un dipolo magnético, un pequeño imán que tiene dos polos: el positivo en un extremo y el negativo en el extremo opuesto.

En una zona todos los dipolos pueden estar orientados con el polo positivo mirando hacia la derecha. En otra zona siguiente, siempre a continuación en sentido circular, los dipolos podrían tener todos el polo positivo mirando hacia la izquierda. Se ve pues que pueden darse dos situaciones diferentes para la

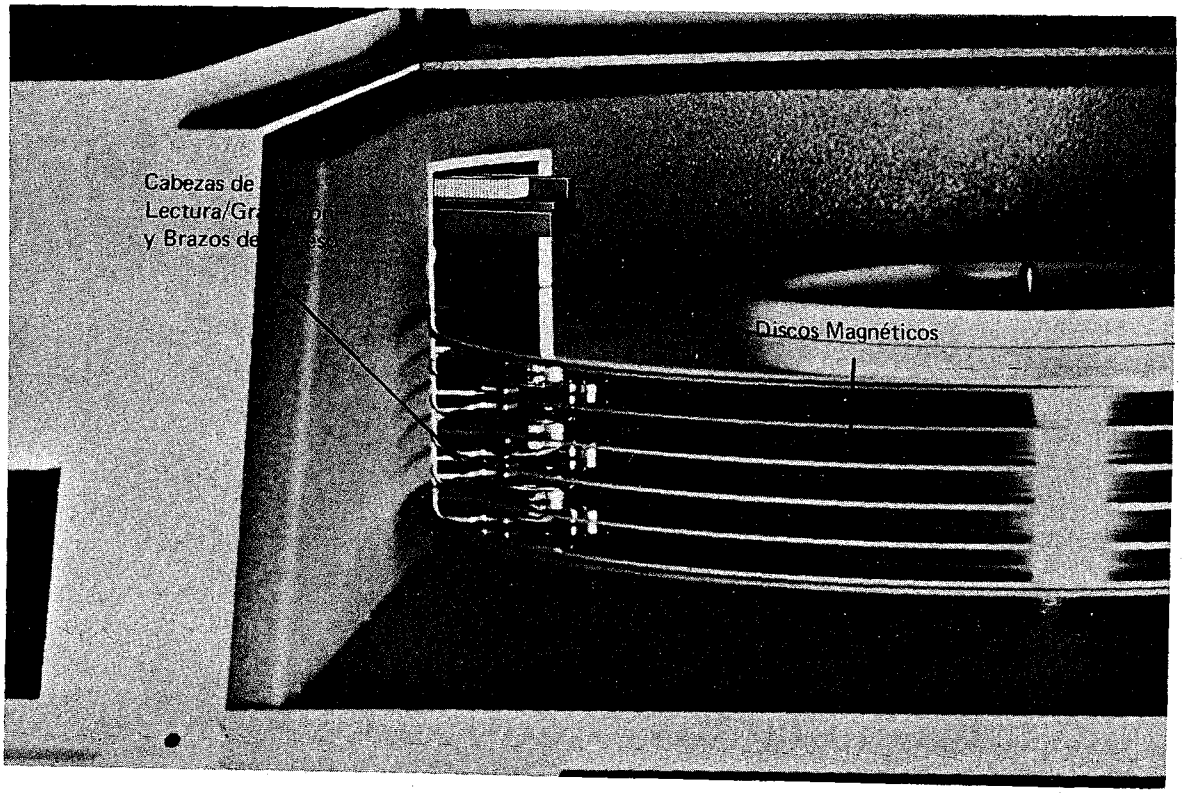


Fig. 4.1. Fotografía de un dispack en la que se pueden apreciar los cabezales de lectura-grabación.

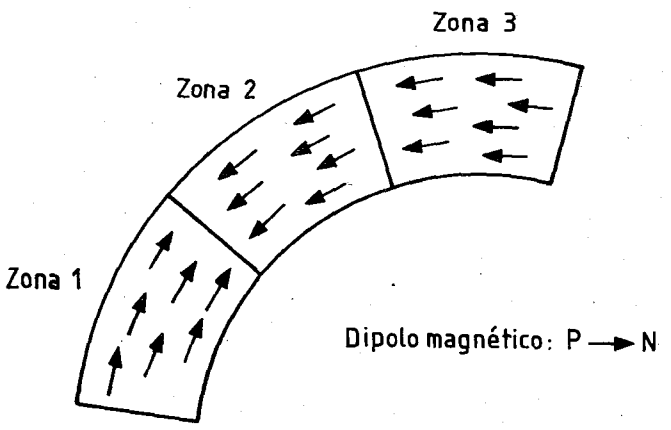


Fig. 4.2. De la zona 1 a la zona 2 la orientación de los dipolos ha cambiado, esto se interpreta como que existe el bit 1 almacenado (por ejemplo). De la zona 2 a la zona 3 la orientación se mantiene constante, la información almacenada es el bit 0 (al contrario del anterior). El conjunto de información grabado es 10.

orientación de los dipolos, lo que significa que puede almacenarse un bit; una orientación podría ser el bit 0 y la otra el bit 1. Realmente en la mayoría de los casos el código empleado no es el expresado, sino lo que decide que existe un bit almacenado, por ejemplo el 1, es cuando de una zona a la siguiente la orientación ha cambiado, mientras que se considera que existe almacenado el otro bit (el cero en el ejemplo) cuando la orientación permanece en la misma dirección de una zona a la siguiente.

4.2. OPERACIONES

Las operaciones relacionadas con el almacenamiento en un disco son de dos tipos:

— Escritura.
— Lectura.

4.2.1. Escritura

La escritura en el disco se efectúa gracias al fenómeno físico de la interacción entre la corriente eléctrica y los campos magnéticos. Una corriente eléctrica crea un campo magnético y éste es capaz de orientar en un sentido u otro a las partículas del disco.

Según sea el sentido de la corriente que recorre el sistema de la cabeza de escritura, así será el sentido de campo magnético, y por lo tanto el sentido de la orientación de las partículas de hierro. Para un sentido de la corriente las partículas se orientarán con el polo positivo hacia la derecha, mientras que si el sentido de la corriente se hace contrario las partículas se orientarán con el polo positivo hacia la izquierda.

Los sucesivos bits se graban en las distintas áreas del disco porque éste siempre está girando, mientras que la cabeza de escritura se mantiene fija.

4.2.2. Lectura

La lectura se basa en el mismo fenómeno de la relación entre los fenómenos magnéticos y los eléctricos.

Mientras que la cabeza de lectura va pasando por encima de la superficie del disco, el campo magnético creado por la orientación de todas las partículas influye en la cabeza. En el momento que existe un cambio en la orientación de

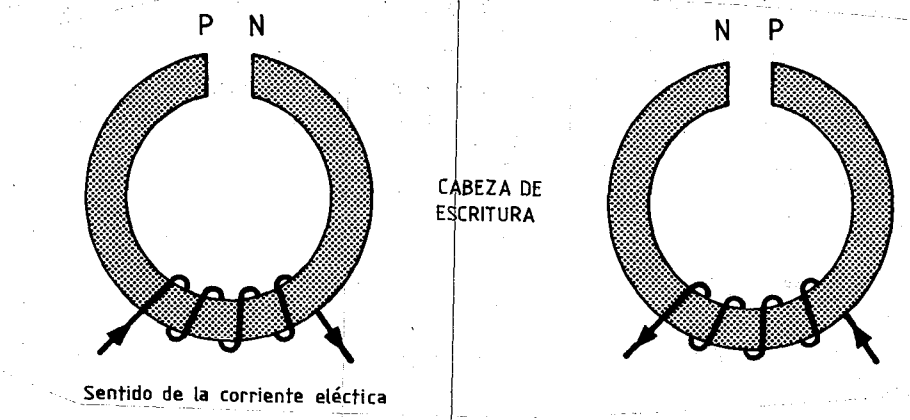


Fig. 4.3. La escritura en un disco se produce a través de la interacción entre la corriente eléctrica y los campos magnéticos.

las partículas hay un cambio brusco en el campo magnético, que pasa de tener un sentido a tener el contrario. Esta variación brusca en el sentido del campo magnético genera un voltaje eléctrico en la cabeza de lectura, interpretándose como un uno (si seguimos la codificación hecha anteriormente). En caso de que no haya diferencia en la orientación de un área a la siguiente, no hay voltaje generado y se interpreta como que la información grabada es un cero.

El disco es un soporte que permite la escritura y la lectura de la información, así pues es una memoria reutilizable, según la clasificación que se dio anteriormente.

El conjunto de los once discos se recubre con una funda guardapolvo cuando no está en uso, para evitar que las partículas de polvo se depositen en su superficie, impidiendo la lectura o la escritura. Cuando el conjunto de discos está en su soporte se tiene habitualmente un sistema de renovación y filtrado del aire que elimina las partículas de polvo que pudieran encontrarse en suspensión y depositarse en las cabezas o en los discos.

4.3. DESCRIPCIÓN LÓGICA

En los discos, por ser un dispositivo de almacenamiento de la información, deben poderse representar todas las estructuras de datos mencionadas en lecciones anteriores. Así deben poderse guardar caracteres, campos, registros, ficheros.

Puesto que se ha visto que el disco es capaz de almacenar bits, también podrá estructurar estos bits como campos, registros, etc. El modo de organizar todas las estructuras de información en el disco es lo que se engloba bajo el nombre de *estructura lógica.*

Se verán seguidamente estos métodos.

4.3.1. Sectores

Cada una de las caras de un disco se divide lógicamente (sin ninguna separación física) en 20 sectores, que corresponden a la figura geométrica de sector circular.

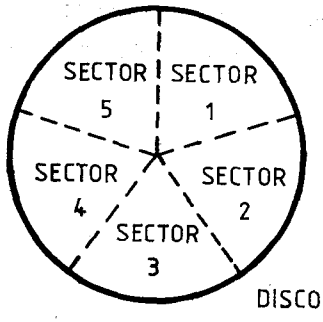


Fig. 4.4. Ejemplo de un disco dividido en 5 sectores.

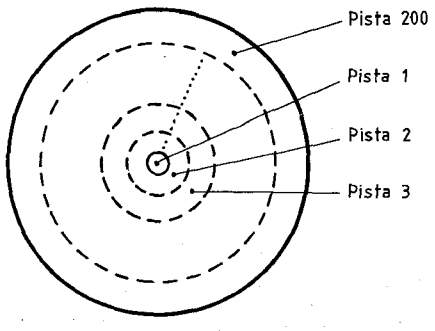


Fig. 4.5. Las pistas en un disco son concéntricas.

4.3.2. Pistas

Aparte de esta división en sentido radial se hace otra división en sentido circular que se llaman las *pistas* o posiciones de grabación. Una pista es una corona circular en el disco. Existen en total 200 pistas por cada sector del disco.

Aunque el espacio físico de un sector es menor hacia el centro del disco que en la periferia, la cantidad de información que puede almacenarse es la misma en uno que en el otro.

Por la descripción anterior las posiciones de grabación por cada cara son en total 4000 (20 × 200). Como existen 20 caras en las que se puede grabar, en un conjunto de discos las posiciones de grabación son 80.000 (4000 × 20).

A cada una de estas posiciones puede llegar una pareja de cabezas directamente controlada por un mecanismo de posición, que a su vez está gobernado por el programa del ordenador. Para el control de la posición el programa trabaja con una *dirección* (un número que indica esa posición). Así pues, cada una de las posiciones de grabación tiene asignado un número.

Cada una de estas posiciones puede contener varias palabras almacenadas. El número de las mismas depende del sistema de discos usado. En cada una de esas posiciones se va a grabar un registro lógico. Se ha visto ya que con la utilización de la técnica de bloqueo puede grabarse más de un registro en cada una de esas posiciones. Por ahora, para mayor claridad de exposición se supondrá que no existe bloqueo.

4.3.3. Formatos de grabación

Para la grabación en las zonas pueden usarse dos formatos principalmente:

— Con registro de dirección.
— Con registro clave.

Con registro de dirección

En este formato cada una de las pistas tiene una zona de grabación llamada *dirección de la pista* donde se ha grabado el número que identifica a la pista entre todas las otras. Mientras la cabeza está buscando una pista lee esta zona y averigua si ha llegado a la pista que buscaba o si tiene que seguir moviéndose.

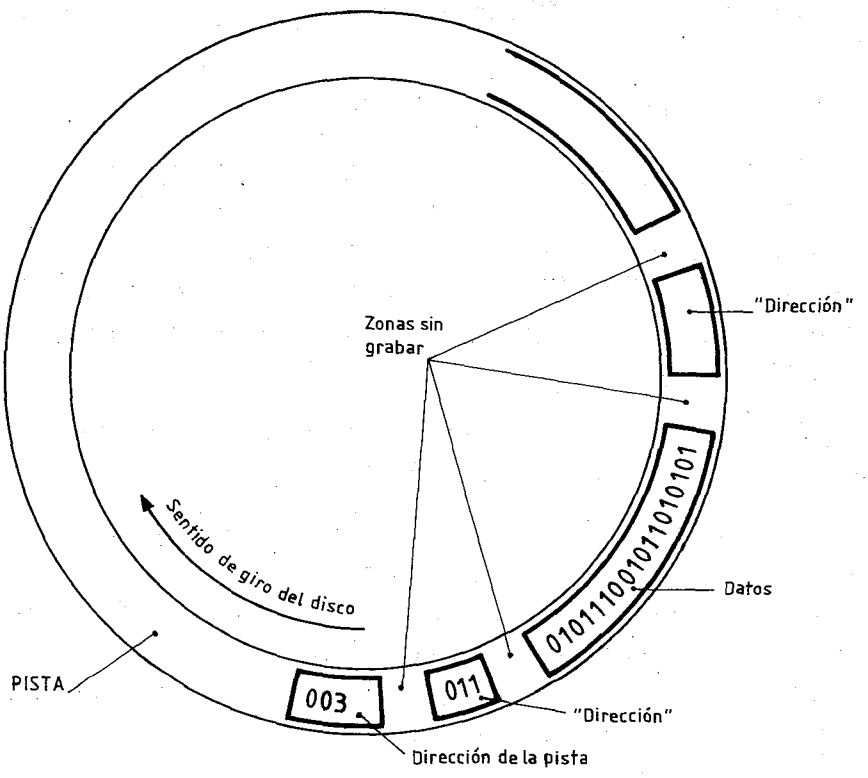


Fig. 4.6. Ejemplo del formato de grabación con registro de dirección. Se puede observar el número de dirección.

Una vez ha llegado a la pista en cuestión sigue buscando la posición en la que tiene que escribir o leer. Esta posición está identificada por un número llamado «dirección» que precede a toda la información grabada en la zona. Este número es realmente la dirección de la posición.

La disposición completa de una posición de grabación sería sucesivamente: la zona «dirección», donde está grabada la dirección de la posición que viene a continuación; seguidamente una zona sin grabar que separa la zona «dirección» de la zona de datos; después viene el espacio para la grabación de todos los datos; por último se encuentra una zona sin grabación, que separa lo anterior de la siguiente zona «dirección».

Además de esta disposición de zonas, en algún lugar de la pista se coloca la *dirección de la pista,* rodeada por ambos lados por zonas sin grabar.

Con registro clave

Este formato es idéntico al anterior salvo que en cada una de las posiciones de grabación (identificadas por la zona «dirección») se añade una zona de grabación, llamada *clave,* que identifica el contenido del registro que está grabado en esa zona «dirección». Esta zona está separada de las demás por convenientes zonas sin grabar. Además la zona *clave* precede a la zona «dirección».

Esta zona puede ser explorada por la cabeza de lectura para identificar un registro.

Como cuestión aparte, la grabación de la información en los discos se hace, como en todos los soportes, en base a un código para representar los caracteres. En este código, que hace uso de un número determinado de bits para la representación de un carácter, (usualmente ocho), tiene en ese conjunto incluidos los bits de paridad necesarios para la corrección de los errores que pudieran existir en la información.

4.3.4. Fases de la lectura/escritura

Desde que un programa necesita leer o escribir una información en el disco hasta que esto sucede, ha transcurrido un cierto tiempo en el que el sistema de control del disco se ha preparado y ha ejecutado la acción. Las fases por las que pasa este proceso son las siguientes:

1. Movimiento de acceso.
2. Puesta en marcha de la cabeza.
3. Espera en rotación.
4. Transferencia de datos.

1.ª fase: movimiento de acceso.

Puesto que un sistema ordenador poseerá más de un grupo de discos habitualmente (cada uno en una unidad), en esta fase se decide qué grupo de discos se necesita para grabar o leer la información. Al finalizar la fase el grupo de 20 cabezas se ha posicionado en el grupo de discos correspondiente y en la pista correcta.

2.ª fase: puesta en marcha de la cabeza.

Ahora se elige uno de los 20 discos, precisamente aquel en el que la información se quiere escribir o leer. Para ello se *activa* una de las 20 cabezas, precisamente la correspondiente a la cara del disco elegido.

3.ª fase: espera en rotación.

En esta fase se pone en marcha el motor que hace girar los discos. Sólo uno de los discos nos interesa y aunque giren todos, sólo hay uno elegido, el correspondiente a la única cabeza excitada. Durante esta rotación la cabeza va revisando las direcciones de las zonas hasta dar con la que necesita, en este momento empieza la 4.ª fase.

4.ª fase: transferencia de datos.

Durante esta fase los datos son transferidos del disco al ordenador o viceversa, según se trate de una lectura o de una escritura respectivamente.

Se define *tiempo de acceso* como el tiempo que transcurre desde que el ordenador sabe que tiene que utilizar un disco hasta que las cabezas se posicionan sobre el área del disco que se va a utilizar.

Se denomina *tiempo de transferencia* al tiempo que transcurre durante el trasvase de la información del ordenador al dispositivo o viceversa, según se trate de lectura o escritura.

Comparando con las fases de la lectura y escritura se puede decir que el tiempo de acceso es el que se tarda en cubrir las tres primeras fases y el de transferencia es el que se tarda en cubrir la fase 4.ª.

4.4. TIPOS DE DISCOS

Los diferentes sistemas de ordenador con diferentes potencias y volumen de información manejados utilizan distintos tipos de discos con más o menos posibilidades de almacenamiento. Una clasificación general es la siguiente:

— Discos flexibles.
— Discos rígidos.

La diferente denominación que se da a los dos grupos tiene que ver con las propiedades físicas del material con que están hechos los discos.

4.4.1. Discos flexibles

Estos discos tienen el soporte hecho con un tipo de plástico llamado mylar. La parte magnética, la que realmente graba la información, está formada por un recubrimiento de óxido de hierro.

Las características de este disco no permiten una densidad de almacenamiento muy elevada (alrededor de 20 pistas por centímetro de radio). Las cabezas van rozando la superficie mientras la exploran y por lo tanto las velocidades de rotación deben ser pequeñas. A causa de estas velocidades los tiempos de acceso son grandes.

Su aplicación principal se da en sistemas de ordenador no muy potentes.

4.4.2. Discos rígidos

Son llamados también de alto rendimiento a diferencia de los flexibles, las densidades de integración alcanzadas son del orden de 400 pistas por centímetro de radio. En estos sistemas la cabeza va siempre flotando en una capa de aire que es generada por la rotación del disco y por la forma de la cabeza. El tiempo de acceso es disminuido grandemente en relación a los discos flexibles.

El material con el que está constituido el disco es aluminio para el soporte y óxido de hierro revistiéndolo.

Como última novedad en el almacenamiento de información en discos están los discos explorados por rayo láser, en los que pueden lograrse densidades de almacenamiento aún mayores por ser el láser un rayo de luz que se puede concentrar en regiones más pequeñas. También al disminuirse las partes mecánicas el tiempo de acceso disminuye.

En estos discos la información está grabada mediante agujeros en la superficie del disco. La presencia de un agujero puede significar el bit 1 y la ausencia el bit cero, por ejemplo.

En el momento de la lectura se lanza un rayo de luz que se refleja diferentemente si existe o no agujero. Se recoge este rayo reflejado y se puede decidir si está grabado un cero o un uno.

Para la escritura se hace uso de las propiedades del láser. El láser quema una pequeña parte de la superficie del disco quedando un agujero de diámetro muy pequeño. Naturalmente, las memorias desarrolladas con este sistema son de tipo no reutilizable, ya que no se pueden borrar y volver a grabar.

4.5. PROCESO TARJETA-DISCO

En todo el desarrollo del estudio de los discos siempre se ha visto al disco acompañado del ordenador. Nunca se ha hablado de una «perforadora» de disco, o de una máquina que puede manejar el operador a partir de un documento base para crear ficheros en el disco.

El proceso de creación de ficheros en disco (en el fondo, de grabar la información), siempre se tiene que llevar a cabo por el ordenador.

Este proceso necesita dos fases.

— Creación del fichero en tarjetas.
— Creación del fichero en discos.

La primera fase del proceso que se sigue es totalmente idéntico al comentado en la lección de tarjetas perforadas.

La creación del fichero en disco empieza cuando estas tarjetas son leídas por el ordenador. Este designa un lugar para la información de la tarjeta en una posición del disco. Una vez encontrado físicamente este lugar, la máquina graba los datos en esa posición.

El proceso tendría un algoritmo que puede ser el siguiente:

— Lee una tarjeta.
— Busca una dirección que libre en el disco.
— Graba la información en el disco.
— Repetir 1, 2, 3, mientras no se hayan acabado las tarjetas.

El punto dos del algoritmo es el más complicado.

Ejemplo de proceso con utilización de discos.

Se tiene un disco en el que se ha almacenado, para cada estudiante de una escuela los siguientes datos.

— nombre
— dirección

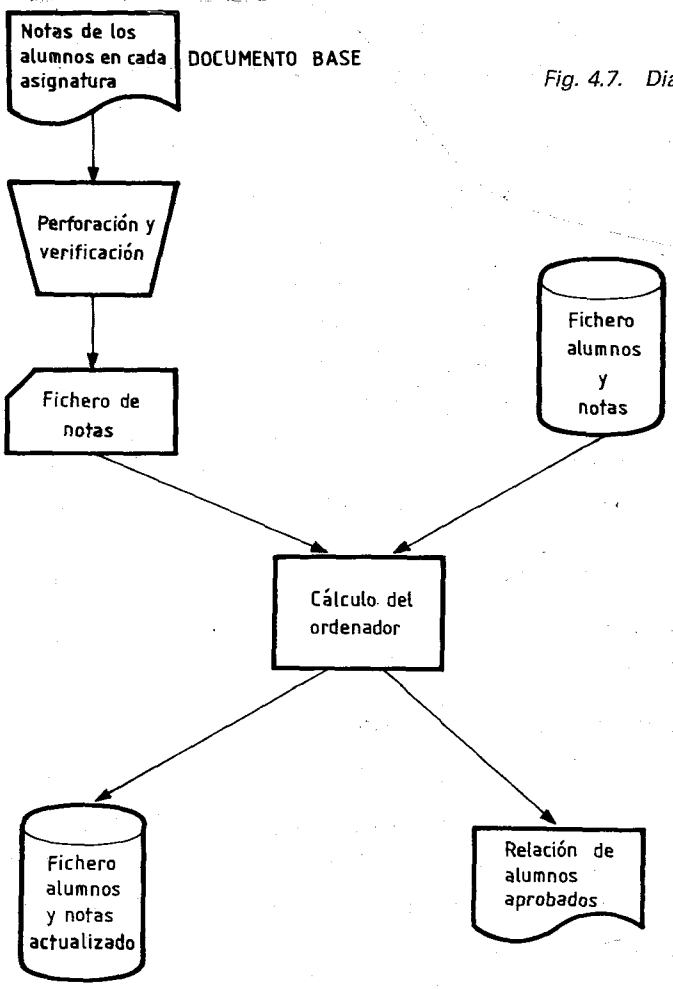


Fig. 4.7. Diagrama de flujo del ejemplo.

— notas en cada una de las asignaturas
— créditos acumulados
— código del alumno.

Se entenderá por créditos una puntuación que se le da al alumno por haber aprobado una asignatura. Cuando el alumno ha alcanzado un número determinado de créditos ya se considera aprobado en esa especialidad y puede dársele el título.

Se quiere actualizar el fichero con los resultados de los exámenes que se

han acabado de realizar y dar un listado de todos los que ya tienen un número de créditos suficientes para salir de la escuela con el título.

La primera operación que se debe hacer es crear un fichero con todas las nuevas notas de los alumnos, naturalmente en tarjeta perforada.

En este fichero estará el código del alumno y todas las nuevas notas que ha obtenido. Una vez perforado y verificado, el ordenador empieza a leerlo, tarjeta por tarjeta.

Con el código del alumno encuentra la zona donde están los datos anteriores del alumno y graba a continuación los nuevos datos (este proceso tiene muchos más pasos de los que están expuestos aquí, pero estos se consideran suficientes para entender el método general).

El contador donde se han grabado los créditos del alumno se incrementa y se compara con el mínimo que necesita el alumno. En caso de superarse este mínimo se imprimirán el nombre del alumno en la salida.

Todo este proceso realizado con una tarjeta se repite hasta que se hayan acabado las tarjetas con las correcciones.

El organigrama que ilustraría este proceso se muestra en la figura 4.7.

4.6. SISTEMA COMPUTADOR CON DISCOS

Los elementos mínimos de que debería disponer un sistema de ordenador que trabaja con discos son los siguientes:

— Unidad central de proceso.
— Periféricos de perforación y verificación de fichas, lectora de fichas.
— Discos y unidades de discos.
— Impresora o periférico de salida.

Con estos elementos pueden realizarse las operaciones básicas de computación y manejo de discos.

4.7. SISTEMAS DE SEGURIDAD

En el caso de los discos el sistema para recuperar la información en caso de destrucción vuelve a ser la copia de todo el contenido del disco u otro dispositivo de almacenamiento.

Además de la seguridad de todo el disco, la grabación en él se efectúa mediante un código apropiado para que el sistema se dé cuenta de si existe o no un error en los bits que está grabando o leyendo.

Uno de los códigos dispone para esta corrección de dos caracteres de verificación cíclica (del inglés *cyclic check characters*). Estos dos caracteres se colocan al final de las zonas importantes, esto es, de la zona «dirección», de la zona *clave* y de la zona de *datos*.

En una instrucción de escritura el programa se encarga de calcular el valor de estos caracteres según la combinación de los bits de datos que pertenecen a cada una de las zonas. En las instrucciones de lectura se vuelven a calcular los caracteres de verificación cíclica y se comparan con los que se han leído. En caso de igualdad entre ambos tipos el proceso se ha desarrollado sin ningún error.

EJERCICIOS

1. ¿Cómo se llama el conjunto de discos de un mismo eje?

   a) Display.
   b) Dispack.
   c) Disco.

2. Se entiende por tiempo de transferencia El tiempo que transcurre desde que la información va al ordenador desde el dispositivo de memoria

3. Cuando se graba en un disco utilizando un registro de dirección, ¿qué significa?

   a) Que cada pista está dividida en sectores.
   b) Que a cada una de las posiciones se le añade una clave.
   c) Que cada una de las pistas tiene una zona de grabación denominada dirección de pista.

4. En las fases de lectura-escritura ¿cuál es la tercera fase?

   a) Transferencia.
   b) Espera en rotación.
   c) Puesta en marcha de la cabeza.

5. ¿Cuál es la diferencia entre los discos flexibles y los rígidos?

   a) Los flexibles se utilizan poco y los rígidos se utilizan más.
   b) El sentido de giro de uno es inverso al otro.
   c) Los flexibles se utilizan en sistemas poco potentes y los rígidos son de alto rendimiento.

5.

MANEJO DE DATOS

Tablas

CUADRO RESUMEN DE LA LECCIÓN

Manejo de datos (I): Tablas

- Variables simples
- Vectores
  - Nombre y valor de una variable
  - Formato de una variable
- Matrices
- Elementos de una tabla
- Asignación de la memoria
  - Secuencial
  - Encadenada
  - Comparación entre secuencial y encadenada
- Manejo de tablas

Para usar correctamente el ordenador es muy importante el estudio de las estructuras de los datos, sus relaciones y las técnicas necesarias para el manejo de dichas estructuras.

Los programas que procesa el ordenador operan frecuentemente con *tablas de datos.* Una tabla está formada por distintos valores numéricos o datos almacenados (en la memoria central o en una auxiliar) y relacionados entre sí. Cada uno de estos valores numéricos se referencian mediante una *variable.* Es decir, una variable indica la posición de memoria en donde se encuentra almacenado un cierto valor numérico o cierta información.

Para una variable cualquier, X, sólo puede reservarse una sola posición de memoria. Por lo tanto, para esta variable puede almacenarse un solo valor numérico en esa posición de memoria. Si se desean reservar varias posiciones de memoria para la variable X, basta con asociar a dicha variable un subíndice, es decir, X (I). De esta forma, el mismo nombre de la variable, X, puede referenciar a diferentes posiciones de memoria según el valor del subíndice I. Este subíndice indica, por lo tanto, una determinada posición de memoria dentro del conjunto de posiciones referenciadas por la variable X.

Las variables para las que sólo se reserva una posición de memoria se denominan *variables simples,* y las variables para las que se reservan más de una posición de memoria se llaman *tablas* o *variables con subíndice.*

Así pues, estos conjuntos de datos almacenados en la memoria pueden referenciarse mediante dos tipos de variables:

— Variables simples.
— Variables con subíndice o tablas.

A su vez las tablas pueden ser de tres tipos:

— Variables con un subíndice *(vectores).*
— Variables con dos subíndices *(matrices).*
— Variables con más de dos subíndices *(tablas multidimensionales).*

5.1. VARIABLES SIMPLES

Una *variable* es un elemento de los lenguajes de programación que indica una dirección de memoria. En lugar de decir: la dirección de memoria 111, por ejemplo se podrá decir la variable X, y se estará hablando de lo mismo. La utilización de una variable es exclusiva de los lenguajes de alto nivel. En los lenguajes máquina la referencia a una posición de memoria se hace por su dirección y no por un símbolo que la exprese.

En las variables pueden considerarse tres aspectos:

— El valor de la variable.
— El nombre de la variable.
— El formato de la variable.

5.1.1. Nombre y valor de una variable

El *valor de la variable* es el dato que contiene la posición de memoria a la que se hace referencia.

El *nombre de la variable* es un símbolo que se usa para llamar a esa variable. Siempre que se quiera hacer referencia a una variable se empleará su nombre. El símbolo que se le dé a dicho nombre depende del lenguaje que se use, pero en general es un conjunto de caracteres. El tipo de caracteres usados designa el tipo de variable que se va a utilizar, esto es, el tipo de datos que puede contener la variable.

Ejemplo

Sea la posición de memoria 000, que contiene como dato la letra H (un valor alfanumérico). A esa posición de memoria se le da el nombre de variable A$. Se puede decir que el valor de la variable A$ es «H», que se suele escribir como A$ = «H». (Fig. 5.1.)

Si en otro momento del programa se cambia el contenido de la posición de memoria, y ésta pasa a tener el dato «I», la variable tendrá el valor A$ = «I».

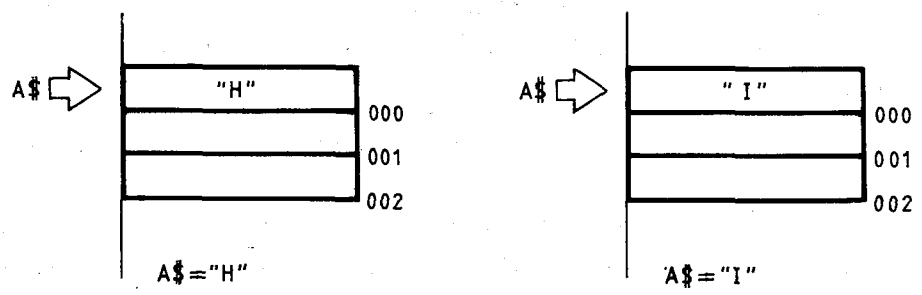


Fig. 5.1. Posiciones de memoria.

Así, una variable nombra siempre a una posición de memoria. Puesto que el dato almacenado en esa posición de memoria va variando a lo largo del programa, el valor de la variable también lo hará.

Al introducir datos al ordenador desde un dispositivo de entrada también se debe usar una variable. En este caso, la variable se refiere a la posición de memoria donde va a almacenarse ese dato de entrada. El ordenador recoge el dato y lo lleva a la posición de memoria.

5.1.2. Formato de una variable

Habitualmente, para la lectura de un registro se emplea la técnica del formato.

El formato explica al ordenador cómo se separan los diversos campos en un registro y qué contiene cada uno. Este es el caso de formato de entrada.

También existe un formato de salida que explicaría al ordenador como se distribuye el espacio de la impresión (sea un papel, una pantalla o cualquier dispositivo) para cada una de las variables a las que va a darse salida.

Ejemplo de formato de entrada:

Sea una información grabada en una tarjeta perforada. Se ha perforado un dato numérico de 5 dígitos, después otro de 10, a continuación un dato alfanumérico de tres caracteres y por último un dato alfanumérico con 30 caracteres.

El formato equivale a una plantilla que se pone encima de la ficha para ver cuántas partes tiene.

En el ejemplo la tarjeta consta de cuatro campos, dos de ellos numéricos y los dos últimos alfanuméricos. A cada campo se le hace corresponder una variable que es respectivamente:

L ACS B$ C1$

El formato para este caso es:

```
FORMAT(F5.0, F10.0, A3, A30)
```

Se ha usado para codificar la instrucción que designa el formato, la sintaxis del lenguaje FORTRAN.

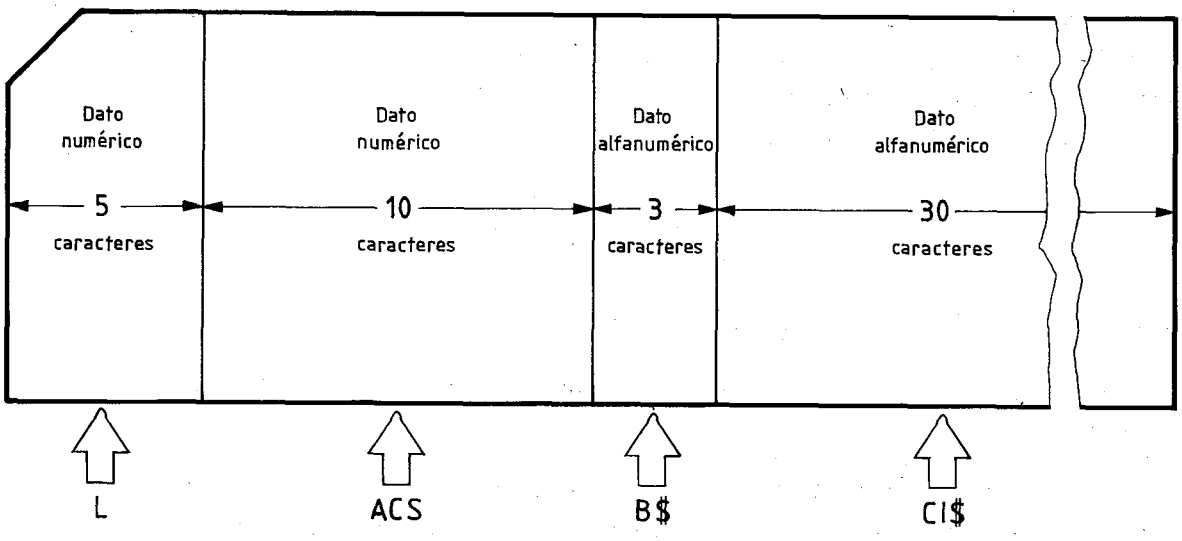


Fig. 5.2. Formato de entrada.

La explicación del formato es:

— F5.0 significa que hay un dato numérico que ocupa 5 columnas y no tiene punto decimal (F indica formato en punto fijo).

— F10.0 de la misma forma es un dato numérico que ocupa 10 columnas sin punto decimal.

— A3 designa un dato alfanumérico que ocupa 3 columnas.

— A30 designa un dato alfanumérico que ocupa 10 columnas.

Todas las tarjetas que se lean con este formato se interpretarán de la misma forma. Si por confusión, el operador coloca en el primer campo un dato en las columnas, el ordenador interpretará la 5.ª columna como también perteneciente al dato, lo que será erróneo.

Estos datos serán almacenados en posiciones de memoria diferentes. Cada una de esas posiciones vendrá nombrada por una variable.

Si en la tarjeta del ejemplo se ha perforado la lista de caracteres:

189761763434576ABCGHIJTUYBDINHRORNJSYOPENJHDYTIS

La separación en campos gracias al formato da:

18976	1763434576	ABC	GHIJTUYBDINHRORNJSYOPENJHDYTIS
L	ACS	B$	C1$

Los valores de las variables serán por lo tanto

L = 18976

ACS = 1763434576

B$ = «ABC»

C1$ = «GHIJTUYBDINHRORNJSYOPENJHDYTIS»

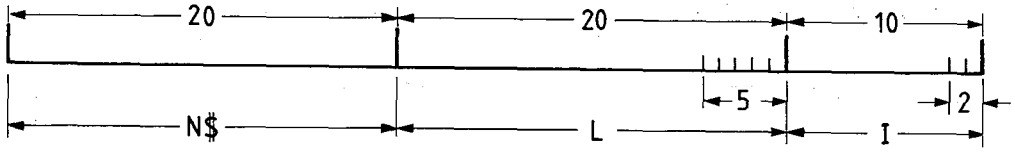
Las comillas en las variables B$ y C1$ quieren significar que es un dato alfanumérico.

Ejemplo de formato de salida:

Se va a escribir en una impresora de papel el contenido de tres posiciones de memoria. Las posiciones vienen definidas por las variables N$, L I. Estas variables son, respectivamente, una alfanumérica y dos numéricas. La primera numérica está representada en coma flotante y la segunda en coma fija.

A priori no se conoce la cantidad de dígitos almacenados en cada una de las variables de modo que no se puede dar un formato exacto para ellas. El formato indica, en este caso, las posiciones de escritura que se reservan a cada variable. El programador dará como formato el caso que se espera que ocupe más posiciones. Se dan 20 posiciones para la primera alfanumérica, otras 20 posiciones para la de coma flotante que viene a continuación (reservando 5 de ellas para la parte decimal) y por último, se darán 10 posiciones para la coma fija (dos de ellas para la parte decimal).

En zona de escritura se dividirá de la forma siguiente:



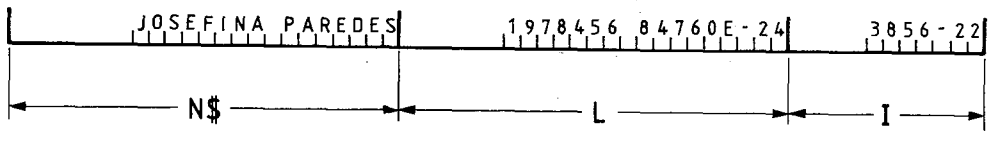
Si los contenidos reales de la variable son:

N$ = «JOSEFINA PAREDES»

L = 1987456.8476E−24

I = 3856.22

La zona de impresión quedará rellena de la siguiente manera.



5.2. VECTORES

Los *vectores,* también llamados *variables con un subíndice,* son variables que manejan un conjunto de posiciones de memoria. En cada una de estas posiciones de memoria se colocan datos que están relacionados entre sí.

A todo ese conjunto de posiciones de memoria se le da el nombre general de *vector.* A cada una de las posiciones de memoria de que consta el vector, se le da el nombre del vector y un número entre paréntesis que designa el orden que ocupa esa posición dentro del conjunto total. Depende del sistema en particular si al primer elemento del conjunto se le da el número cero o el uno.

Como ejemplo sea un vector con tres *componentes* (a cada una de las posiciones de memoria que forman el vector se le llama componente). El vector se llamará N(I). El nombre del vector es N y se pone entre paréntesis una variable para designar que es un vector ((I)), si no se pusiera el paréntesis y la variable interior, se podría confundir con las variables comunes, sin subíndice.

El valor de esta variable interna al paréntesis (o subíndice, como también se le llama) variará entre 1 y n, siendo n el número total de componentes del vector, habiendo empezado a nombrar el primer elemento con el subíndice 1.

El vector del ejemplo tiene como componentes: N (1), N (2), N (3), siempre considerando que el primer componente se designa por el subíndice uno y no por el cero. [Designando el subíndice primero por cero, los nombres de los componentes serían: N(0), N(1), N(2)].

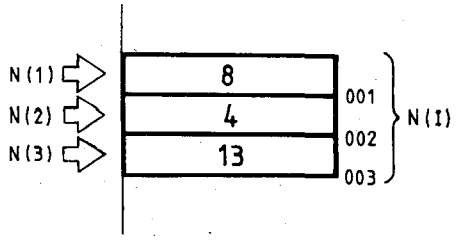


Fig. 5.3. Elementos de un vector.

Los valores de estas variables podrían ser, por ejemplo:

N(1) = 8
N(2) = 4
N(3) = 13

lo cual significa que el contenido de cada una de las posiciones de memoria es 8, 4 y 13 respectivamente (Fig. 5.3).

Las aplicaciones más importantes de los vectores son aquellas en que el subíndice tiene algo que ver con el contenido del componente.

Por ejemplo, se quiere realizar el cálculo de los cuadrados de los primeros números naturales. El subíndice sería el número natural y el contenido del componente el valor del cuadrado.

En el caso de los cuadrados de los tres primeros números naturales, se tendría un vector:

N(1) = 1
N(2) = 4
N(3) = 9

El uso de este vector en una aplicación sería inmediato. Si se quisiera conocer el cuadrado del número 3, se buscaría el contenido del componente con subíndice 3 en el vector.

Otra aplicación sería el almacenamiento de resultados, o cálculos que se efectúan uno detrás del otro en el tiempo.

Al que se obtiene en primer lugar se le coloca en el primer componente del vector, al siguiente en el segundo componente y así sucesivamente.

5.3. MATRICES

Las *matrices* son variables con dos subíndices. Si se hace una representación de los datos en forma de rectángulo, con filas y columnas, un dato se puede identificar por la columna a que pertenece y por la fila. Ver figura 5.4.

Así, el dato 3 pertenece a la columna 1, fila 3.

En la notación de una variable con dos subíndices o una matriz, se expresa

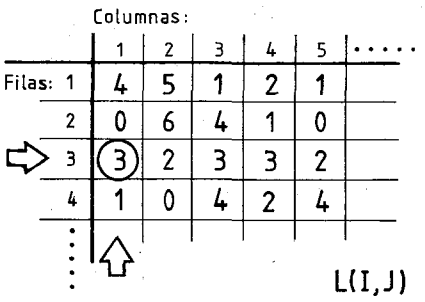


Fig. 5.4. Matriz.

el nombre de la matriz, de la misma forma que el nombre de una variable, seguido por un paréntesis en el que están dos subíndices.

Un ejemplo del nombre de una matriz es:

L(I,J)

donde los subíndices I,J varían y recorren todas las filas y todas las columnas. El primer subíndice que se encierra en el paréntesis (I en este caso), indica el número de la fila y el segundo (J), el nombre de la columna que se está considerando. La numeración que se le da a las filas y columnas puede empezar por el cero para designar el primer componente o por el 1, igual que en los vectores.

Existen sistemas de ordenador que pueden usar variables con más de dos subíndices.

El número de componentes que tiene una matriz se calcula multiplicando el número de filas por el número de columnas que posee, ya que una fila y una columna particulares definen a un solo componente. El modo en que los componentes de la matriz se asignan a cada una de las posiciones de la memoria depende del sistema en particular. En ningún caso el programador tiene que intervenir en esta asignación, que es realizada por el compilador.

Ejemplo

Calcular el número de componentes de una matriz con once filas y tres columnas.

El número de componentes es:

11 filas * 3 columnas = 33 componentes.

Por lo tanto el ordenador debe reservar 33 posiciones para el almacenamiento de esta matriz.

Al igual que los vectores, cada componente de la matriz tiene un nombre y un valor.

La matriz:

$$\begin{bmatrix} 3 & 4 & 5 \\ 2 & 7 & 6 \end{bmatrix}$$

que tiene dos filas y tres columnas, y que va a nombrarse por A(I,J), tiene como valores de sus componentes los siguientes:

A(1,1) = 3	A(2,1) = 2
A(1,2) = 4	A(2,2) = 7
A(1,3) = 5	A(2,3) = 6

que podrían almacenarse en el ordenador en sucesivas posiciones como se muestra en la figura 5.5.

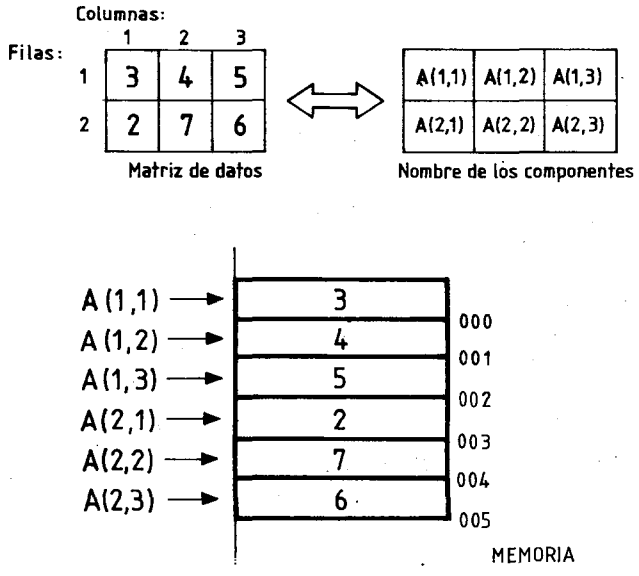


Fig. 5.5. Almacenamiento de una matriz.

En la programación se presentan situaciones en las que la forma más conveniente de representación de los datos es la matriz. Un ejemplo es el de la resolución de sistemas de ecuaciones.

Se presenta aquí cuál sería la disposición de los componentes de la matriz y sus valores para el siguiente ejemplo de sistema de ecuaciones:

$$2x + 3y = 4$$

$$5x + 2y = 20$$

La matriz correspondiente a los coeficientes de las incógnitas es:

$$\begin{bmatrix} 2 & 3 \\ 5 & 2 \end{bmatrix}$$

Los términos independientes (cantidades a la derecha del igual) se representan por el vector: (4,20)

Llamando C(I,J) a la matriz de coeficientes y T(K) al vector de términos independientes, los valores de los componentes del vector y matriz son:

T(1) = 4
T(2) = 20
C(1,1) = 2
C(1,2) = 3
C(2,1) = 5
C(2,2) = 2

5.4. ELEMENTOS DE UNA TABLA

La tabla está formada por un conjunto de *registros.* Los registros constituyen los datos que se almacenan. A menudo, a los registros se les da el nombre de *nodos,* especialmente en las tablas que tienen estructura de *árbol.*

Además de los datos, un registro contiene un campo para los *punteros.*

Los punteros son unos campos de los registros que indican la dirección de un registro en la tabla. Estos son pues los elementos que se utilizan para enlazar los diferentes registros de datos de la tabla. (Se entiende por *enlazar dos registros* cuando la dirección de uno es mencionada en el otro.)

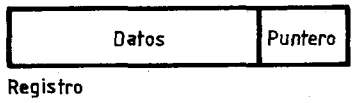


Fig. 5.6. Esquema de un registro.

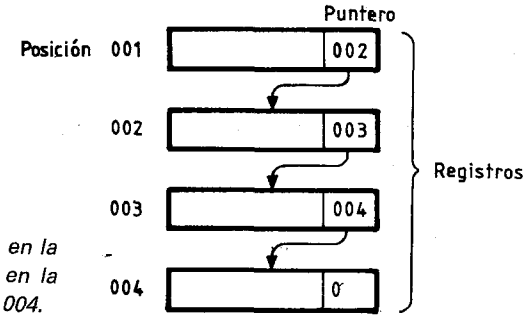


Fig. 5.7. El puntero del registro grabado en la posición 001 señala al registro grabado en la posición 002. El 002 al 003 y el 003 al 004.

5.5. ASIGNACIÓN DE POSICIONES EN LA MEMORIA

El estudio de la asignación tiene por objeto el decidir qué posiciones de almacenamiento van a usarse para guardar todos los elementos de la tabla. Existen dos modos de efectuar la asignación:

— Secuencial
— Encadenada.

5.5.1. Asignación secuencial

En este modo de direccionamiento los registros se colocan en la memoria uno detrás de otro, secuencialmente. Donde acaba un registro empieza el siguiente. Se continúa almacenando de este modo hasta que se acabe la tabla.

Para conocer la dirección donde se ha grabado un registro se debe efectuar un pequeño cálculo.

Sea L_0 la dirección donde se ha grabado la primera palabra del registro.

k el número de palabras que tiene cada registro.

c el número registro que deseamos grabar o leer.

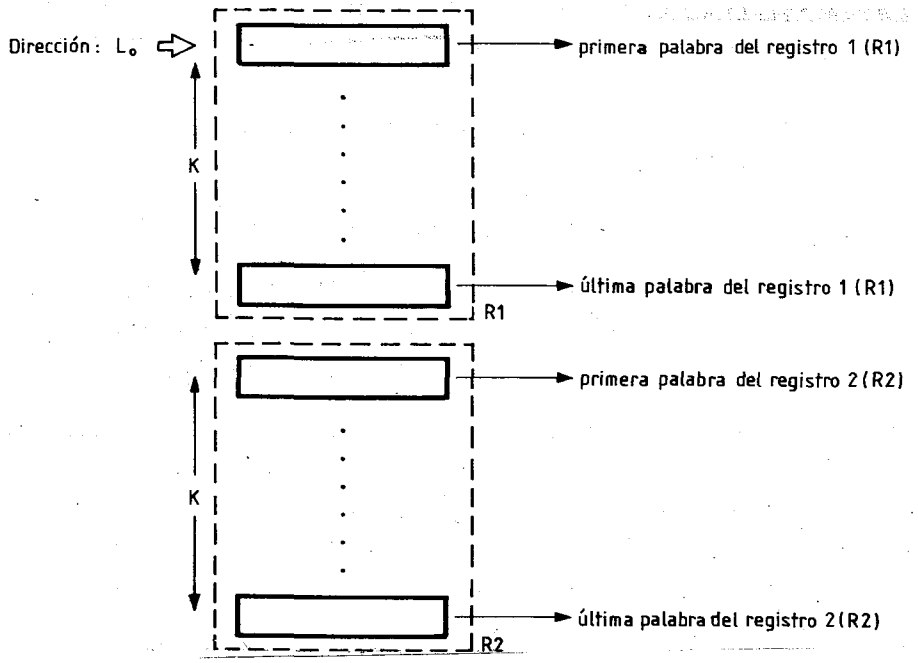


Fig. 5.8. Cálculo de la dirección de un registro.

La dirección del registro número *c* viene dada por la expresión

dirección = $L_o + (c \times k)$.

En la grabación de la tabla ocupa una cantidad de memoria total:

Cantidad de palabras de memoria que utiliza = $n \times k$, donde n es el número total de registros que existe en la tabla.

5.5.2. Asignación encadenada

Esta asignación coloca los registros en cualquier posición de la memoria. Cada registro dispone para su localización de un puntero que señala la posición de memoria en que se encuentra el siguiente registro de la tabla.

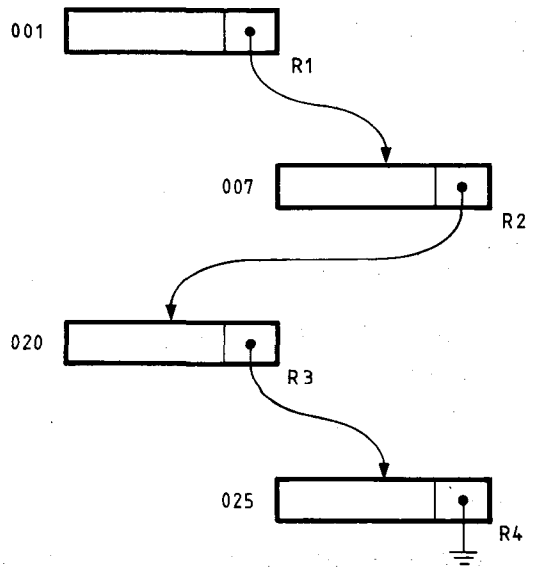
Para facilitar la localización de los registros en la memoria se confecciona una tabla en la que se indica el número de registro, la posición en que está y el siguiente registro a que se accede.

Un ejemplo de asignación encadenada puede verse en la figura 5.9.

5.5.3. Comparación entre asignaciones secuenciales y encadenadas

Las características de la asignación secuencial frente a la encadenada son:

1. La encadenada necesita más memoria, puesto que además de los registros de los datos, posee los punteros que enlazan los distintos registros. Además precisa del almacenamiento de la tabla de direcciones.
2. En una asignación encadenada varias tablas pueden ocupar la misma zona de memoria. En la secuencial, hasta que no se acaba una tabla no se puede empezar la siguiente. En la encadenada pueden colocarse registros de varias tablas en posiciones consecutivas de memoria, puesto que la existencia del puntero asegura la localización de todos los componentes de la tabla.



001	R1	007
007	R2	020
020	R3	025
025	R4	⏚

Fig. 5.9. Asignación encadenada.

3. En una tabla con asignación encadenada es muy fácil eliminar elementos. Unicamente deben corregirse los punteros adecuados que señalen a los elementos eliminados.

4. Análogamente, en la encadenada pueden insertarse nuevos elementos. En cambio, en la asignación secuencial no pueden dejarse huecos entre un registro y otro puesto que tienen que estar colocados todos los elementos uno detrás de otro. De este modo sólo se pueden insertar o eliminar registros del principio o final de la tabla, donde no se pierde la secuencia.

5. La asignación encadenada favorece la unión de dos tablas, o su separación. Otra vez el manejo y la corrección en los punteros de la respuesta. Es más fácil cambiar los punteros que reordenar una tabla secuencial.

6. Las referencias al azar a un determinado registro de la lista son mucho más rápidas en el caso secuencial. Por ejemplo el acceder al registro número k de la tabla precisa un tiempo determinado en la asignación secuencial, pero necesita un número k de iteraciones en la asignación encadenada. Esto es porque se empieza a recorrer la tabla encadenada desde la primera posición y se va variando el valor de una variable que siempre indica el registro siguiente.

5.6. MANEJO DE TABLAS

A continuación se presentan todos los casos posibles de operaciones o tratamientos que se pueden efectuar en una tabla.

Acceder al registro número K

La operación consiste en definir qué serie de pasos se han de dar hasta llegar al registro k. (k es una letra que se ha utilizado para designar a un registro cualquiera. En cada caso k tiene el valor del número de registro que se vaya a considerar.)

Insertar un registro después del K

Se desarrollan todas las operaciones —hasta que la tabla queda ampliada con un nuevo elemento colocado a continuación del elemento k. Si es una estructura con asignación secuencial se colocará físicamente detrás del k. Si la

asignación es encadenada estará en otro lugar físico de la memoria pero enlazado con el registro k.

Suprimir el elemento K

Se explica el modo de conseguir una tabla con un elemento menos, precisamente el que ocupa el lugar k: qué registros han de moverse, qué punteros modificarse, etc.

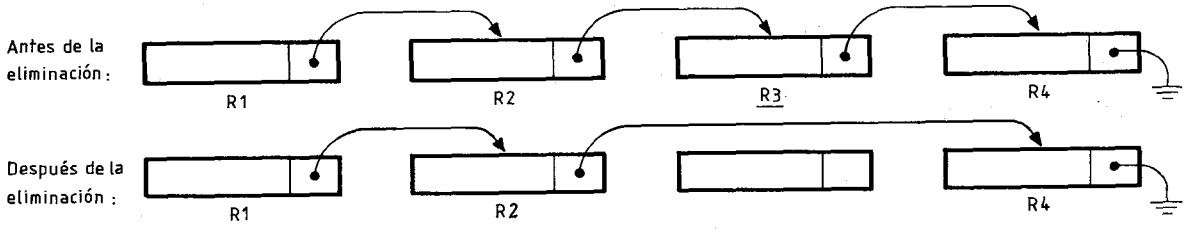


Fig. 5.10. Eliminación de un elemento en una tabla con asignación encadenada.

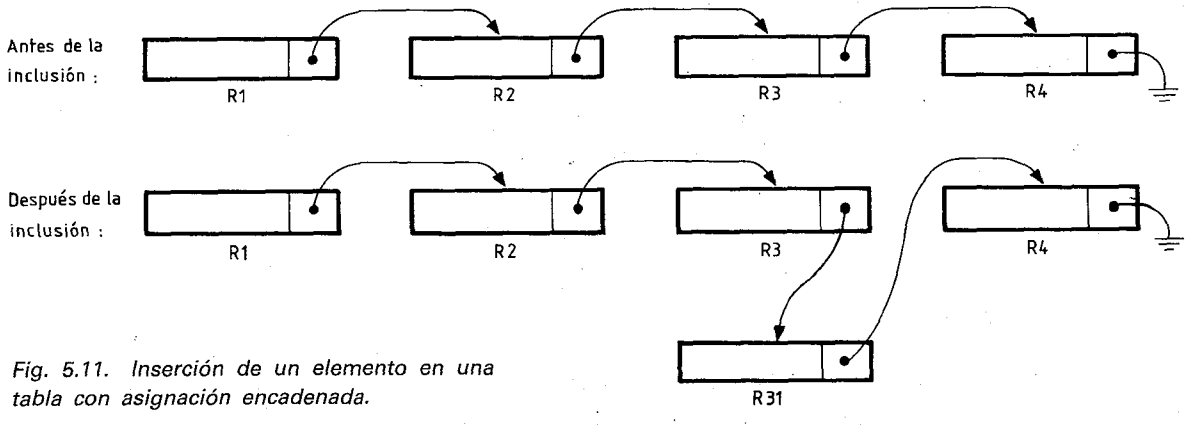


Fig. 5.11. Inserción de un elemento en una tabla con asignación encadenada.

Combinar varias tablas en una sola

Debe estudiarse la secuencia de operaciones que se ha de seguir para conseguir una tabla única a partir de varias tablas. Este es realmente un trabajo de creación de una tabla nueva, teniendo que definir el contenido de los registros, el tipo de asignación, etc.

Separar una tabla en dos

En este caso se crean dos nuevas tablas a partir de una sola. El problema no difiere mucho del mencionado en el apartado anterior, pues también es un trabajo de creación de tablas.

Copiar una tabla

Se debe explorar cada uno de los registros de una tabla y copiarlo respectivamente en otro lugar de la memoria.

Determinar el número de registros de una tabla

Este caso desarrolla una tarea de contabilidad del número de registros. El modo de realizar esa contabilidad variará según el tipo de asignación que se haya hecho.

Clasificar una tabla según un campo determinado

Esta es una operación compleja en la que se tiene que volver a construir una tabla reordenada según un campo particular de los registros. La mayoría de fabricantes proporcionan el sistema con los programas necesarios para esta operación. A menudo se conoce dicha operación con el nombre de «sorting» (clasificación).

Buscar registros que tengan un valor determinado en un campo

El problema general se conoce con el nombre de búsqueda en una tabla. Se debe explorar dicha tabla hasta dar con los registros que cumplan una condición. Existen varios algoritmos desarrollados para minimizar el tiempo de exploración, puesto que si se fuera mirando registro por registro a menudo la operación sería muy lenta.

EJERCICIOS

1. Una variable indica:

   a) Una dirección en memoria.
   b) El dato de una dirección en memoria.
   c) El símbolo de una palabra.

2. El formato explica al ordenador .....
   .....

3. Las tres variables de un formato de salida son:

   a) Numéricas.
   b) Una alfanumérica y dos numéricas.
   c) Dos alfanuméricas y una numérica.

4. Los vectores se llaman también .....

5. Cuando el subíndice tiene algo que ver con el contenido del componente, será:

   a) Una variable.
   b) Una variable subindicada.
   c) Una variable indicada.

6. Una matriz es un vector con ..... subíndices.

7. El número de componentes de una matriz con 10 filas y 5 columnas es:

   a) 15
   b) 5
   c) 50

8. Una tabla puede estar almacenada sólo en:

   a) La memoria central.
   b) La memoria auxiliar.
   c) Las dos respuestas son válidas.

9. Un conjunto de registros define .....

10. Un nodo es:

   a) Un puntero.
   b) Un registro.
   c) Una tabla.

11. Los punteros indican la dirección de:

   a) Un registro en la tabla.
   b) La tabla en la memoria central.
   c) Un dato dentro de un registro.

12. El estudio de la asignación pretende decidir .....
   .....
   .....

13. La dirección de un registro viene dada por la expresión .....

14. La cantidad de palabras de memoria necesarias para grabar una tabla es:

   a) Cantidad de registros × cantidad de palabras del registro.
   b) Cantidad de registros + cantidad de palabras del registro.
   c) Cantidad de registros / cantidad de palabras del registro.

15. Para unir dos tablas será más favorable:

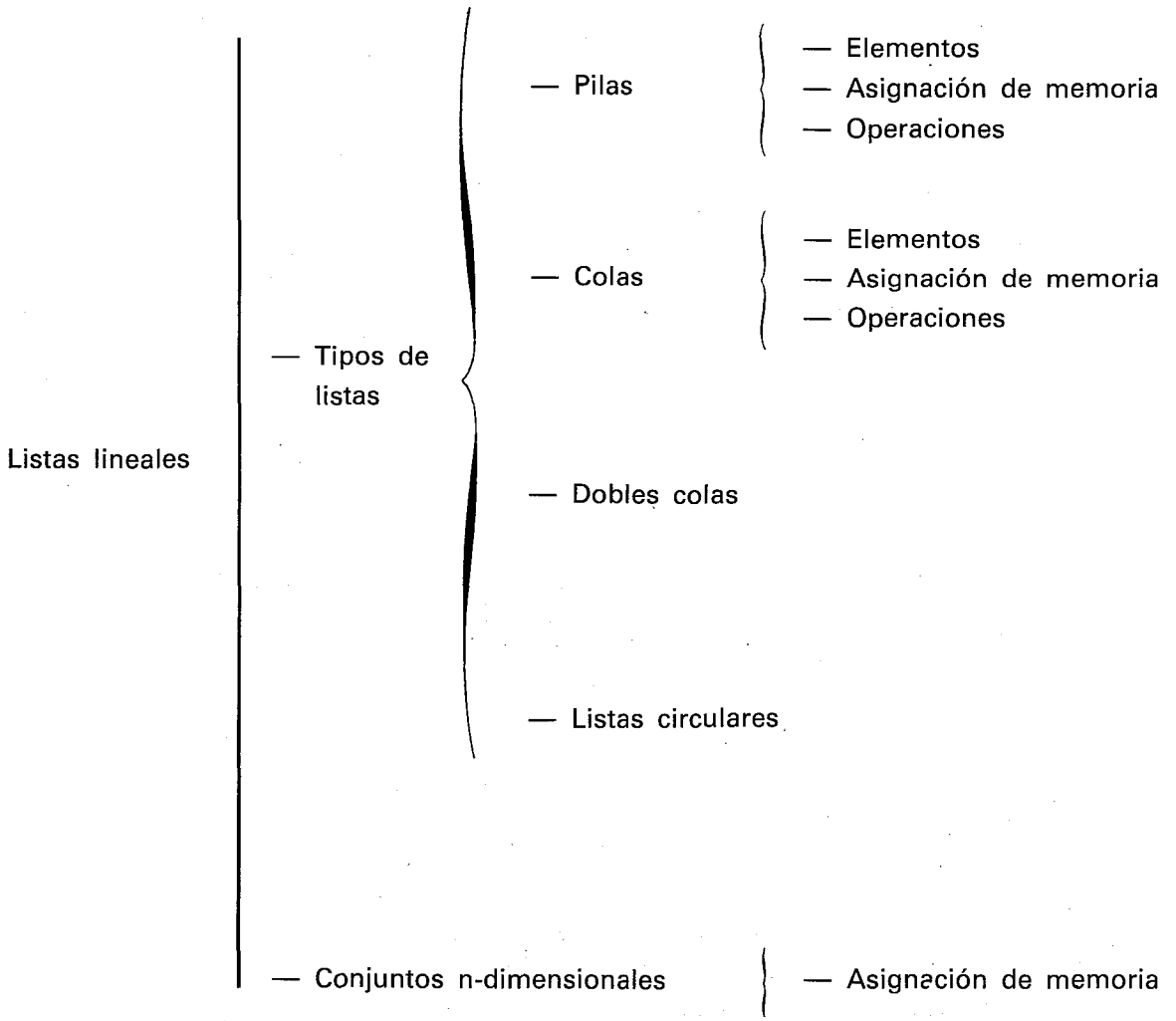
   a) La asignación secuencial.
   b) La asignación encadenada.
   c) Cualquiera de ellas.

6.

MANEJO DE DATOS (II)

Listas lineales

CUADRO RESUMEN DE LA LECCIÓN



En una tabla debe considerarse el tipo de estructura en que están representados los datos, es decir, el tipo de relaciones entre los datos de la misma. Una de estas estructuras de datos es la *lista lineal.*

Una *lista lineal* es un conjunto ordenado de elementos, llamados *nodos,* que se relacionan entre sí de la misma forma que los elementos de una tabla con un subíndice *(vector).*

La diferencia entre un *nodo* y un elemento de una tabla de un subíndice reside en que el nodo puede estar formado por un conjunto de valores numéricos, mientras que un elemento de una tabla es un sólo valor numérico.

Podemos definir una lista lineal como aquella estructura cuyas propiedades más importantes son las posiciones relativas de los *nodos.* Es decir, lo que interesa es la posición de un registro, saber cuál es el primero o cuál el último.

Esta estructura es de una dimensión (de ahí la denominación de lineal). El término que precede al registro k es el $k - 1$, el posterior es el $k + 1$, etc.

La asignación de memoria en una lista lineal puede ser en las dos formas descritas: secuencial o encadenada.

Mediante la asignación secuencial, para almacenar la lista se emplean posiciones de memoria consecutivas. Mediante la asignación encadenada, los nodos no están almacenados secuencialmente, pero los datos dentro de cada nodo están almacenados secuencialmente en posiciones de memoria. Con cada nodo se almacenan uno o más punteros que indican la primera posición de memoria utilizada por el nodo siguiente de la lista lineal. Se utilizará una asignación u otra según el tipo de lista.

6.1. TIPOS DE LISTAS

Las diversas listas que pueden existir se clasifican en orden a las operaciones más comunes que se hagan en ellas. Según el modo y lugar en que se dan las inserciones y supresiones, las listas se pueden clasificar en:

— Pilas.
— Colas.
— Dobles colas.
— Listas circulares.

6.1.1. Pilas

La pila es una lista en la que todas las inserciones y supresiones se hacen por el mismo extremo. De este modo la pila crece y decrece de tamaño en un solo sentido.

Elementos en las pilas

Una pila puede simbolizarse por un esquema como el de la figura 6.1.

En esta figura 6.1. se han mostrado los registros con rectángulos. Cada uno lleva para su identificación un número de R1 a R5 (para el caso representado en la figura de una pila de 5 elementos).

El registro que está en el lado de la pila por donde se insertan o se eliminan registros de la pila es el llamado *cabeza* de pila.

La *base* de la pila es el nombre que se le da al registro más inferior. El registro *cabeza* es el primero que se eliminará en

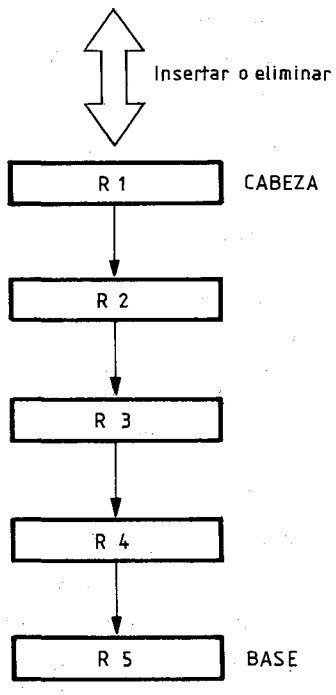


Fig. 6.1. Esquema de una Pila. ▶

una operación de eliminación, en cambio la *base* esperará a que se hayan eliminado todos los anteriores antes de eliminarse.

Cuando llega un elemento nuevo a la pila se coloca delante de la *cabeza* y queda convertido automáticamente en *cabeza,* pasando la cabeza antigua a segundo lugar.

Para el manejo de la lista son necesarias al menos dos variables:

a) La variable que indica en qué dirección está almacenada la cabeza.

b) La variable que indica en qué dirección está almacenada la base.

Asignación de memoria

Para las pilas, tanto el método secuencial como el método encadenado son válidos.

Para la construcción de una pila según el método secuencial son necesarias dos variables.

— Una variable que indica el lugar del primer elemento almacenado (que se llamará PRIM)

— Otra variable que indica la longitud del registro (que se llamará LONG).

La posición donde se debe colocar el siguiente registro (POS) se obtiene sumando el valor de la variable que indica el primer elemento y el producto de la variable que indica la longitud del registro por el número del registro que se quiere almacenar (K), disminuido en una unidad.

$$POS = LONG * (K - 1) + PRIM$$

Así, si es el segundo registro se sumará el valor de la variable que indica el primer elemento y la que indica la longitud de registros.

$$POS = LONG + PRIM$$

Si PRIM es la dirección 001 y LONG es 10 palabras, POS será la dirección 011.

Una vez insertado el registro en esa posición calculada, se cambiarán los valores que identifican la pila.

La variable que indica la cabeza señalará a la posición donde se ha insertado este nuevo registro. La variable que indica la base mantendrá su valor sin variarse (ya que la base no se ha modificado).

En cuanto a la asignación encadenada aparece el uso de una nueva variable y de un campo. Se necesita:

— Una variable que siempre indique una posición de memoria libre (donde se pueden ir añadiendo elementos a la pila).

— Un campo al final de cada registro que indique la dirección donde está almacenado el siguiente registro, suele llamarse *puntero de enlace* o *enlace.* En el caso del último registro, que no está encadenado con ningún otro, se coloca como contenido de ese campo un valor que indicará que es el último, por ejemplo un cero. La condición que debe tener ese valor es que no señale a ninguna dirección de memoria.

El proceso que se sigue para añadir un elemento a la pila es como sigue:

— Buscar la nueva posición de memoria donde se va a incluir. Esto se realiza leyendo el valor de la variable que indica la posición de memoria libre.

— Insertar el nuevo registro.

— Actualizar el valor de la variable con otra posición de memoria libre.

— Modificar el puntero que señala la cabeza con la dirección donde se ha colocado este registro nuevo (este registro es el que se convierte en cabeza).

— Enlazar el registro anterior con el nuevo mediante el puntero de enlace.

— El puntero que indica la base no ha de modificarse porque no ha variado.

Las precauciones que se deben tener al crear más de una pila secuencial son que una pila no *colisione* con otra. El término colisión tiene el significado de que una pila no se almacena dentro de la zona de almacenamiento de la otra.

Cuando sólo hay dos pilas se evita la colisión haciendo crecer las pilas en sentidos diferentes dentro de la memoria, unas crecen ocupando posiciones con un número de dirección más pequeño y las otras crecen ocupando posiciones que tengan una dirección creciente (Ver fig. 6.2.).

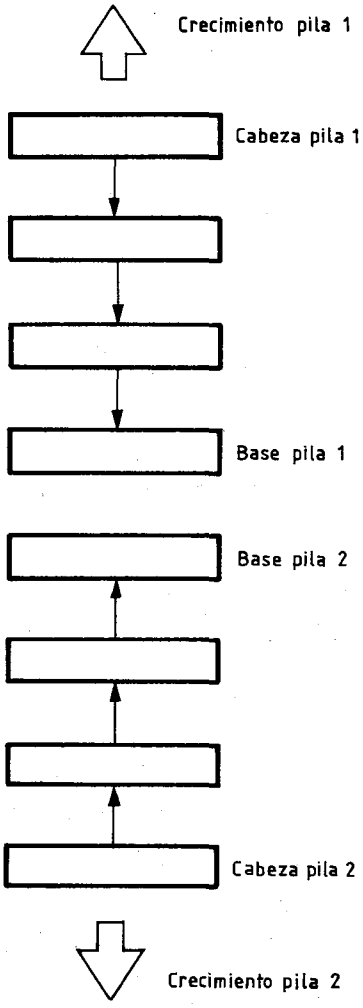


Fig. 6.2. Crecimiento de las Pilas para evitar la colisión.

Antes de efectuar una operación de este tipo en la tabla se debe comprobar que no se ha sobrepasado la capacidad que se tenía asignada a la tabla. En el caso límite de capacidad no será posible una inserción y el programa proporcionará una señal de error.

Operaciones en las pilas

Por la constitución de la pila la inserción o eliminación de un nuevo registro no puede hacerse en cualquier lugar de esta. Debe hacerse siempre en la

cabeza. El proceso de inserción se ha comentado en el apartado anterior, para cada uno de los tipos de asignación.

Eliminación

En este punto se comenta la operación de eliminar un registro.

En el caso secuencial

El registro que se debe eliminar se ha llamado cabeza de la pila.

Las fases de la operación son:

— *Obtener la dirección de la cabeza. (Leyendo la variable de cabeza.)*

— *Eliminar este registro.*

— *Actualizar la variable que indica la cabeza. Se le coloca la dirección del registro siguiente a la cabeza, que se calcula de la fórmula que nos da la dirección.*

Cabeza = Cabeza − Longitud registro.

En la fórmula se ha hecho uso de la palabra cabeza como nombre de la variable que designa la posición de la cabeza. También se utiliza el nombre *longitud registro* para designar la cantidad de palabras que contiene un registro, y que está almacenado en otra variable de la pila.

Es imposible eliminar registros de una pila que esté vacía. Esta condición debe comprobarse antes de que entre en funcionamiento el algoritmo que elimina registros. La última ocasión en que puede eliminarse un registro en una pila es cuando la variable designa la base. (Esto es cuando la base y la cabeza son un mismo registro.)

En el caso encadenado

Después de verificar la posibilidad de que la operación se efectúe, la inserción requiere la ejecución de los siguientes pasos:

— Localización del registro cabeza, leyendo la variable que lo indica.

— Eliminación del registro.

— Actualización del puntero de posiciones de memoria libres, apuntando

a la dirección de donde hemos eliminado el registro que ahora queda libre.

— Cambio del enlace.

— Actualización de la variable cabeza, señalando al siguiente registro.

— Eliminación del enlace con el registro eliminado. Quitamos del campo del puntero enlace la dirección de la antigua cabeza y colocamos en su lugar el valor cero (o el considerado para indicar que no hay enlace).

Acceso al registro que ocupa el lugar K

En el caso secuencial se empezará por el primer registro (la cabeza). Se calculará la posición de memoria en que se encuentra, de acuerdo con la fórmula de direccionamiento, el registro que se contiene en esa dirección.

En el caso encadenado se necesita una variable de enlace. Esta variable toma el valor del siguiente registro.

Se inicializa la variable con la dirección del primer registro. Si el campo donde está el puntero de enlace y se coloca como nuevo valor de la variable. Con este nuevo valor se explora el siguiente registro. Este proceso se repite k veces, momento en el cual la variable señala al registro que nos interesa.

Combinar varias listas para formar una

En este proceso la secuencia de operaciones depende del modo en que se deba efectuar la combinación. Si sólo se trata de colocar una lista detrás de otra el problema es el de creación de una lista a partir de una posición de memoria. Este problema ya está resuelto en apartados anteriores. La posición de memoria a partir de la cual debe crearse la lista es justo la siguiente a donde acaba la anterior lista. En el caso secuencial vendrá dada por un número, que indica la dirección; mientras que en caso encadenado se localizará por el puntero. Una vez creadas todas las listas se tendrá que conservar la cabeza de la primera lista y la base de la última eliminándose todas las demás cabezas y bases.

Sea por ejemplo, una pila en la que la cabeza se ha almacenado en la posición 003 y la base en la posición 103. El n.º de palabras que contiene cada registro es 10. Calcular el número de registros de que dispone esa pila:

N° de direcciones utilizadas:

$$= 103 - 003 = 100 \text{ direcciones}$$

N.° de registros en la tabla:

$$\frac{100 \text{ direcciones}}{10 \text{ palabras/registro}} = 10 \text{ registros.}$$

De modo que se trata de una lista con 10 registros: R1, R2 ... R10.

Un caso más general de unión de varias pilas es aquel en el que se hace una previa clasificación, mediante un campo, de todas las listas y una vez clasificadas se unen. El proceso que se sigue para unir las dos listas es:

Se leen los elementos primeros de ambas listas. Se comparan y se colocan en una lista nueva en el orden que proceda según el resultado de la comparación.

Determinación del número de registros

Esta operación se hace, en una estructura secuencial, a partir de las direcciones de los registros que son base y cabeza de la pila, una vez conocido el número de palabras que posee cada registro.

Puesto que todos los registros están almacenados en posiciones consecutivas, el número total de direcciones que se utilizan para el almacenamiento de la lista se calcula de la siguiente forma:

$$|\text{Dirección de la cabeza} - \text{dirección de la base}|$$

Se toma el valor absoluto (representado por las rayas verticales) de este número, para evitar que sea negativo.

El resultado de esta operación es el número de palabras que utiliza la memoria para el almacenamiento de la tabla. A partir de ese número se calcula el número de registros que existen en la tabla. Este número se halla a partir del dato de las palabras que posee cada registro.

N.° de registros = N.° palabras/N.° palabras por cada registro.

En una estructura encadenada el número de registros se calcula mediante un programa de más pasos. Este programa va explorando cada uno de los

registros, más exactamente los punteros de enlace, y va incrementando el contador de registros cada vez que ha acabado de explorar uno. El número almacenado en la variable que se ha llamado contador de registros nos da el número de estos, existentes en la tabla.

6.1.2. Colas

Se denomina cola a una lista lineal en la que las inserciones se hacen en un extremo y las supresiones en el otro. La cola crece por un lado y decrece por el otro.

Elementos de las colas

Las colas se pueden representar gráficamente por el esquema de la figura 6.3.

El registro en el extremo donde se eliminan registros se llama *frente*. El registro en el extremo donde se insertan registros se denomina *final*.

Estos dos registros son los importantes de la cola y deben tenerse unas variables cuyo valor sea el contenido, sea la dirección de la posición de memoria donde se almacena cada uno de ellos.

Asignación de memoria

En las colas la asignación puede ser secuencial y encadenada. Teniendo las mismas características que para las pilas.

Se deben conducir los punteros a las direcciones siguientes.

Operaciones en las colas

Las operaciones en las colas son las siguientes:

— acceso al término que ocupa el lugar K
— inserción de un registro
— supresión de un registro.

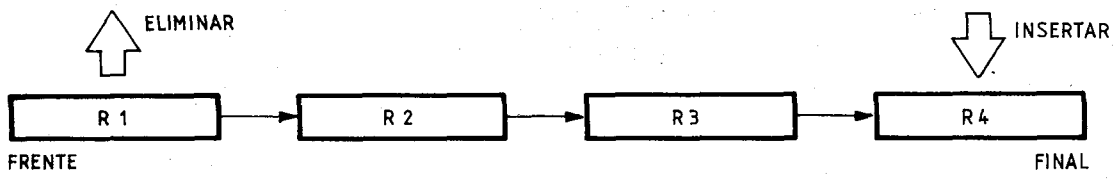


Fig. 6.3. Esquema de una Cola.

Acceso al término que ocupa el lugar K

En la asignación secuencial se emplea el mismo procedimiento que para las pilas. Ahora bien, en lugar de empezar por el registro que allí se llamaba cabeza, se empleará el registro que se llama *frente.* La fórmula a emplear es la misma que se empleó en las pilas, pues es la fórmula de todas las listas lineales.

Inserción de un registro

Sólo se puede insertar en el *final.* Por ello se debe utilizar el puntero que indica a este registro final. En el caso secuencial sólo se cambiará el puntero final, que ahora apuntará al nuevo registro introducido. En el caso encadenado se cambiará el puntero final y además se actualizarán los punteros de enlace entre un registro y otro. Al igual que en las listas se debe tener un puntero que apunte a la dirección de memoria libre, con el fin de que sea ahí donde se inserte el nuevo registro.

Supresión de un registro

Por la definición de cola sólo se puede suprimir el registro denominado final. Una vez eliminado el siguiente registro es el que se convierte en final.

Los pasos que se deben dar para conseguir la eliminación de un registro son los mismos que los comentados en las listas. En este caso se emplea el puntero final en lugar del cabeza que se usó en la pila.

En esta estructura se debe tener en cuenta también los casos en que por una inserción se supere el número máximo de elementos que puede tener la

cola o la realización de una eliminación cuando no exista ningún elemento en la cola. Otra vez la condición para este último caso es que el puntero de *frente* y el puntero *final* apunten a la misma dirección de memoria.

6.1.3. Dobles colas

En una doble cola las inserciones y eliminaciones pueden hacerse tanto en un extremo como en el otro. La doble cola es el tercer tipo de lista lineal cuando se hace la clasificación según el lugar de inserción y eliminación de los registros.

La doble cola se puede presentar gráficamente como en la figura 6.4. Los registros extremos son ahora llamados *derecha* o *izquierda* por distinguir de algún modo un extremo del otro.

La doble cola se trata del mismo modo que se hacía con las pilas, ya que tiene las características de pilas y de las colas. En lugar de tratar la cabeza se trata ahora la izquierda o la derecha según corresponda.

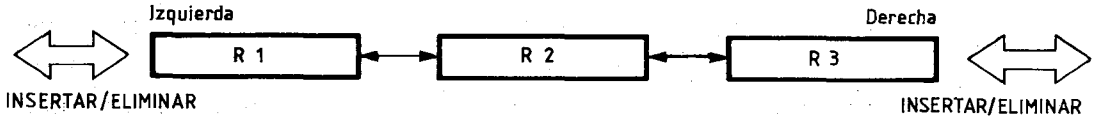


Fig. 6.4. Esquema de una Doble cola.

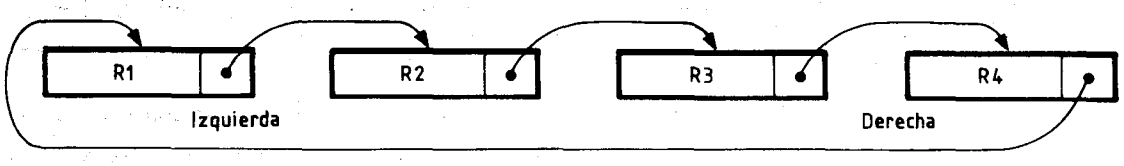


Fig. 6.5. Estructura de una lista circular.

6.1.4. Listas circulares

El nombre completo es el de *lista circular encadenada.* Es una variante de la lista encadenada. Se consigue esta estructura a partir de la lista encadenada cuando se enlaza un extremo con el otro. Es por esta razón por la que recibe el nombre de lista circular, no tiene ningún extremo abierto. La idea de izquierda y derecha no puede conservarse para esta lista. Por ser la lista circular, lo único que se puede hacer es colocar dos punteros en dos puntos adyacentes, uno sería la izquierda y el otro la derecha. Podemos colocar también un puntero de entrada a la lista.

La estructura de lista circular tiene un diagrama como el de la figura 6.5.

6.2. CONJUNTOS N-DIMENSIONALES

Una generalización de las listas lineales es el conjunto n-dimensional. Este puede considerarse como un conjunto de listas lineales en el que cada registro del conjunto pertenece a n listas lineales.

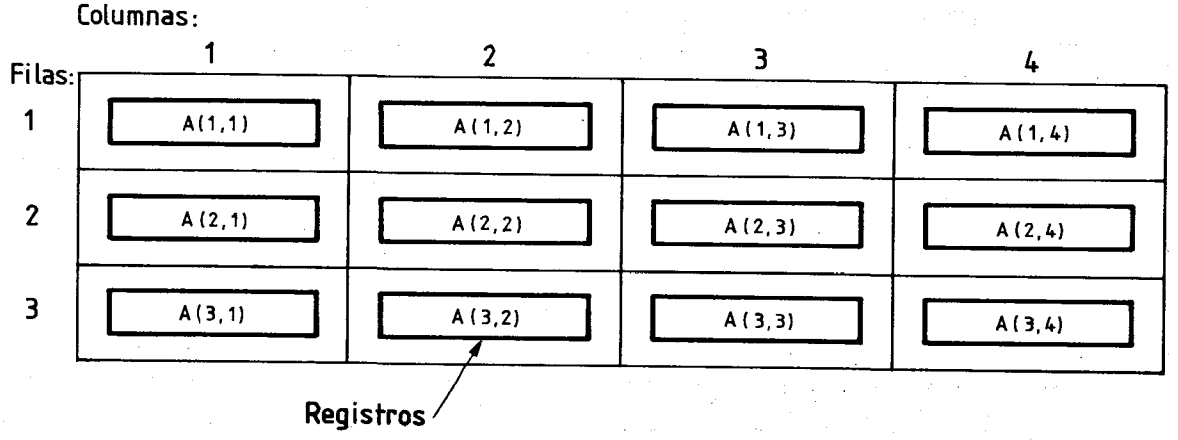


Fig. 6.6. Conjunto n-dimensional.

Consideramos n como representante de cualquier número natural (1, 2, 3 ...). El caso más claro para visualizarse es el de n = 2, o conjunto de 2 dimensiones. Puede considerarse este conjunto como rectángulo formado por una cuadrícula en el que en cada intersección existe un registro (Fig. 6.6).

Si se observa la figura se ve que un registro puede considerarse como perteneciente a dos listas lineales, una de las filas y otra de las columnas.

El registro nombrado como A(2,1) pertenecerá a la lista formada por los elementos de la fila 2:

A(2,1) A(2,2) A(2,3) A(2,4)

y también a los elementos de la columna 1:

A(1,1) A(2,1) A(3,1)

En general, en un conjunto de dos dimensiones el total de tablas lineales que se deben formar para su representación es tantas como filas haya, más tantas como columnas haya en dicho conjunto.

La expresión es m + n, siendo m el número de filas y n el número de columnas.

Ejemplo:
En el conjunto de la figura el número de listas lineales necesarias para la representación de todos los elementos es: 3 + 4 = 7 listas lineales.

Para el caso más general de un conjunto de n dimensiones no se tiene una representación gráfica de la tabla, salvo en el caso de n = 3 en el que la representación es un ortoedro en el espacio. El conjunto se puede representar por un conjunto de listas lineales.

6.2.1. Asignación de memoria

La asignación es la misma que para una lista lineal puesto que tiene una representación mediante listas lineales. Es decir, puede ser de dos tipos:

— asignación secuencial
— asignación encadenada.

Asignación secuencial

La expresión que calcula la posición de un registro en un conjunto de dos dimensiones es:

Posición de $A(I;J) = a_0 + a_1 . I + a_2 . J$

Esta expresión significa que el registro que ocupa la fila I y la columna J en la representación gráfica de la tabla de dos dimensiones se almacena en una posición que depende de I y de J. Los valores a_0, a_1 y a_2 en la fórmula son valores constantes. (a_0 representa la posición de memoria a partir de la cual se empieza a almacenar el conjunto de dos dimensiones).

Ejemplo

Calcular la posición en la que se almacenará el elemento de la fila 3 y de la columna 2. La posición del primer elemento de la tabla es la 001 y las constantes a_1 y a_2 son 15 y 2 respectivamente.

Posición $A(3,2) = 001 + 15 \times 3 + 2 \times 3 = 1 + 45 + 6 = 052$

En el caso general de una tabla de n dimensiones, la expresión conserva la misma forma que en el caso anterior, añadiéndole un sumando por cada una de las dimensiones de que consta. Estos sumandos añadidos constan igualmente del producto de una constante y del índice de la dimensión.

La posición del elemento A(3,5,8,3) será, según lo visto en este párrafo:

Posición de A(3,5,8,3) = Posición de A(1,1,1,1) +

$$+ a_1 \times 3 + a_2 \times 5 + a_3 \times 8 + a_4 \times 3$$

donde se ha considerado que el primer elemento de la tabla es el A(1,1,1,1).

Asignación encadenada

En este caso cada uno de los registros deben tener n campos de enlace, uno para cada uno de las listas a que pertenece.

Para ver un caso práctico de cómo se enlazan unos registros con otros se presenta el ejemplo siguiente.

Sea un conjunto de 6 personas. Cada registro tiene la información de una persona. Los campos que posee son: sexo, edad, ojos y color de cabello.

Esta información puede presentarse como una tabla en la que cada persona pertenece a una intersección de 4 listas lineales, una para el sexo, otra para la edad, otra para el color de ojos y otra para el color del cabello. El modo en que se constituyen los enlaces es efectuar un enlace de todos los registros que tengan una característica común. Estarán enlazados todos los registros que tengan el mismo color de pelo, también lo estarán todos los que tengan el mismo sexo, y así sucesivamente hasta completar los cuatro enlaces (Fig. 6.7).

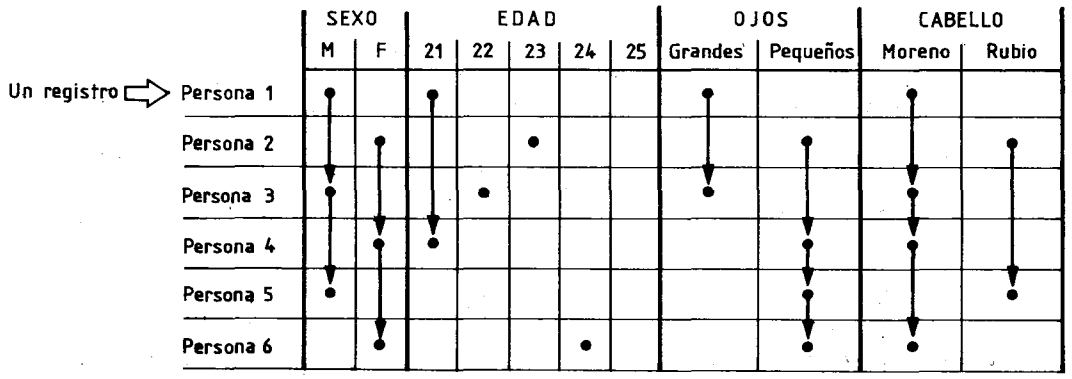


Fig. 6.7.

EJERCICIOS

1. Las listas se clasifican en .....

2. Un elemento nuevo en una Pila se coloca:

   a) Delante de la cabeza.
   b) Detrás de la cabeza.
   c) El último de la Pila.

3. Para el manejo de una lista son:

   a) Necesarios al menos dos punteros.
   b) Necesarias al menos dos variables.
   c) Necesarias dos direcciones.

4. Según el método secuencial, la suma de la dirección del primer registro al producto de la longitud de cada registro por el número de registros, indica .....

5. El último campo de un registro en método encadenado indica:

   a) El final del registro.
   b) La dirección del siguiente registro.
   c) La longitud del siguiente registro.

6. ¿Cuándo no es posible una nueva inserción en una Pila? .....
.....

7. En la asignación encadenada, la localización de la cabeza de una lista viene dada por:

   a) Un número.
   b) Un puntero.
   c) La base de la lista.

8. Una lista lineal, cuando las inserciones se hacen en un extremo y las supresiones en el otro extremo:

   a) Se denomina Cola.
   b) Se denomina Pila.
   c) Se denomina Cabeza o Base.

9. Las características de asignación de las Colas son:

   a) Secuenciales o encadenadas.
   b) Las mismas que para las Pilas.
   c) Dependen del tipo de Cola.

10. En una Cola, los registros se pueden suprimir e insertar:

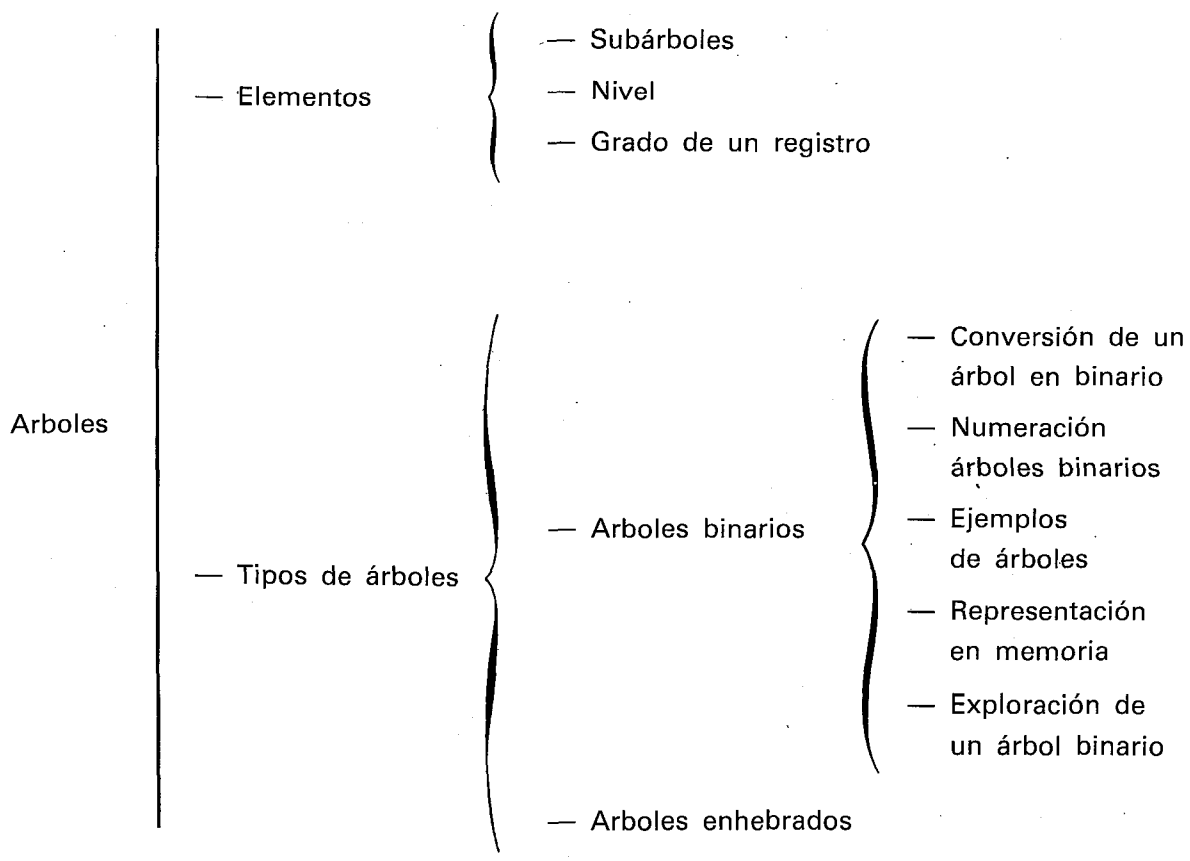
    a) Por el principio.
    b) Por cualquier extremo.
    c) Por el final.

7.

MANEJO DE DATOS (III)

Arboles

CUADRO RESUMEN DE LA LECCIÓN



Los elementos de una tabla pueden estar relacionados jerárquicamente mediante una estructura en *árbol.* Una estructura en árbol implica una relación de ramificación entre los elementos (llamados *registros* o *nodos*). Así pues, un árbol se compone de registros y enlaces (relaciones) entre los registros del árbol.

El *árbol* es una de las estructuras de información más importantes. Además de ser una estructura de almacenamiento de datos, muchos algoritmos hacen uso de una estructura de este tipo para su desarrollo. Una condición tiene varias posibilidades y a su vez, cada una de éstas lleva a otras condiciones. Así, sucesivamente, se van ramificando.

El árbol es una estructura no lineal. Esto quiere decir que un registro no sigue al otro como en las listas lineales sino que un registro sigue a más de un registro.

7.1. ELEMENTOS DE LOS ÁRBOLES

Se puede definir un árbol como un conjunto de registros (nodos). En este conjunto se consideran dos puntos:

1. Existe un registro especial llamado *raíz.*
2. Todos los registros restantes forman varios conjuntos que no se mezclan entre sí y que por sí mismos tienen estructura de árbol.

En esta definición está implícita la idea de ramificación al decir que un árbol está formado por otros árboles. La ramificación es la característica más importante de los árboles.

La representación de un árbol se muestra en la figura 7.1. donde se aprecia la *raíz* que origina el árbol y los distintos conjuntos que parten de ella.

7.1.1. Subárboles

Se llama *subárbol* de una raíz a cada uno de los árboles que parten de esta raíz. Un subárbol tiene las mismas propiedades que un árbol.

La idea de subárbol es muy importante para la exploración de un árbol. Como que un árbol está formado por subárboles, un algoritmo de exploración de un árbol es la exploración de los distintos subárboles hasta encontrar la raíz de todo el árbol, momento en que la exploración se acaba.

7.1.2. Nivel

Se denomina *nivel* al conjunto de los registros que existen en cada ramificación.

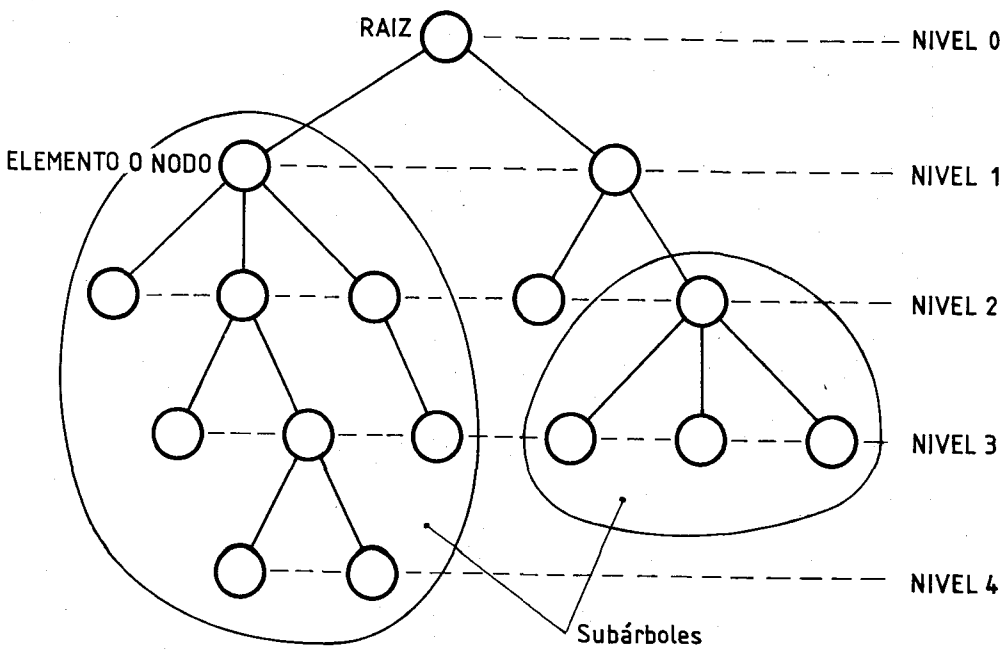


Fig. 7.1. Estructura de un árbol.

El nivel cero está constituido siempre por un solo registro, la raíz. Los niveles se numeran sucesivamente y en orden creciente a partir de este número cero.

El siguiente nivel (nivel 1) está formado por todos los registros que salen de la raíz en su primera ramificación. Los niveles siguientes estarían formados por los registros que aparecen de las ramificaciones de los registros del primer nivel. Así, se irían desarrollando sucesivamente los niveles hasta que no existiesen más ramificaciones.

7.1.3. Grado de un registro

El *grado de un registro* es el número de subárboles que existen a partir de ese registro. Existirán registros de grado cero, en los que no se ramifica más el árbol; registros de grado 1, que sólo tienen un subárbol, etc.

Un registro de grado cero se denomina *registro terminal* u *hoja del árbol.* En contraposición, se denomina *rama de un árbol* a un registro no terminal.

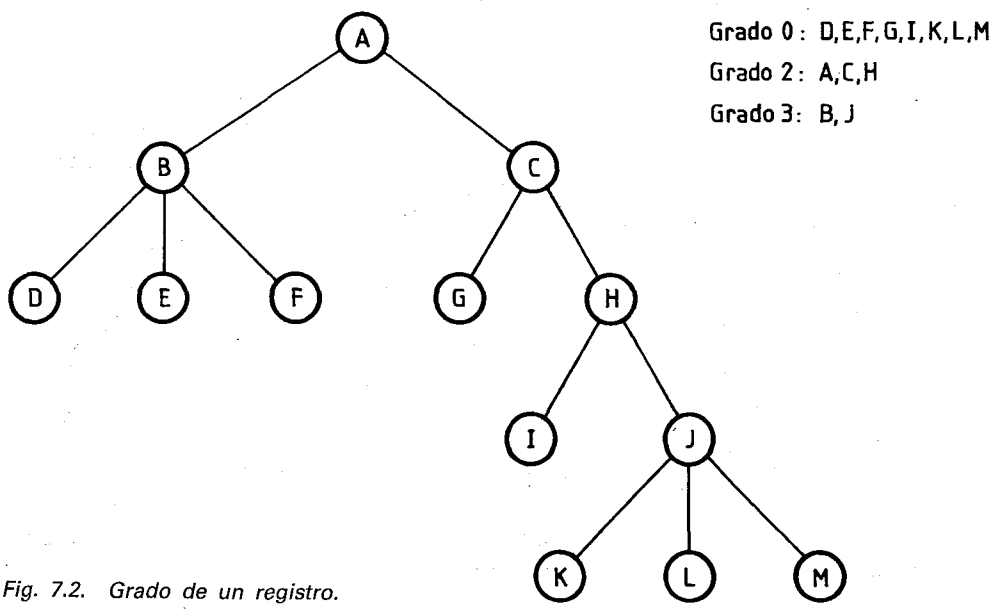


Fig. 7.2. Grado de un registro.

7.2. TIPOS DE ÁRBOLES

Una primera clasificación es aquella en la que se tiene en cuenta el grado de los registros del árbol.

Según este criterio los árboles se pueden clasificar en árboles binarios y árboles enhebrados (o no binarios).

Esta clasificación resalta la importancia de los árboles binarios. Se verá cómo se puede convertir un árbol cualquiera en binario. Este hecho explica que los árboles binarios se estudien con más profundidad que cualquier otro tipo.

7.2.1. Árboles binarios

Un árbol binario es aquel en que todos sus registros tienen grado uno o dos, salvo las hojas, que tienen grado cero.

La representación gráfica de un árbol binario se muestra en la figura 7.3. En dicha figura se ve cómo cada registro sólo tiene dos ramas como máximo, una a la izquierda y otra a la derecha. Así, un registro siempre tiene dos subárboles, y desde un registro se puede llegar a otros dos, que se denominarán en el texto registro izquierdo y registro derecho.

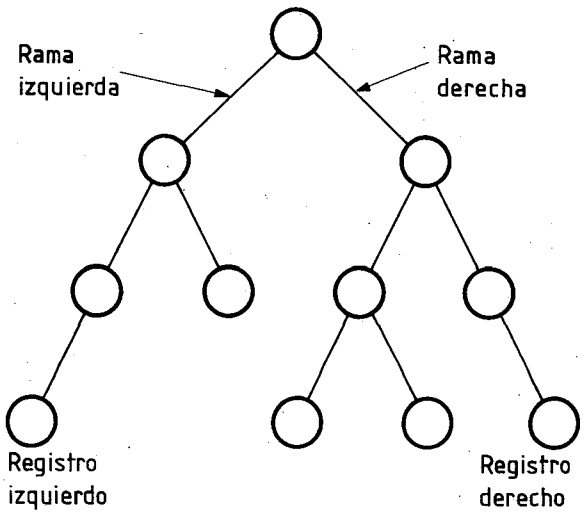


Fig. 7.3. Arbol binario.

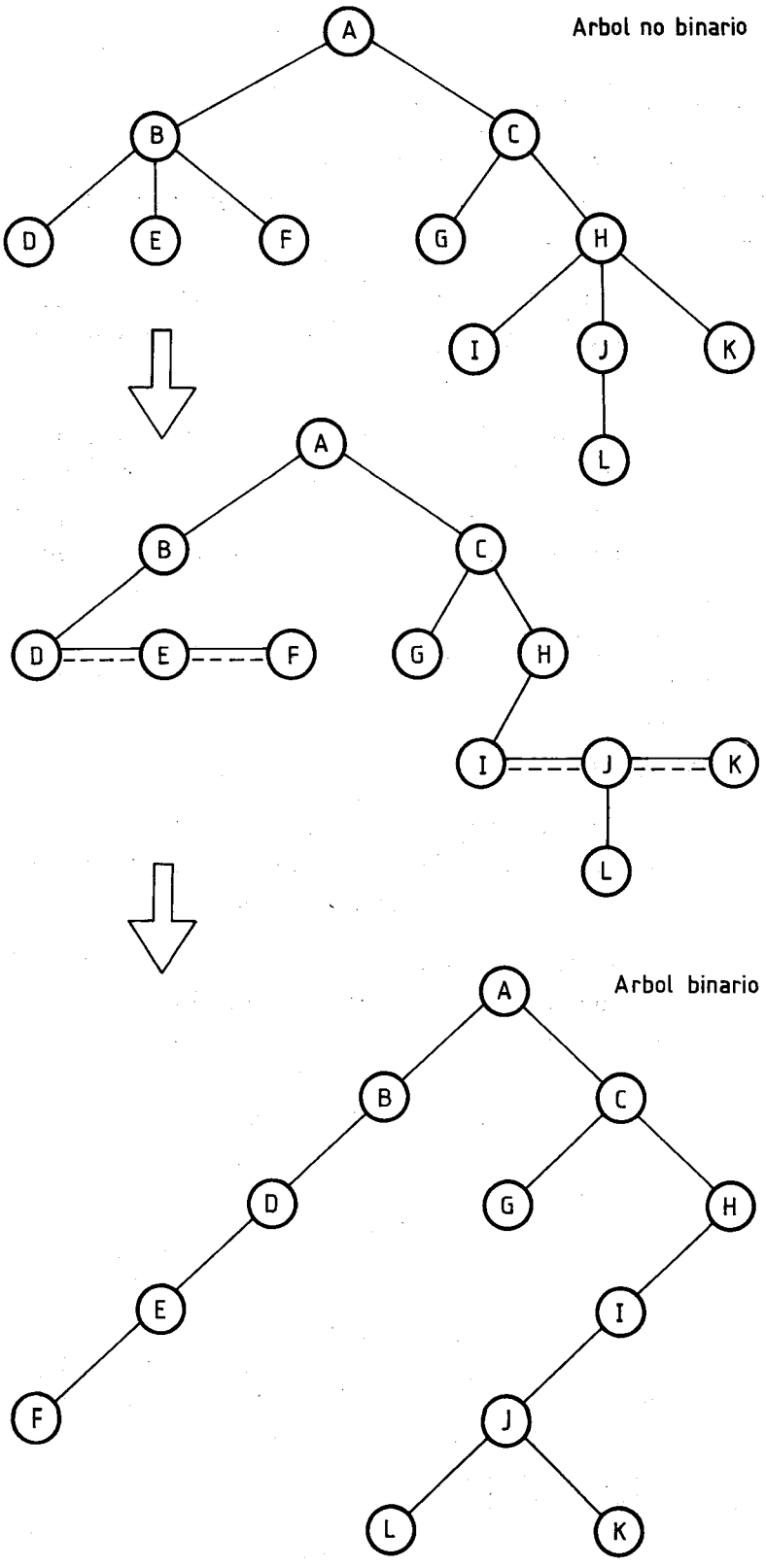


Fig. 7.4. Conversión de un árbol no-binario en árbol binario.

Conversión de un árbol no binario en binario

Un árbol no binario siempre puede convertirse en uno binario. Para ello se eliminan las ramas que sobrepasen el número de dos y se forma con ellas un nuevo nivel, del árbol binario.

La condición que debe cumplirse para que la conversión se efectúe correctamente es que el nuevo árbol tenga el mismo encadenamiento de registros que tenía el anterior.

El método de conversión consiste en enlazar todos los hijos de cada familia y eliminar los enlaces verticales con el padre excepto en el primer hijo, el más a la izquierda. (La nomenclatura «padre-hijo» parece conveniente aquí para representar que un padre (registro) de un nivel tiene varios hijos (ramas) en el nivel siguiente).

El método completo se muestra paso por paso en la figura 7.4.

Numeración de los árboles binarios

Se debe dar un nombre a cada uno de los registros que contiene el árbol. Este nombre o numeración sirve para identificar los registros de que se compone el árbol y para relacionarlos con los demás registros.

En los diagramas suele emplearse una letra para la numeración del registro. Estas letras se van dando en orden alfabético empezando por la raíz, de izquierda a derecha y de arriba hacia abajo.

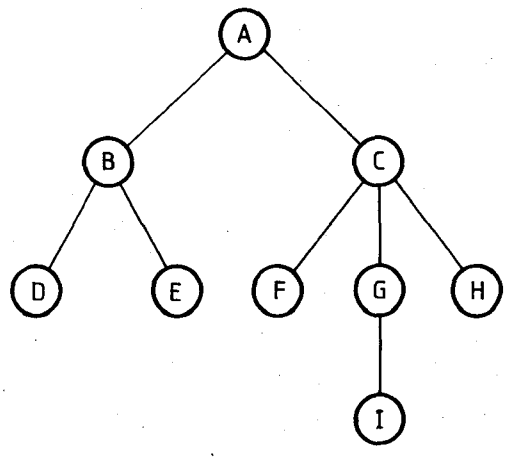


Fig. 7.5. Notación alfabética de los registros de un árbol.

Nivel 0	A : 1
Nivel 1	B : 1.1
	C : 1.2
Nivel 2	D : 1.1.1
	E : 1.1.2
	F : 1.2.1
	G : 1.2.2
	H : 1.2.3
Nivel 3	I : 1.2.2.1

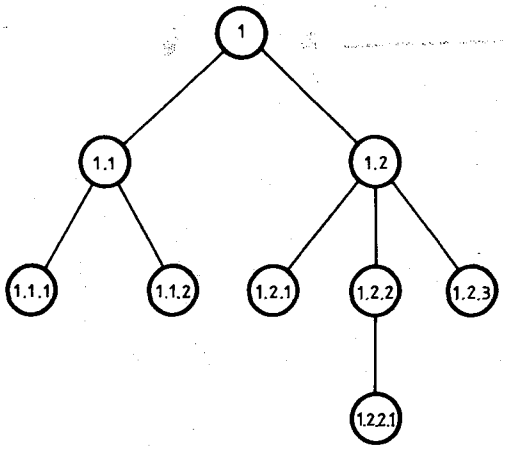


Fig. 7.6. Notación decimal de Dewey.

Una notación más conveniente es de la llamada «Notación Decimal de Dewey». En esta notación se da un número a cada uno de los registros. Este número se va ampliando nivel por nivel con una cifra más.

Para el árbol de la figura 7.5. la notación sería la que muestra la figura 7.6.

Ejemplos de árboles

Un primer ejemplo de la información estructurada en forma de árbol son los cuadros sinópticos que encabezan las lecciones de este texto. Cada una de estas lecciones se dividen en apartados, éstos en subapartados y pueden estructurarse en forma de árbol, (Fig. 7.7.).

Otro ejemplo es la representación de cadenas de caracteres que expresan expresiones matemáticas. La estructura en árbol es conveniente para conocer el orden en que se deben efectuar las operaciones que componen la expresión. Precisamente es de esta forma como la máquina almacena internamente las expresiones y conoce el orden de ejecución de sus operaciones.

La expresión (a − b (c × d + h × i)) × j podía tener un árbol como el de la figura 7.8.

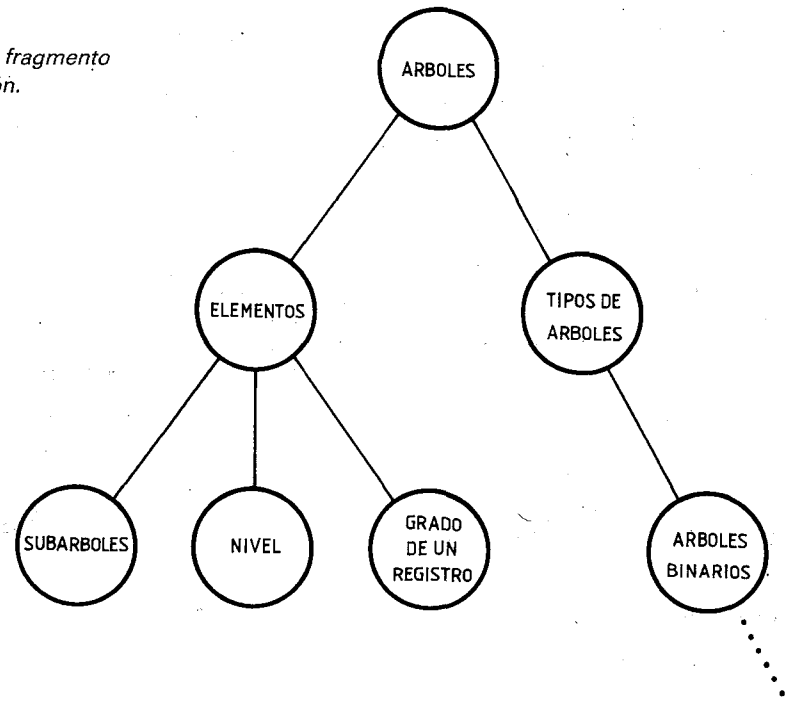


Fig. 7.7. Estructura en árbol de un fragmento del cuadro sinóptico de esta lección.

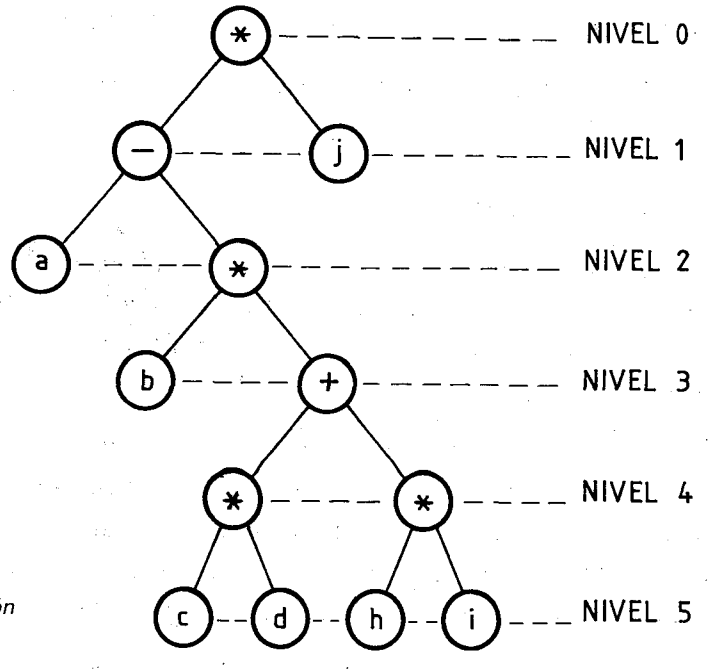


Fig. 7.8. Arbol correspondiente a la expresión

[a − b * (c * d + h * i)] * j

Para la creación del árbol es conveniente comenzar por los niveles más altos (nivel 5), es decir, por las operaciones que se deben realizar en primer lugar. En este caso son las interiores al paréntesis. Por un lado el producto $c \times d$ y por el otro, y en el mismo nivel el producto $h \times i$. (Nivel 4).

A continuación, en el nivel 4 se colocan las operaciones que se realizaron anteriormente en orden jerárquico. En el nivel 3 se hace la suma de los anteriores productos.

El siguiente paso es multiplicar el resultado de la suma anterior por *b*. Lo que constituye el nivel 2.

Se realiza la diferencia *a* menos el producto anterior. (Nivel 1). En este caso se ha expresado en el árbol el orden en que se manejan los operandos. Puesto que la diferencia no es conmutativa, *a* debe ponerse en la rama de la izquierda significando minuendo, y todo el conjunto de la rama de la derecha, es el sustraendo.

Como último paso o nivel cero, se debe realizar el producto de la diferencia realizada en el apartado anterior por *j*.

Como se ve es un árbol binario puesto que una operación tiene siempre dos operandos y un signo de operación. En cualquier subárbol, el signo de operación es el padre y los operandos los hijos. En algunos casos los operandos son registros únicos y en otros son subárboles que deben ser *operados previamente*.

Como ejemplo, en la página siguiente se presentan todos los pasos de una operación que tiene la expresión aritmética del árbol de la figura 7.9.

Resolver: $(4 - 5 \times (1 \times 2 + 2 \times 4)) \times 6$

Se debe notar que en cada paso se elimina todo un subárbol y se sustituye por un único registro, resultado de la operación.

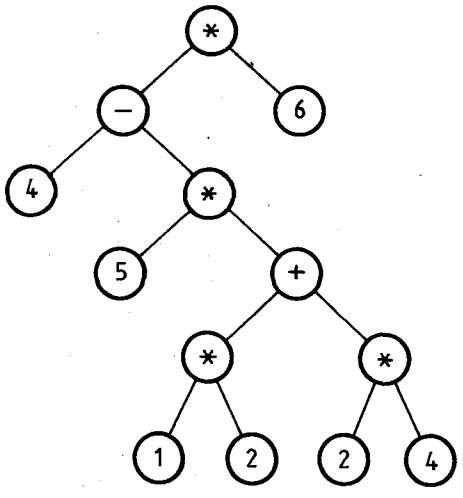
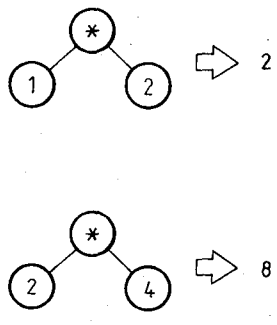
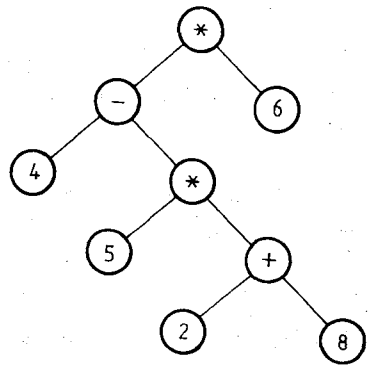
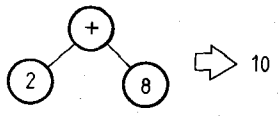
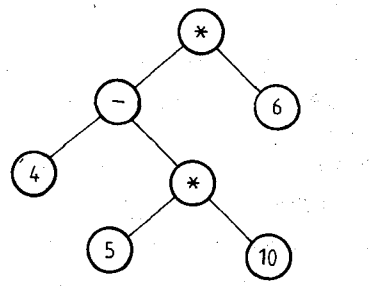
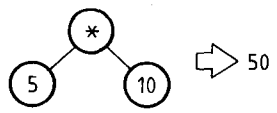
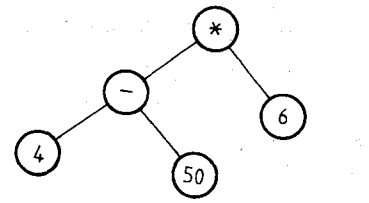
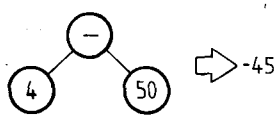
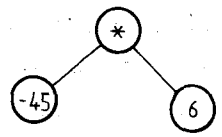
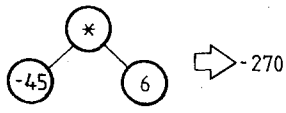


Fig. 7.9. ▶

ETAPA	OPERACION EFECTUADA	NUEVO ARBOL
1		
2		
3		
4		
5		

Exploración de un árbol binario

La operación más importante en árboles binarios es la *búsqueda* de algún registro. Esta es parte integrante de la operación de *exploración* o *recorrido del árbol,* de la que se hablará en este apartado.

Las operaciones de inserción y eliminación de registros siguen los mismos principios que en las listas:

— Búsqueda del elemento donde se va a insertar o modificar.
— Inserción o eliminación.
— Modificación de los punteros.

Los punteros indicarán la nueva dirección donde se ha colocado el registro nuevo. Este a su vez, deberá tener sus punteros señalando a sus hijos, al de la izquierda y al de la derecha, en caso de que existan.

Se denomina *exploración o recorrido de un árbol binario* a aquella operación en la que se visita cada uno de los registros del árbol una sola vez.

Distinguiremos dos modos diferentes de recorrido.

— recorrido por ramas
— recorrido por niveles.

Recorrido por ramas

En el recorrido por ramas se van visitando sucesivamente los registros, siempre pertenecientes a diferentes niveles en cada visita.

Existen varios modos de realizar este recorrido:

— Recorrido en preorden.
— Recorrido en inorden.
— Recorrido en postorden.

La diferencia entre uno y otro es el momento en que se visita la raíz.

A) Recorrido en preorden

El algoritmo para el recorrido es el siguiente:

— Visitar la raíz.
— Recorrer el subárbol izquierdo.
— Recorrer el subárbol derecho.

La raíz de la que se habla en el algoritmo es cada una de las raíces de los subárboles que se van encontrando. Los tres pasos en los que se ha dividido el algoritmo se realizan siempre que el árbol no esté vacío.

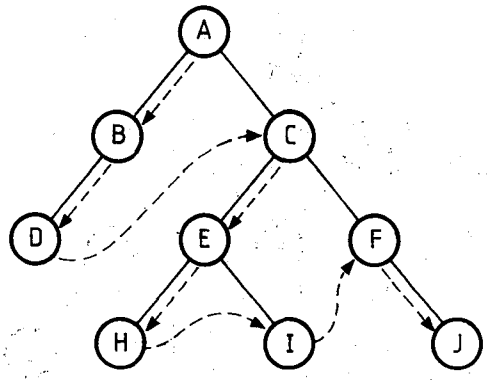


Fig. 7.10. Recorrido preorden de un árbol.

Como ejemplo, en la figura 7.10 se presenta la secuencia de los registros visitados para la exploración preorden de un árbol. Se marca con línea punteada el recorrido.

La exploración empieza en la raíz del árbol, A. Se empieza a recorrer el árbol por la rama de la izquieda. Se llega a B. Considerando a B como la nueva raíz se sigue por la rama de la izquierda. Se llega a D. El árbol está vacío y se explora la rama de la derecha. También está vacía. Se explora a continuación la rama de la derecha desde la raíz A. Se llega al registro C. La rama de la izquierda desemboca en E. Considerando esta nueva raíz, se explora la rama izquierda, llegando a H. Después, la rama de la derecha es explorada y se llega a I, acabándose el árbol. Se vuelve a considerar la raíz C y se explora el subárbol derecho, llegando a F. La rama izquierda de F está vacía. Se explora la derecha y se llega a J, donde el árbol se acaba.

La secuencia de visita a los registros es: A B D C E H I F J

B) Recorrido en inorden

En este caso la raíz se visita después de haber visitado la rama izquierda pero antes de la derecha. El algoritmo que usa es el siguiente:

— Recorrer el subárbol izquierdo.
— Visitar la raíz.
— Recorrer el subárbol derecho.

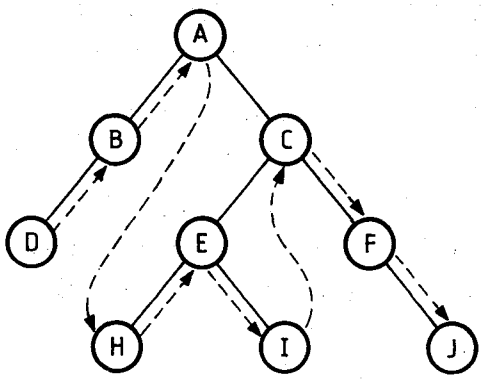


Fig. 7.11. Recorrido inorden de un árbol.

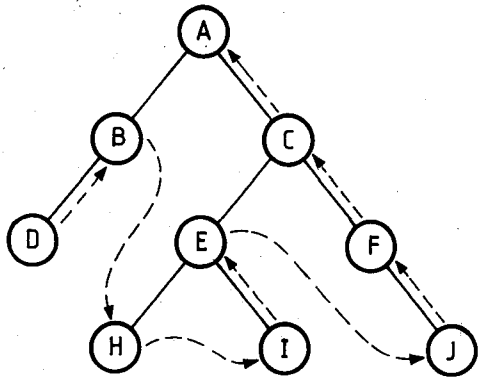


Fig. 7.12. Recorrido postorden de un árbol.

Esos tres pasos se siguen mientras el árbol no está vacío, caso en que se recorrerá sin hacer nada.

C) Recorrido en postorden

El ejemplo usado es el mismo que en los dos casos anteriores.

Se empieza a explorar por el primer registro de la izquierda D. Después de explorar la derecha, se pasa a la raíz, B. Se explora a continuación la rama derecha, empezando por el extremo más a la izquierda H. Seguidamente se visita esa rama de la derecha, I y después la raíz, E. Se explora la rama derecha. Se empieza visitando el registro J (los de la izquierda están vacíos). Se sigue por el F, constituído en raíz de subárbol, y por último se visita el C, la raíz del subárbol de la rama de la derecha. El registro visitado en último lugar es la raíz del árbol completo, A.

La secuencia en la que se ha visitado el árbol es: D B H I E J F C A

Recorrido por niveles

Una forma alternativa de recorrido de árboles es el recorrido nivel por nivel. Como indica su nombre ahora se deben seguir cada uno de estos niveles para el recorrido, en lugar de las ramas.

Se empieza por el nivel más pequeño (cero), es decir, la raíz del árbol y se visita en primer lugar. A continuación se pasa al nivel siguiente y se empiezan a visitar los registros de todo ese nivel, comenzando por la izquierda. No se pasa al nivel siguiente hasta que no se ha acabado de explorar el nivel anterior.

Como ejemplo se presenta la secuencia de visita de registros para el árbol de la figura 7.13.

La raíz se visita en primer lugar, A. Este nivel cero ya no contiene más elementos y se pasa al nivel siguiente (nivel 1). En este nivel, y empezando por la izquierda, se visita el registro B. En ese mismo nivel se visita el registro C, que es el último registro del nivel.

En el siguiente nivel se visitan sucesivamente los registros D, E, F. Una vez se ha llegado al registro F, se pasa al nivel 3, el último del árbol. En este nivel el orden de exploración es H, I, J.

La secuencia de todos los registros visitados es: D B A H E I C F J.

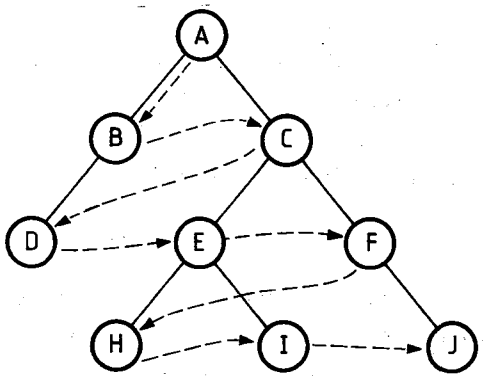


Fig. 7.13. Recorrido por niveles de un árbol.

7.2.2. Árboles enhebrados

El árbol enhebrado es aquel en que las hojas no se representan en memoria como hojas, sino que se enlazan con raíces anteriores.

Con este modo de representación en memoria, el recorrido en el árbol es mucho más rápido porque se vuelve directamente a la raíz superior que interese en el recorrido.

Estos nuevos enlaces se denominan *hilos,* para distinguirlos de los enlaces verdaderos, los que dan la estructura del árbol.

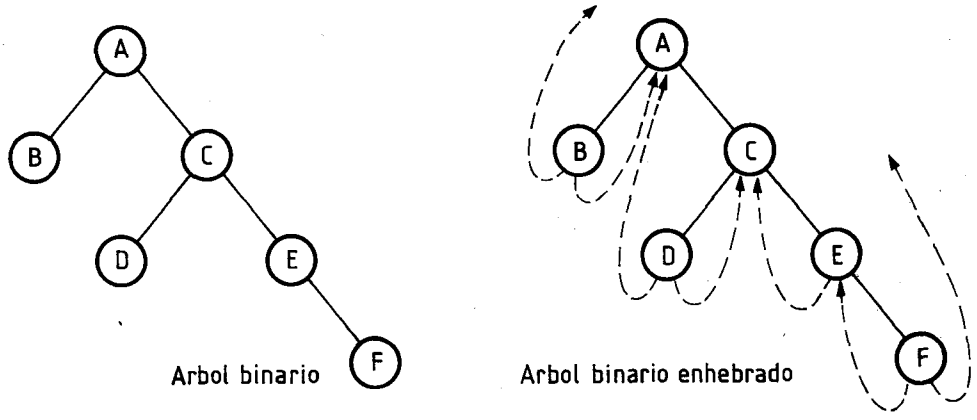


Fig. 7.14. Arbol sin hilos y con hilos.

El modo de conseguir un árbol enhebrado a partir de uno normal es colocar siempre dos enlaces en cada uno de los registros. Si un registro ya tiene los dos enlaces no se le coloca ningún hilo. A los registros que tienen un enlace se les coloca un solo hilo y a los registros hojas o registros terminales se le coloca los dos hilos. Los hilos deben enlazar con las raíces inmediatamente superiores. Si sólo se puede colocar un hilo, éste enlaza con la raíz superior al siguiente (el «registro padre»). En el caso de dos hilos uno enlazará con el registro «padre» y el otro con el registro superior siguiente al registro «padre».

Véase en la figura 7.14 un ejemplo de árbol sin hilos y el mismo ampliado con hilos.

En la representación en memoria de esta estructura de árboles se debe colocar algún campo que indique el tipo de enlace que es, si es hilo o enlace del árbol, y además otro campo que indique si el registro es hoja o no.

EJERCICIOS

1. Si un árbol es una estructura no lineal, entonces:

   a) Un registro sigue a más de un registro.
   b) No tendrá raíz.
   c) Los registros se mezclan entre sí.

2. Un árbol está formado por una raíz y:

   a) Los registros que están enlazados con la raíz.
   b) Varios subárboles.
   c) Los subárboles que están enlazados con la raíz.

3. Los niveles de un árbol son:

   a) Equivalentes a cero.
   b) Las ramificaciones de los registros.
   c) Los registros de cada ramificación.

4. El grado de un registro es .....
   .....

5. Un registro de grado cero:

   a) Se llama hoja de árbol.
   b) Es el primer registro del árbol.
   c) Es el primer registro de un subárbol.

6. Los árboles se clasifican en binarios y

   a) Terciarios.
   b) No binarios.
   c) Binarios múltiples.

7. En la notación de Dewey, los registros se representan por:

   a) Números.
   b) Letras.
   c) Letras y números.

8. La operación más importante en árboles binarios es la .....
   .....

9. El recorrido por ramas puede ser

   a) Preorden, inorden.
   b) Preorden, postorden.
   c) Preorden, inorden, postorden.

10. En un árbol enhebrado, las hojas:

    a) Se enlazan con raíces anteriores.
    b) Poseen nivel cero.
    c) Poseen nivel uno.

SOLUCIONES DE LOS EJERCICIOS

1. Ficheros.

1. *c*); 2. *c*); 3. *b*); 4. *c*); 5. *b*); 6. *b*); 7. *b*); 8. *a*); 9. *c*).

2. Ficheros de acceso secuencial. Tarjetas perforadas.

1. *a).*
2. *b).*
3. *b).*
4. El conjunto de papeles de donde se obtiene la información para la creación de un fichero.
5. *c).*
6. *b).*
7. Modo de hacer posible la recuperación de la información.

3. Ficheros de acceso directo. Cintas magnéticas.

1. *b).*
2. *c).*
3. Cada una de las zonas longitudinales en que puede considerarse dividido el ancho de la cinta.
4. *a).*
5. *c).*
6. *b).*
7. *a).*
8. Las fichas de control del programa.
9. *b).*
10. *c).*
11. *b).*
12. *a).*
13. *c).*

4. *Ficheros de acceso secuencial. Discos magnéticos.*

1. *b*).
2. Al tiempo que transcurre durante el trasvase de la información del ordenador al dispositivo o viceversa.
3. *c*).
4. *b*).
5. *c*).

5. Manejo de datos (I). Tablas

1. *a).*
2. Cómo se separan los diversos campos en un registro y qué contiene cada uno de ellos.
3. *b).*
4. Variables con subíndice.
5. *b).*
6. Dos.
7. *c).*
8. *c*).
9. Una tabla.
10. *b*).
11. *a*).
12. Las posiciones de memoria que almacenarán los elementos de una tabla.
13. $D = L_0 + C \times K$
    D = Dirección.
    L_0 = Dirección de la primera palabra del registro.
    C = número del registro buscado.
    K = número de palabras por registro.
14. *a*).
15. *b*).

6. Manejo de datos (II). Listas lineales

1. Pilas, Colas y Dobles colas.
2. *a*).

3. *b*).
4. La posición del siguiente registro.
5. *b*).
6. Al llegar a su límite de capacidad.
7. *b*).
8. *a*).
9. *b*).
10. *c*).

7. Manejo de datos (III). Arboles

1. *a*).
2. *c*).
3. *c*).
4. El número de subárboles que existen a partir de este registro.
5. *a*).
6. *b*).
7. *a*).
8. Búsqueda de registros.
9. *c*).
10. *a*).

Indice General

Contenido 7

Introducción 9

1. FICHEROS

Cuadro resumen 13
1.1. Definición 14
1.2. Gestión de ficheros 16
1.2.1. Definición de la estructura del fichero 16
1.2.2. Creación del fichero 16
1.2.3. Actualización del fichero 17
1.2.4. Interrogación del fichero 17
1.3. Organización de los ficheros 17
1.3.1. Ficheros de acceso secuencial 18
— Secuenciales sin clave 18
— Organización secuencial indexada 22
1.3.2. Ficheros de acceso aleatorio 25
— Ficheros de acceso aleatorio indexado 25
— Fichero de acceso aleatorio mapeado 27
1.4. Instrucciones de manejo de los ficheros 29
1.4.1. Las instrucciones de identificación del fichero 29
1.4.2. Las instrucciones de control del dispositivo 30
1.4.3. Las instrucciones de lectura y escritura 31
1.5. Funciones para cálculos de la dirección 31

1.5.1. Simplificación de un campo para formar la clave 32
1.5.2. Rutinas de aleatoriedad 36
— Método del resto de la división 36
1.6. Criterios de elección de ficheros 37

Ejercicios 39

2. FICHEROS DE ACCESO SECUENCIAL (I): Tarjetas perforadas

Cuadro resumen 43
2.1. Descripción física 44
2.2. Codificación 46
2.2.1. Caracteres numéricos 46
2.2.2. Caracteres alfabéticos y especiales 47
2.3. Componentes del sistema de fichas perforadas 48
2.3.1. La perforadora 49
2.3.2. Verificadora de tarjetas 50
2.3.3. Clasificadora 51
2.3.4. Interpretadora 52
2.3.5. La intercaladora 53
2.4. Sistemas de seguridad 55

Ejercicios 58

3. FICHEROS DE ACCESO SECUENCIAL (II): Cintas magnéticas

Cuadro resumen 61
3.1. Descripción física 62
3.2. Lectura y escritura 62
3.3. Descripción lógica 64
3.4. Grabación de ficheros 64
3.5. Sistemas de seguridad 65
3.5.1. Ciclo del «abuelo» 66
3.5.2. Uso del anillo protector 66
3.5.3. Uso de las etiquetas electrónicas 66
3.6. Concepto de bloqueo 68
3.6.2. Aumento de la capacidad 69
3.7. Sistemas de ordenador con cinta magnética 70
3.8. Otros sistemas de almacenamiento 72
3.8.1. Los tambores magnéticos 72
3.8.2. Las células de datos 72
Ejercicios 73

4. FICHEROS DE ACCESO DIRECTO: Discos magnéticos

Cuadro resumen 79
4.1. Descripción 80
4.2. Operaciones 82
4.2.1. Escritura 82
4.2.2. Lectura 82
4.3. Descripción lógica 84
4.3.1. Sectores 84
4.3.2. Pistas 85
4.3.3. Formatos de grabación 85
— Con registro de dirección 85
— Con registro clave 87
4.3.4. Fases de la lectura/escritura 87
4.4. Tipos de discos 88
4.4.1. Discos flexibles 89
4.4.2. Discos rígidos 89
4.5. Proceso tarjeta-disco 90
4.6. Sistema computador con discos 92
4.7. Sistemas de seguridad 92
Ejercicios 93

5. MANEJO DE DATOS (I). Tablas

Cuadro resumen 97
5.1. Variables simples 99
5.1.1. Nombre y valor de una variable 99
5.1.2. Formato de una variable 100
5.2. Vectores 104
5.3. Matrices 105
5.4. Elementos de una tabla 108
5.5. Asignación de posiciones en memoria 109
5.5.1. Asignación secuencial 109
5.5.2. Asignación encadenada 111
5.5.3. Comparación entre asignaciones secuenciales y encadenadas 111
5.6. Manejo de tablas 112
Ejercicios 115

6. MANEJO DE DATOS (II). Listas lineales

Cuadro resumen 119

6.1. Tipos de listas 121
6.1.1. Pilas 121
— Elementos en las pilas 121
— Asignación de memoria 122
— Operaciones en las pilas 124
6.1.2. Colas 128
— Elementos de las colas 128
— Asignación de memoria 128
— Operaciones en las colas 128
6.1.3. Dobles colas 130
6.1.4. Listas circulares 131
6.2. Conjuntos N-dimensionales 131
6.2.1. Asignación de memoria 132
— Asignación secuencial 133
— Asignación encadenada 133

Ejercicios 135

7. MANEJO DE DATOS (III). Arboles

Cuadro resumen 139
7.1. Elementos de los árboles 140
7.1.1. Subárboles 141
7.1.2. Nivel 141
7.1.3 Grado de un registro 142
7.2. Tipos de árboles 143
7.2.1. Arboles binarios 143
— Conversión de un árbol no binario a binario 145
— Numeración de los árboles binarios 145
— Ejemplos de árboles 146
— Exploración de un árbol binario 150
7.2.2. Arboles enhebrados 153

Ejercicios 155

Soluciones de los ejercicios 157

En este volumen se describen los fundamentos principales para la introducción de pleno en el campo de la Programación. Los conceptos expuestos aparecen de una forma progresiva que desemboca en la preparación del lector a un más profundo estudio de los campos específicos de la operativa informática.

El proceso de datos, instrumentos, partes, componentes, y relaciones entre todos ellos, vienen descritos de manera que su asimilación y entendimiento sea perfectamente asequible a toda persona que acompañe a la exposición con un mínimo de estudio, constancia e interés.

Al final de los capítulos se expone una serie de ejercicios que sirven para apreciar si la materia estudiada está perfectamente comprendida y el mismo lector puede juzgarlo mediante el solucionario que subsiguientemente se adjunta.

Una característica importante del libro es la subdivisión con que se presentan los temas más importantes que conduce a una exposición explícita y clara, siempre de gran ayuda para el lector.

El dibujo de la cubierta es cortesía de GENERAL DE COMPUTADORES, S.A.