Tecnología del PC

<http://www.zator.com/Hardware/H0_4.htm>

## Índice

* [**Portada**](http://www.zator.com/Hardware/index.htm)(inicio)
  + [0.1  Prólogo](http://www.zator.com/Hardware/H0_1.htm)
  + 0.2  Resumen de contenido
  + [0.3  Índice](http://www.zator.com/Hardware/H0_3.htm) (esta página)
  + [0.4  El ordenador electrónico digital](http://www.zator.com/Hardware/H0_4.htm)
  + 0.5  Conceptos básicos de electricidad y electrónica
* [**1 El Hardware**](http://www.zator.com/Hardware/H1.htm)
  + 1.1  Problemas. Mantenimiento y Averías
  + 1.2  La Fuente de Alimentación
* [**2 La Placa-base**](http://www.zator.com/Hardware/H2.htm)
  + [2.1  Entradas/Salidas](http://www.zator.com/Hardware/H2_1.htm)
  + [2.2  Buses locales](http://www.zator.com/Hardware/H2_2.htm)
  + [2.3  Acceso directo a memoria](http://www.zator.com/Hardware/H2_3.htm)
  + [2.4  Interrupciones](http://www.zator.com/Hardware/H2_4.htm)
    - [2.4.1 Servicios BIOS](http://www.zator.com/Hardware/H2_4_1.htm)
    - [2.4.2 Servicios del Sistema](http://www.zator.com/Hardware/H2_4_2.htm)
  + [2.5  Puertos E/S](http://www.zator.com/Hardware/H2_5.htm)
    - [2.5.1  Puertos serie](http://www.zator.com/Hardware/H2_5_1.htm)
    - [2.5.1.1  La UART](http://www.zator.com/Hardware/H2_5_1_1.htm)
    - [2.5.2  Puertos paralelo](http://www.zator.com/Hardware/H2_5_2.htm)
    - [2.5.3  Puertos USB](http://www.zator.com/Hardware/H2_5_3.htm)
  + [2.6   Factor de forma](http://www.zator.com/Hardware/H2_6.htm)
* [**3 El procesador**](http://www.zator.com/Hardware/H3.htm)
  + [3.1  Aspectos tecnológicos](http://www.zator.com/Hardware/H3_1.htm)
    - [3.2  Arquitectura del procesador](http://www.zator.com/Hardware/H3_2.htm)
    - [3.2.1  Conexiones exteriores](http://www.zator.com/Hardware/H3_2_1.htm)
* [**4 El BIOS**](http://www.zator.com/Hardware/H4.htm)
  + [4.1  Carga inicial](http://www.zator.com/Hardware/H4_1.htm)
  + [4.2  Comprobación del Hardware](http://www.zator.com/Hardware/H4_2.htm)
  + [4.3  Carga del Sistema Operativo](http://www.zator.com/Hardware/H4_3.htm)
  + [4.4  Soporte de dispositivos](http://www.zator.com/Hardware/H4_4.htm)
* [**5 Memoria interna**](http://www.zator.com/Hardware/H5.htm)
  + [5.1  Direccionar la memoria](http://www.zator.com/Hardware/H5_1.htm)
  + [5.2  Memoria Cache](http://www.zator.com/Hardware/H5_2.htm)
  + [5.3  Aspectos tecnológicos](http://www.zator.com/Hardware/H5_3.htm)
    - [5.3.1  Errores de memoria](http://www.zator.com/Hardware/H5_3_1.htm)
    - [5.3.2  Tecnologías de memoria](http://www.zator.com/Hardware/H5_3_2.htm)
    - [5.3.3  Montaje físico](http://www.zator.com/Hardware/H5_3_3.htm)
* [**6  Interfaces**](http://www.zator.com/Hardware/H6.htm)
  + [6.1  Interfaz ATA/IDE](http://www.zator.com/Hardware/H6_1.htm)
  + [6.2  Direccionamiento CHS / LBA](http://www.zator.com/Hardware/H6_2.htm)
  + [6.3  Interfaz SCSI](http://www.zator.com/Hardware/H6_3.htm)
    - [6.3.1  Aspectos físicos](http://www.zator.com/Hardware/H6_3_1.htm) (cables, conectores y "Jumpers" de configuración)
  + [6.4  Interfaz PCI](http://www.zator.com/Hardware/H6_4.htm)
  + [6.5  Interfaz Serial-ATA](http://www.zator.com/Hardware/H6_5.htm)
  + [6.6  Interfaz ESDI](http://www.zator.com/Hardware/H6_6.htm)
* [**7  Dispositivos de entrada**](http://www.zator.com/Hardware/H7.htm)
  + [7.1  Teclado.  Aspectos morfológicos y eléctricos](http://www.zator.com/Hardware/H7_1.htm)
    - [7.1.1 Funcionamiento del teclado](http://www.zator.com/Hardware/H7_1_1.htm)
    - [7.1.2 El teclado y el SO](http://www.zator.com/Hardware/H7_1_2.htm)
    - [7.1.2a  El teclado en MS-DOS / Windows](http://www.zator.com/Hardware/H7_1_2a.htm)
    - [7.1.2b  El teclado en Linux](http://www.zator.com/Hardware/H7_1_2b.htm)
  + [7.2  Dispositivos señaladores](http://www.zator.com/Hardware/H7_2.htm)
* [**8  Almacenamiento externo**](http://www.zator.com/Hardware/H8.htm)
  + 8.1  Discos duros
    - 8.1.1  Generalidades
    - [8.1.2  Estructura lógica de los discos](http://www.zator.com/Hardware/H8_1_2.htm) (Generalidades)
      * [8.1.2a  Sistemas de ficheros](http://www.zator.com/Hardware/H8_1_2a.htm)
        + [8.1.2a1  Sistemas FAT & NTFS](http://www.zator.com/Hardware/H8_1_2a1.htm)
        + [8.1.2a2  Sistemas UNIX/Linux](http://www.zator.com/Hardware/H8_1_2a2.htm)
      * [8.1.2c  El principio del disco](http://www.zator.com/Hardware/H8_1_2c.htm) (Sector de arranque)
        + [8.1.2c1  Particiones](http://www.zator.com/Hardware/H8_1_2c1.htm) (volúmenes lógicos)
        + [8.1.2c2  Sector de carga](http://www.zator.com/Hardware/H8_1_2c2.htm) ("Boot sector")
        + [8.1.2c3  Particionador](http://www.zator.com/Hardware/H8_1_2c3.htm) (**fidisk**)
        + [8.1.2c4  Instalar el sistema de ficheros](http://www.zator.com/Hardware/H8_1_2c4.htm)
      * [8.1.2d  Estructura jerárquica de un volumen](http://www.zator.com/Hardware/H8_1_2d.htm) (Directorios y Ficheros)
  + [8.2  Discos externos](http://www.zator.com/Hardware/H8_2.htm)
    - [8.2.1  Unidades para transporte de datos](http://www.zator.com/Hardware/H8_2_1.htm)
      * [8.2.1a  Disco de estado sólido](http://www.zator.com/Hardware/H8_2_1a.htm)
      * 8.2.1b  Discos duros (spin disks)
* [**9  Sistemas de video**](http://www.zator.com/Hardware/H9.htm)
  + [9.1  Física del color](http://www.zator.com/Hardware/H9_1.htm)
    - [9.1.1  Medir la luz y el color](http://www.zator.com/Hardware/H9_1_1.htm)
  + [9.2  Sistemas cromáticos](http://www.zator.com/Hardware/H9_2.htm)
  + [9.3  Sistemas de representación](http://www.zator.com/Hardware/H9_3.htm)
  + [9.4  Representación numérica del color](http://www.zator.com/Hardware/H9_4.htm)
  + [9.5  Factor Gamma y ajustes del monitor](http://www.zator.com/Hardware/H9_5.htm)
  + 9.6  Controladoras
  + 9.7  Monitores
* [**10  Sistema de sonido**](http://www.zator.com/Hardware/H10.htm)
  + [10.1  Física de sonido](http://www.zator.com/Hardware/H10_1.htm)
  + [10.2  Tratamiento digital del sonido](http://www.zator.com/Hardware/H10_2.htm)
  + [10.3  Codecs de audio/video](http://www.zator.com/Hardware/H10_3.htm)
  + [10.4  Interfaz MIDI](http://www.zator.com/Hardware/H10_4.htm)
* **11  Sistemas de comunicación**
  + [11.1  Comunicación por módem](http://www.zator.com/Hardware/H11_1.htm)
    - [11.1a  Velocidad de conexión](http://www.zator.com/Hardware/H11_1a.htm)
    - [11.1b  Estándares de comunicación](http://www.zator.com/Hardware/H11_1b.htm)
    - 11.1c
* **12 Redes**
  + 12.1  Generalidades
    - [12.2  Introducción, el modelo OSI](http://www.zator.com/Hardware/H12_2.htm)
    - 12.2.1  Redes TCP/IP (Internet)
  + [**12.4  Ethernet**](http://www.zator.com/Hardware/H12_4.htm)
    - [12.4.1  Estándares Ethernet](http://www.zator.com/Hardware/H12_4_1.htm)
    - [12.4.2  Instalación de par trenzado](http://www.zator.com/Hardware/H12_4_2.htm)
    - [12.4.2a  Medida de los conductores](http://www.zator.com/Hardware/H12_4_2a.htm)
    - [12.4.3  Instalación de una pequeña LAN en Windows](http://www.zator.com/Hardware/H12_4_3.htm)
    - 12.4.4  Problemas de conexión en Windows
  + **12.5  Tecnologías inalámbricas**
    - 12.5.0  Generalidades, el espectro radioeléctrico.
    - 12.5.1  IRDA
    - 12.5.2  Bluethooth
    - 12.5.3  Wireless LAN (IEE 802.11)

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H0_3.htm#TOP)  Inicio](http://www.zator.com/Hardware/H0_3.htm#TOP)

**El Ordenador Electrónico Digital**

**§1  Presentación:**

Imaginamos sobradamente conocidos por cualquier estudiante los principios generales sobre los que se basan los ordenadores electrónicos digitales, las máquinas en las que probablemente se ejecutarán sus programas. Sin embargo, no está de más un brevísimo repaso a fin de refrescar y traer al tiempo presente algunos conceptos.

**Nota**: En muchos países hispano-hablantes de cultura técnica más influenciada por USA, se utiliza el término "computadora" por ordenador; más parecida a su equivalente inglés "Computer". Por su parte, la versión castellana deriva del francés "Ordinateur".

**§2 Ordenador electrónico digital**

Cualquier dispositivo capaz de resolver problemas mediante proceso de la información en forma discreta. Recordemos que aquí el término "digital" se usa en contraposición a "analógico",  lo que significa que la información se representa internamente de forma numérica (números racionales). Mediante el recuento, comparación y manipulación de estos números de acuerdo con un conjunto de instrucciones almacenadas en su memoria, un ordenador digital puede realizar multitud de tareas: Realizar complejos cálculos matemáticos; reproducir una melodía, o ganar una partida de ajedrez al campeón mundial humano de la especialidad.

La información se presenta en forma de **datos** e **instrucciones** (algoritmos) para su manipulación, y como hemos señalado, está representada internamente de forma "digitalizada", es decir, reducida a cantidades discretas representadas por números, y estos a su vez expresados en **formato binario**. Este formato utiliza magnitudes que solo pueden adoptar dos valores mutuamente exclusivos (cierto y falso; cero o uno; encendido o apagado; etc). Por consiguiente, pueden representarse con solo dos dígitos O y 1. En contraposición el formato decimal tiene diez dígitos (del 0 al 9 en el sistema arábigo). Lo anterior es válido no solo para la información textual (representada numéricamente mediante el código ASCII, Unicode, o cualquier otra convención); también las imágenes, los colores, los sonidos y los algoritmos, están almacenados en el ordenador en formato binario.

**Nota**: No confundir los conceptos: **electrónico** (basados en la electrónica); **digital** (utilizar magnitudes discretas representadas por números) y **binarios** (utilizar este modo de representación, una lógica y un álgebra de Boole). Nada impide la existencia de ordenadores no electrónicos (por ejemplo ópticos, biológicos o basados en fluidos); también han existido y existen ordenadores analógicos (no digitales); y tampoco existe inconveniente teórico para la existencia de ordenadores no binarios.

El hecho de que los actuales ordenadores electrónicos digitales sean binarios, se debe, entre otras razones, a que los circuitos digitales actuales están especialmente adaptados a la utilización de la lógica binaria (volveremos sobre esto más adelante), pero han existido intentos de construirlos en base a otras lógicas, por ejemplo "ternarias". Actualmente se empieza a hablar de la computación cuántica, basada en algo distinto a dos estados mutuamente exclusivos.

Resulta así, que el universo de las actuales computadora electrónica digitales es un universo de números binarios que representan información en forma de datos e instrucciones.

Aunque en último extremo internamente todo son números, a los informáticos les gusta referirse a los primeros (los datos), clasificándolos en tres grandes apartados que se corresponden mejor con la imagen que el humano tiene de ellos: **numéricos**;  **textuales** (o alfanuméricos) y **binarios**.  Cuando estos datos no están en la memoria interna sino en dispositivos externos (generalmente discos magnéticos u ópticos), se agrupan en ficheros, de los que existen infinidad de tipos, aunque es frecuente agruparlos en tres grandes categorías: Textuales; ejecutables y binarios.

**§3 Datos numéricos**

Se refieren evidentemente a las magnitudes numéricas clásicas de las matemáticas. Aunque han existido otros sistemas (por ejemplo el sistema de numeración Romano), nuestra cultura "Occidental" actual, utiliza para su representación exclusivamente el denominado **sistema de numeración decimal con cero [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H0_4.htm#Sistema decimal)**, inventado por la antigua cultura Indostánica; sin embargo, la notación que empleamos es arábiga [[5](http://www.zator.com/Hardware/H0_4.htm#[5])]. La razón de que haya prevalecido frente al sistema latino es simplemente que es mucho más adecuado para las operaciones aritméticas.

En cuanto al sistema de codificación empleado para representarlos en los ordenadores, ya hemos dicho que es el **sistema de numeración binario [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H0_4.htm#Sistema binario)**. Su álgebra es la de Boole [[1](http://www.zator.com/Hardware/H0_4.htm#[1])], y por lo demás, el concepto es universal y no tiene idiomas o matices (suerte de los matemáticos). La notación empleada para representar gráficamente estos dígitos binarios suele ser de Ceros y Unos.

La necesidad de representar cantidades positivas y negativas ha motivado la aparición de algunas variedades de representación (binarias desde luego) a las que nos referiremos a en el capítulo dedicado a la Representación interna y rango de las variables numéricas (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.4a](http://www.zator.com/Cpp/E2_2_4a.htm)) del manual C++; su conocimiento es importante, ya que nos permitirá conocer como están almacenados internamente los números y comprender el significado de algunas las operaciones de C++.

Contra lo que ocurre con los datos textuales (a los que nos referimos inmediatamente http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif), existen diversas formas de representación simbólica (en el texto de los programas) de las cantidades numéricas, de forma que no siempre se representan en formato decimal como sería de esperar. En la sección Formas de representación simbólica (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.4b](http://www.zator.com/Cpp/E2_2_4b.htm)) se exponen detalles de los formatos utilizados.

**§4 Datos textuales**

Se refiere a la idea que todos tenemos de aquella información que puede ser representada mediante la escritura. Este tipo de datos tiene tres atributos o características que conviene matizar:

* **Lengua** natural [[2](http://www.zator.com/Hardware/H0_4.htm#[2])] en que se haya representado, que puede ser cualquiera de las conocidas: Español, Francés, Inglés, Alemán, etc.
* **Sistema de escritura** utilizado, que para nosotros es el denominado "Sistema occidental" (que se basa en un alfabeto de raíz latina), aunque naturalmente existen otros: Árabe, Chino, Japonés, Griego, etc.
* **Sistema de codificación** empleado para traducir los grafos del alfabeto a números. Se empezó utilizando el Us-ASCII, pero actualmente el sistema más utilizado es Unicode (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.1a](http://www.zator.com/Cpp/E2_2_1a.htm)).

**§5 Datos binarios**

La informática trata cada vez más con los tipos de datos denominados "Multimedia", **sonidos** e **imágenes** [[3](http://www.zator.com/Hardware/H0_4.htm#[3])], en multitud de formas y calidades: Monoaural, estereo, Dolbi-digital, MP3, etc. para los primeros; estáticas o en forma de flujo (stream); en color, o en blanco y negro, etc. para las segundas.

Aunque tanto los sonidos como las imágenes son de naturaleza intrínsecamente analógica, los informáticos han inventado sistemas para digitalizarlas (reducirlas a números) con la menor pérdida de calidad y de la forma más comprimida posible. Aquí se habla de "codecs" para referirse a los diversos algoritmos que a la postre reducen los sonidos o imágenes a ficheros de números y posteriormente reconstruyen los sonidos e imágenes a partir de los números. Estos ficheros son denominados de forma genérica como "**binarios**".

**§6 Los sistemas de representación numérica**

Una vez establecido que el universo interno del ordenador electrónico digital es numérico, resulta interesante resaltar que las características físicas de los dispositivos eléctricos, hacen que sea muy fácil representar con ellos magnitudes binarias; simplemente haciendo corresponder los dos posibles valores de la variable con dos estados físicos de un circuito o dispositivo. Por ejemplo, con los estados de: Conducción no conducción; tensión, sin tensión; magnetizado no magnetizado; con luz, sin luz; etc.

Como el álgebra de Boole maneja variables que solo pueden tener dos estados ("cierto" y "falso"), es muy fácil representar y emular comportamientos y ecuaciones de dicha álgebra mediante circuitos electrónicos. Así, construir un circuito electrónico que responda (que proporcione como salida) el OR, AND, NOR, XNOR, etc de unas ciertas variables de "entrada", es trivial desde el punto de vista de diseño de circuitos lógicos (puede diseñarse fácilmente un circuito digital que responda a unas "Ecuaciones" lógicas -de Boole- por complejas que sean).

**§6.1** **Sistema decimal**

En cuanto al problema de representar internamente cantidades numéricas, el sistema tradicional (el decimal con cero al que estamos acostumbrados), no es muy adecuado para los dispositivos electrónicos, puesto que aquí, al ser un sistema de base 10,  las variables (las cifras), pueden tener diez valores distintos (los guarismos 0 al 9 si utilizamos el sistema arábigo de numeración).

Recordemos que en este sistema, una cantidad cualquiera, por ejemplo 1798, en realidad se lee como:

1 x 103 + 7 x 102 + 9 x 101 + 8 x 100   ==  1 x 1000 + 7 x 100 + 9 x 10 + 8 x 1

http://www.zator.com/Hardware/images/Bombilla_.gif  Observe la diferencia entre los diversos conceptos involucrados: el número o cantidad "mil setecientos noventa y ocho", su representación en el sistema decimal (**1978**) y las cifras o guarismos que componen su representación en este "sistema de numeración" (en este caso cuatro cifras: **1**, **7**, **8** y **9** colocadas en un cierto orden).

Desde la escuela primaria sabemos que en este **sistema**, el valor de las cifras viene complementado por su posición en el conjunto (decimos que es "posicional"); de forma que el valor total de una expresión viene representado por el producto de su valor-base (0 a 9) multiplicado por la potencia de 10 que corresponda según su posición. Al final se suman los resultados parciales.

Resulta así que en el **sistema decimal**, la cantidad mas alta que se puede representar mediante una cantidad de cuatro cifras, **nnnn** es, como máximo:

9 x 103 + 9 x 102 + 9 x 101 + 9 x 100   ==  9999   (10.000 si le añadimos el cero).

Es fácil verificar que un número decimal de **n** dígitos puede representar como máximo una cantidad igual a **10n** (en el caso del ejemplo, para 4 dígitos, 104 = 10000).

El mismo concepto (la misma cantidad) puede ser representada utilizando distintos sistemas de numeración, por ejemplo: **MDCCXCVIII** en números romanos (por cierto, un sistema muy poco adecuado para las matemáticas).

**§6.2** **Sistema binario**

El **sistema binario** puede representar igualmente cualquier cantidad basándose en cifras que solo pueden tener dos valores, 0 y 1 (http://www.zator.com/Hardware/images/Asterisco.gif).  El sistema es exactamente análogo al decimal, con la diferencia de las potencias; de 2 que es la nueva base (en vez de 10 como en aquel caso). Por tanto, la cantidad binaria  11100000110 se lee:

1x210 + 1x29 + 1x28 + 0x27 + 0x26 + 0x25 + 0x24 + 0x23 + 1x22 + 1x21 + 0x20

Si el lector quiere "sacar la cuenta" verá que conduce igualmente a "mil setecientos noventa y ocho".

http://www.zator.com/Hardware/images/Asterisco.gifUna magnitud que solo puede tener dos valores se denomina **bit**. (abreviatura de "Binary digit", número binario). Es la menor cantidad de información que puede concebirse, y su abreviatura es **b**.  Por ejemplo, 10 Kb son 10.000 bits. Resulta así, que un interruptor que puede estar encendido o apagado es (puede ser) un almacenamiento de 1 bit de información; basta con hacer corresponder "encendido" con uno de los valores y "apagado" con el otro.

El asunto es que estas variables "binarias" (que solo pueden adoptar dos valores distintos), son mucho mas adecuadas para ser representadas mediante dispositivos eléctricos, magnéticos y ópticos, que las "decimales", por lo que es este el "formato" elegido para la representación y almacenamiento de cantidades numéricas en los ordenadores electrónicos digitales. La capacidad de representación de este sistema es sin embargo menor que en el decimal, así, una cantidad binaria de 4 dígitos puede representar como máximo:

1 x 23 + 1 x 22 + 1 x 21 + 1 x 20  == 15  (16 si añadimos el cero)

En el caso de la numeración binaria, también es fácil verificar que un número de **n** dígitos puede representar como máximo una cantidad igual a **2n** . En el caso del ejemplo, para 4 dígitos, 24 = 16).

En la tabla adjunta se expresan las cantidades que se obtienen con la unidad seguida de cierto número **n** de ceros en los sistemas decimal y binario, lo que representa las potencias **n** de la base de numeración. Es decir: 10**n** o 2**n**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Base 10** | **Número decimal** | **Base 2** | **Número binario** | **Equivalente decimal** |
| 100 | 1 | 20 | 1 | 1 |
| 101 | 10 | 21 | 10 | 2 |
| 102 | 100 | 22 | 100 | 4 |
| 103 | 1000 | 23 | 1000 | 8 |
| 104 | 10000 | 24 | 10000 | 16 |
| 105 | 100000 | 25 | 100000 | 32 |
| 106 | 1000000 | 26 | 1000000 | 64 |
| 107 | 10000000 | 27 | 10000000 | 128 |
| 108 | 100000000 | 28 | 100000000 | 256 |
| 109 | 1000000000 | 29 | 1000000000 | 512 |
| 1010 | 10000000000 | 210 | 10000000000 | 1024 |

**§6.3 Otros sistemas de representación numérica**

Además del decimal y el binario (este último es el sistema "natural" del ordenador), en la literatura informática y en los fuentes de los programas se utilizan otras formas de numeración (sobre todo para representar valores constantes). Son los sistemas **octal** (http://www.zator.com/Hardware/images/Ico_hoja.gif [2.2.4b](http://www.zator.com/Cpp/E2_2_4b.htm#Formato octal)); **hexadecimal** (http://www.zator.com/Hardware/images/Ico_hoja.gif [2.2.4b](http://www.zator.com/Cpp/E2_2_4b.htm#Formato hexadecimal)) y **BCD** http://www.zator.com/Hardware/images/Ico_hoja.gif[H0.1w1](http://www.zator.com/Cpp/E0_1w1.htm)).

**§7** **Almacenamiento interno**

Tradicionalmente el almacenamiento interno de los ordenadores se realiza en grupos de 8 (o múltiplos de 8) cifras binarias (bits), estos conjuntos (**octetos**) son la menor cantidad de información que trata el ordenador con entidad propia, reciben el nombre de **bytes**, y suele abreviarse con **B**. Por ejemplo, 16 Kb y 16 KB. se refieren a 16000 bits y 128.000 bits respectivamente.

**Nota**: Hemos señalado que **bit** es la abreviatura de **Binary Digit**, mientras que el nombre **Byte** fue acuñado en la década de los 50 por Werner Bushholz de la IBM [[4](http://www.zator.com/Hardware/H0_4.htm#[4])]; la palabra proviene de una mutación de "bit", hecha de forma que no hubiese confusión al pronunciarlas (**bit** y **byte** suenan muy distintas en inglés).

En ocasiones, además de los octetos, se utilizan agrupaciones de 4 bits, reciben el nombre de **nibble**.  Así que 2 nibbles = 1 Byte.

Podemos usar un símil para entenderlo: aunque nuestro alfabeto tiene 26 caracteres, no solemos utilizarlos aislados. Para que tengan significado se utilizan en grupos (palabras), los ordenadores utilizan también palabras, solo que estas son siempre de la misma longitud (8, 16, 32 o 64 bits según el modelo de procesador).

Según lo anterior, un octeto (una "palabra" de 8 bits) 1 Byte, puede contener un número de hasta  28 = 256, lo que deriva en que si por ejemplo, reservamos una palabra de 8 bits para "describir" (contener) una variable, sabemos de antemano que dicha variable no va a poder adoptar nunca mas de 256 estados distintos. Esto, unido al hecho de que la forma de representación interna utiliza el sistema binario y los elementos y circuitos físicos son igualmente binarios (pueden adoptar solo dos estados), hacen que las potencias de dos: 8, 16, 32, 64 etc. son "números mágicos" en el mundo de las computadoras.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H0_4.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H0_4.htm#TOP)

[1] George Boole (1815-1864).  Matemático inglés cuyos trabajos teóricos sobre conjuntos de números binarios (que solo pueden adoptar dos valores), ayudó a establecer las bases de la moderna lógica simbólica. Su álgebra de la lógica binaria es también denominada álgebra Booleana.

[2] Nos referimos a las lenguas o lenguajes "Naturales" en contraposición a los "Artificiales" utilizados en informática. Los primeros han sido inventados espontáneamente por el hombre (a excepción del Esperanto), y son el resultado de mutaciones que se pierden en la noche de los tiempos. Los segundos han sido y están siendo inventados por los informáticos para comunicarse con los ordenadores (computadoras).

[3] Ya se está experimentando con olores; creo que incluso existe alguna patente al respecto.

[4] Por supuesto, byte puede escribirse con mayúscula y minúscula, pero es costumbre informática utilizar la mayúscula "Byte", mientras que bit, se suele escribir siempre con minúscula.

[5] Ahora que el asunto parece estar de moda, digamos que no solo debemos a la cultura Islámica las cifras numéricas, incluso muchos textos clásicos han llegado hasta nosotros a través del Islám.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H0_4.htm#TOP)  Inicio](http://www.zator.com/Hardware/H0_4.htm#TOP)

## 2 La placa-base

##### §1  Presentación

En la arquitectura PC, el núcleo del sistema está integrado en una sola placa, denominada **placa-base** ("Motherboard" o "Mainboard"). En los sistemas actuales de sobremesa, es una placa de circuito impreso multicapa de unos 600 cm2 en la que se incluyen elementos de montaje superficial (soldados), amén de zócalos y conectores para diversos elementos desmontables. Existen diversos tamaños y disposiciones (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.6](http://www.zator.com/Hardware/H2_6.htm)); el aspecto de una de estas placas se muestra en la página adjunta (http://www.zator.com/Hardware/images/Ico_hoja.gif [Figura](http://www.zator.com/Hardware/H2_w1.htm)).

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/H2-F1.gif  Representación esquemática de una placa-base. |

##### §2 Componentes

A grandes rasgos, los componentes principales de la placa base son los que se han esquematizado en la figura adjunta:

* **Bus externo** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus).
* **Procesador** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Procesador)(UCP).
* **BIOS** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#BIOS)
* **Chipset**. Juego de chips auxiliares [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#ChipSet).
* **Memoria** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#memoria).
* **Conectores** de E/S, incluyendo alimentación [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Conectores auxiliares).

##### §3 Bus

**Observación**: Lo que sigue es una descripción somera de las características funcionales y principios operativos del bus del PC. Hemos de advertir que en la literatura técnica leerá que existen varios estándares de bus: ISA, EISA, VESA, MCA etc. En realidad esto se refiere a las especificaciones mecánicas de los conectores (zócalos) que se incluyen en la placa-base para insertar dispositivos tales como placas de video, módems internos, controladores de disco, etc. etc. De hecho es frecuente que una placa base tenga zócalos de más de un tipo (por ejemplo, ISA y PCI).

El elemento principal de la placa-base es el **bus** [[1](http://www.zator.com/Hardware/H2.htm#[1])], una especie de autopista que la recorre y que actúa como espina dorsal del ordenador. La mayoría de los datos pasan por esta vía para ir de un sitio a otro (el ordenador se parece a uno de esos pueblos del salvaje Oeste -de las películas- que solo tienen una calle).

**Nota**: Como muchas otras cosas de esta tecnología tan cambiante, el símil propuesto es de una exactitud relativa. La tendencia apunta a la creación de otros "buses" secundarios para misiones específicas que van y vienen con las modas tecnológicas de cada momento. Ver en el siguiente epígrafe algunos comentarios sobre los denominados **Buses locales**  (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.2](http://www.zator.com/Hardware/H2_2.htm)).

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/pcBus.jpg |

Esta autopista pasa por los zócalos y conectores de las tarjetas de expansión y periféricos, y es frecuente que en algunos equipos portátiles el bus tenga una conexión externa (en la figura anterior se ha señalado como "Conexión exterior"). Las denominadas "Dock stations" se acoplan a dichos conectores formando una prolongación del bus, lo que permite que puedan añadirse dispositivos externos.

La inclusión del bus obedece a dos razones: Por un lado es un elemento necesario para transportar datos entre las partes del ordenador. De otro, es el resultado de un criterio de diseño de los creadores del PC.  La adopción de una arquitectura abierta que permitiese conectar la mayor cantidad de dispositivos, lo que exigía que todas las señales estuviesen presentes en los zócalos de conexión y que sus características estuviesen debidamente documentadas. Ambas premisas se cumplieron y el resultado ha sido uno de los mayores éxitos comerciales de todos los tiempos.

El bus es un conjunto de pistas conductoras grabadas en la placa-base.  Precisamente su característica más importante es el número de tales pistas, pues cuanto mayor sea este número, mayor es la cantidad de bits de la señal que transporta. Como cada conductor puede transportar un bit, en realidad no se suele hablar de "conductores" para referirse a la anchura del bus, sino de "bits" (más concretamente bits de datos).

**Nota**: Cada línea transporta un bit de valor 0 o 1. Un valor "alto" (5 V.) se considera un 1; un valor "bajo" (0 V. = masa) se considera un 0. Determinados sucesos ocurren cuando en una línea se recibe un 1, se denominan "activo alto". Por contra, otros sucesos ocurren cuando en ciertas líneas se recibe un 0, se dice entonces que dicha línea es de "activo bajo".

En los primeros PCs, el bus de datos era de 8 bits (a pesar que el procesador utilizado, el 8088 era de 16 bits), por lo que se podía transferir 1 octeto cada vez, y las direcciones debían ser manejadas en dos partes (esto dio origen a una forma de direccionamiento denominada "Segmentada", a la que nos referiremos a continuación [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de direcciones)). El "ancho" del bus de datos se aumentó posteriormente a 16 bits; en las máquinas actuales es de 32 y la tendencia es ir a 64.

http://www.zator.com/Hardware/images/Bombilla_.gifNo confundir este bus de la placa-base, denominado también **bus externo** [[4](http://www.zator.com/Hardware/H2.htm#[4])] o **FSB** ("Frontside bus"), con el existente en el interior del procesador para comunicar entre sí sus diversos módulos (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3.2](http://www.zator.com/Hardware/H3_2.htm#El bus interno)).  Este último se conoce como **bus interno** o **bus del sistema** ("System bus"), y ni su "anchura" ni su velocidad (frecuencia) tienen porqué coincidir con las del externo. Hemos visto como en el 8086 es de 16 bits, mientras que el bus externo del PC XT que utilizaba dicho procesador, es solo de 8 bits. En los sistemas actuales (2001) una velocidad típica del procesador es del orden de 400/500 MHz, mientras que el FSB suele trabajar a una fracción de aquella (típicamente 100 MHz), además la anchura del bus del procesador suele ser de dos a cuatro veces la del bus externo.

**Nota**: Este divorcio entre la velocidad del procesador y la que puede aceptar la placa-base (bastante inferior a la del primero), comenzó con la introducción del procesador Intel486, lo que a su vez condujo a que la caché L1 cobrara especial relevancia (http://www.zator.com/Hardware/images/Ico_hoja.gif [H5.2](http://www.zator.com/Hardware/H5_2.htm#Tipos de cache)).

Además de los conductores de datos, existen otros auxiliares, de modo que en esta "autopista" tiene en realidad más carriles. En el PC original, a pesar de ser "solo" de 8 bits, existían en realidad 62 conductores (31 en la cara superior "de componentes" -A- y 31 en la inferior "de soldaduras" -B-). En el argot de los ingenieros de hardware estas líneas se conocen con acrónimos y abreviaturas universalmente admitidas, de las que indicaremos algunas.

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/ISA8bits.gif  Esquema de contactos **ISA 8 bits** |

Los zócalos para tarjetas de expansión del PC XT eran del modelo denominado **ISA** ("Industry Standard Architecture") corta o ISA de 8 bits, en el que se podían insertar tarjetas con 62 contactos (31 por cada cara). Posteriormente, en 1984, los modelos AT añadieron 36 nuevas líneas al bus, utilizando el zócalo ISA de 16 bits, que admitía tarjetas con 98 contactos (49 por cara), pero manteniendo un diseño de zócalos que los hacía eléctricamente compatibles con las tarjetas ISA cortas. De estas nuevas líneas, 8 eran de datos (el bus pasaba a ser de 16 bits), cinco nuevas líneas de control de interrupciones, y siete de dirección, puesto que el nuevo ancho de bus permitía direccionar más memoria.

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/ISA16bits.gif  Esquema de contactos (medidas en pulgadas) **ISA 16 bits** |

Además de la compatibilidad mecánica, se había incluido un ingenioso dispositivo que detectaba si la tarjeta insertada era larga o corta para adecuar el modo de direccionamiento al estilo adecuado.  El resultado era que los equipos ISA de 16 bits podían aprovechar las antiguas tarjetas ISA, aunque con un cierto desaprovechamiento de las capacidades de los sistemas AT, una de las cuales era precisamente la capacidad de utilizar 16 bits para direcciones y datos.

En 1988 Compaq y otros fabricantes mejoran el ISA de 16 bits, adoptando el que se denomina bus **EISA** (ISA Extendido), que guardaba compatibilidad hacia atrás con los buses ISA anteriores, y que ha persistido largamente.

En 1993 Intel introdujo el bus **PCI** ("Peripheral Component Interconnect") que proporciona 32 bits. Su versión 2.1 soporta 64 bits y velocidades más rápidas en la frecuencia de reloj del bus.  En la actualidad (2001) **PCI** es el nuevo estándar de facto, y ha sustituido casi completamente a EISA, de forma que empiezan a verse tarjetas que no tienen ningún conector para las antiguas tarjetas EISA.

##### §3.1 Principios de funcionamiento

Para comprender el funcionamiento del bus, y en general del mecanismo de transmisión de la información dentro del ordenador, es preciso hacer tres observaciones previas:

**a:**   En realidad, el **bus externo** puede considerarse dividido en cuatro vías para otros tantos servicios distintos:

* Bus de datos [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de datos)
* Bus de alimentación [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de alimentación).
* Bus de control [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de control).
* Bus de Direcciones [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de direcciones).

**b:**    Señalar que el bus externo responde al modelo que en tecnología de redes se conoce como "Broadcast" o difusión amplia [[5](http://www.zator.com/Hardware/H2.htm#[5])].  En este modo de transmisión, cuando se quiere transmitir alguna información, el dato es puesto en el **bus de datos** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de datos), y la dirección de destino en el **bus de direcciones** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de direcciones). Ambas señales llegan a todos los dispositivos conectados al bus, pero el dato solo es aceptado por aquel cuya dirección coincide con la señalada por el bus (en realidad un dispositivo puede aceptar una sola dirección o un rango de ellas).

**c:**   Todos los sistemas de redes (el bus es una red) tienen un sistema de "direcciones" para saber quién, de entre todos los elementos conectados, debe recibir la información (el mensaje recibido puede a su vez ser la orden de enviar una respuesta). En nuestro caso, los intercambios pueden ser entre cualquier par de dispositivos. Por ejemplo, un almacenamiento en disco y una impresora, pero los que involucran memoria RAM son especialmente importantes, ya que antes o después todos los datos pasan por allí (incluso los datos e instrucciones de los programas en ejecución están en RAM). Esto indujo a los diseñadores del PC a disponer que las transferencias de datos entre la memoria y ciertos dispositivos recibieran un tratamiento especial. Hasta el punto que dentro del bus existe un subsistema específico para controlar estas transferencias, el denominado **acceso directo a memoria** (DMA [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Controlador DMA)).  Lo de "directo" viene a cuento de que en los demás casos, la transferencia de datos entre dos dispositivos A y B requiere la intervención del procesador, mientras que las transferencias DMA lo mantienen al margen.

El resultado es que la comunicación entre la memoria y ciertos dispositivos, o entre esta y el procesador, se efectúa de forma un tanto especial. Mientras que las comunicaciones entre dispositivos, o entre dispositivo y procesador, se realizan mediante un sistema distinto, denominado **puerto de E/S** (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.1](http://www.zator.com/Hardware/H2_1.htm#Direccionar las E/S)).

**d:**   El mecanismo **DMA** es posible porque el PC permite que el microprocesador, que en último extremo gobierna el sistema, ceda el control del bus ("Bus mastering") a otro dispositivo que asume su control durante determinados periodos. La finalidad es suprimir la intervención del procesador cuando no es estrictamente necesario, lo que simplifica y agiliza el proceso.

###### Resumen

Para que pueda conocerse el destino de los datos en esta red de elementos intercomunicados que constituyen el bus, cada dispositivo tiene una dirección. Cualquiera que sea su origen, para enviar un dato a un **puerto** (dispositivo) o a una dirección de **memoria**, la UCP coloca la dirección de destino en el bus de direcciones, a continuación el remitente sitúa el dato en el bus de datos. Cuando esto sucede, el destinatario se da por aludido y lee el dato.

http://www.zator.com/Hardware/images/Bombilla_.gifEs de vital importancia recordar que, para distinguir el tipo de transmisión a realizar por el bus de datos, existe una línea específica en el bus de control, que indica si la dirección colocada en el bus de direcciones se refiere a un "puerto" o a memoria.

##### §3.2 Bus de datos

Junto con el tipo de procesador soportado, la "anchura" del bus de datos es quizás una de las características distintivas más importantes de una placa-base, pues determina la cantidad de información que puede fluir por ella en cada ciclo de reloj.

Las 8 líneas de datos se identifican **D0** a **D7**. Con la introducción del PC AT se añadieron 8 nuevas líneas (**D8** a **D15**), hasta un total de 16. Al tratar de la arquitectura del procesador (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3b1](http://www.zator.com/Hardware/H3_2_1.htm)), veremos que en los primeros PC's algunas de estas líneas eran de doble uso.

**Nota**: Aparentemente la decisión inicial de adoptar un bus de datos de 8 bits a pesar que el procesador 8088 utilizado era de 16 bits, se debió exclusivamente a razones económicas. En la época, la circuitería auxiliar de 8 bits era más abundante y económica que la de 16 bits.

##### §3.3 Bus de alimentación

En los primeros PCs, la fuente de alimentación alimenta a la placa-base a través de dos conectores Molex® de 6 contactos cada uno. En estos cables están las tensiones de alimentación (+12V.; -12V.; +5V. y -5V.); conexiones a masa, y la señal de Power Good, que está relacionada con la señal de "reset" [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Bus de control). A partir de este punto, el bus cuenta con una serie de líneas que trasladan estas tensiones a los dispositivos instalados en la propia placa-base y al resto de tarjetas instaladas en los zócalos.

El bus de alimentación cuenta con líneas para las tensiones +5, -5, +12 y -12 V (la de +5 V está duplicada). Además existen tres líneas conectadas a masa **GND** (para que todos los elementos tengan una referencia de potencial común).

**Nota**: En el IBM PC, la fuente proporcionaba 65 W. Las fuentes actuales suministran de 200 a 250 W de potencia para los sistemas de sobremesa. Además los conectores se han modificado (actualmente los cables que van a la placa-base utilizan un solo conector de 20 pines http://www.zator.com/Hardware/images/Ico_hoja.gif[Conector ATX](http://www.zator.com/Hardware/H2_w2.htm)) y se han añadido nuevas señales, de forma que la fuente puede desconectarse mediante una señal enviada desde la placa base. Esta señal puede ser enviada por el Sistema -desconexión software- o por el usuario desde el teclado.

##### §3.4 Bus de control

Además de los datos y la energía, el bus transporta una serie de señales de control, por lo que a estas líneas se las denomina **bus de control** o del sistema. Las principales son las siguientes:

* **0WS** ("Zero Wait State"). Es utilizado por los dispositivos rápidos conectados al bus para prevenir al procesador que no inserte estados adicionales de espera.
* **AEN** ("Address Enabled") Se utiliza para indicar si es la UCP o el controlador DMA, el que tiene control sobre las líneas de datos y de direcciones en ese momento. Cuando está alta, el controlador DMA controla el bus de datos y el de direcciones, además de las líneas MEMR, MEMW, IOR e IOW,
* **ALE** ("Address Latch Enabled") Cuando está alta, señala que la UCP ha colocado una dirección válida en el bus de direcciones.
* **CLK**. Señal del reloj que conecta directamente con una patilla del procesador (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3b1](http://www.zator.com/Hardware/H3_2_1.htm)). En el XT es de 4.77 MHz, un tercio de la señal del oscilador (OSC). En el AT esta frecuencia se elevó a 7 MHz, la mitad de la señal del oscilador.
* **OSC**. Señal del oscilador http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif;  basada en la frecuencia de un cristal instalado en la placa-base. En el XT es de 14.31818 MHz.
* **I/O CH CHK** ("Channel Check"). Es una señal de detección de errores. Cuando está baja, algún dispositivo del bus ha detectado un error de paridad.
* **I/O CH RDY** ("Channel Ready"). Sirve para avisar al procesador o al MDAC [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Controlador DMA)que un dispositivo lento necesita tiempo extra para estar preparado. Cuando algún dispositivo necesita tiempo extra, baja esta señal, aunque nunca durante un periodo de más de 10 ciclos CLK.
* **IRQ** ("Interrupt request"). En el PC XT existen 6 líneas identificadas **IRQ2** a **IRQ7**, que juegan un papel fundamental en el funcionamiento del ordenador, ya que la tecnología de los ordenadores modernos se basa en un sistema de interrupciones al que nos referiremos más adelante (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.4](http://www.zator.com/Hardware/H2_4.htm)). Estas 6 líneas fueron aumentadas posteriormente en el PC AT, añadiéndose 8, aunque no todas fueron destinadas al servicio de periféricos, por lo que en el nuevo BUS solo aparecen **IRQ10**, **IRQ11**, **IRQ12**, **IRQ14** e **IRQ15**.   
    
  Adelantamos aquí que una interrupción es una señal que se origina en un dispositivo (un periférico) para indicar al procesador que algo requiere su atención inmediata (de ahí su nombre, literalmente "Petición de interrupción"). Se solicita al procesador que suspenda lo que está haciendo para atender la petición.
* **IOR** ("I/O Read") Orden de lectura. Esta línea indica a un dispositivo E/S conectado al bus que debe colocar un dato en el bus de datos.
* **IOW** ("I/O Write") Orden de escritura. Esta línea indica a un dispositivo E/S que debe leer el dato situado en el bus de datos.
* **RESET**. Una señal en esta línea (http://www.zator.com/Hardware/images/Ico_hoja.gif [H4](http://www.zator.com/Hardware/H4.htm#Ref-1)), que conecta con una patilla especial del procesador, origina el reinicio del sistema. Cuando esta línea es puesta a masa (desaparece su tensión) y después vuelve a su valor normal, se origina un reinicio del procesador.
* **SBHE** ("System Bus High Enable"). Cuando está activado indica que se está realizando una transferencia de datos.
* **DRQ** ("DMA Request"). Similares en concepto a las líneas IRQ, pero utilizadas para solicitar acceso directo a memoria. Existen tres de ellas, señaladas **DRQ1** a **DRQ3**.
* **DACK** ("DMA Acknowledge"). Se utilizan para acusar recibo de la petición DRQ correspondiente. Existen cuatro, señaladas **DACK0** a **DACK3**, aunque el mecanismo DMA solo utiliza las tres últimas, DACK0 es utilizada en el XT para señalar un ciclo de refresco de memoria.
* **MEMR** ("Memory Read") Cuando se activa, esta señal indica a la memoria conectada al bus que escriba los datos en el bus de datos.
* **MEMW** ("Memory Write") Cuando se activa, indica a la memoria que almacene los datos situados en el bus de datos.
* **T/C** ("Terminal Count"). Sirve para señalar que el controlador DMA ha alcanzado el final de una transferencia (http://www.zator.com/Hardware/images/Ico_hoja.gif [2.3](http://www.zator.com/Hardware/H2_3.htm#Funcionamiento)).

El sistema de acceso directo a memoria DMA ("Direct Memory Access") es muy importante. Como puede verse, algunas líneas del bus están destinadas a este servicio, que será tratado más adelante con mayor detalle (http://www.zator.com/Hardware/images/Ico_hoja.gif [2.3](http://www.zator.com/Hardware/H2_3.htm#Líneas de control)).

##### §3.5 Bus de direcciones

El bus del PC XT contiene 20 líneas identificadas **SA0** a **SA19**; que en el AT se aumentaron a 27, añadiendo las denominaciones **LA17** a **LA23**. Estas líneas contienen las direcciones que identifican la parte del ordenador que comunica. Las direcciones son de dos tipos; unas se refieren a direcciones de memoria; otras a direcciones de **puertos** de E/S, pero en cualquier caso son de 20 bits para el PC XT y de 27 para los ATs.

Con los datos anteriores, resulta evidente (http://www.zator.com/Hardware/images/Ico_hoja.gif [0.4](http://www.zator.com/Hardware/H0_4.htm#Almacenamiento interno)) que el PC XT podía manejar 1.048.576 direcciones (220), mientras que en el AT esta cantidad asciende a algo más de 134 millones (227). Sin embargo, recordemos que tanto los procesadores 8088 presentes en el XT, como los 80287 del AT, tenían registros de 16 bits (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3b](http://www.zator.com/Hardware/H3_2.htm#Los registros)) que solo podían albergar 65.536 posiciones (216), muy alejado de los valores anteriores. Para resolver el problema, se utilizan las **direcciones segmentadas** (http://www.zator.com/Hardware/images/Ico_hoja.gif [H5.1](http://www.zator.com/Hardware/H5_1.htm#Direccionamiento segmentado)), compuestas por dos palabras de 16 bits que se componen para formar una dirección de 20 bits.

**Nota**: durante las operaciones E/S de puertos, las direcciones SA16-SA19 del 8088 permanecen iguales a cero (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3.2.1](http://www.zator.com/Hardware/H3_2_1.htm#A19-16/S6-3)), por lo que solo pueden utilizarse 16 bits para direcciones de puertos.

##### §4 Procesador

A estas alturas del siglo XXI no creo que exista nadie que necesite explicación general sobre su función e importancia. El procesador es realmente el "Cerebro" del Sistema (cuando empezaron a verse los primeros ordenadores, la gente los llamaba "cerebros electrónicos").

En el siguiente epígrafe (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3](http://www.zator.com/Hardware/H3.htm)) exponemos un resumen de sus partes principales desde el punto de vista que interesan al informático. Aquí señalaremos solamente que, en los PCs actuales, es el chip más grande que puede encontrarse en la placa-base, denominado genéricamente "Procesador", contiene en una sola unidad lo que anteriormente se denominaba Unidad Central de Proceso **UCP** (CPU en la literatura inglesa); el coprocesador matemático **MCP** ("Math coprocessor") y una cierta cantidad de memoria auxiliar (caché) de acceso rápido.

##### §5 Chips auxiliares

**Nota**: Las alusiones en el texto a los chips utilizados se refieren a los PCs originales IBM PC; XT y AT. Aunque actualmente han sido suplantados por un par de circuitos (ver figura 1 [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Ref-0)), sin embargo, la funcionalidad permanece.

El juego de chips auxiliares ("Chipset") es un conjunto de circuitos integrados que realizan misiones específicas de apoyo. Por ejemplo organizar el tráfico en esa autopista que es el bus externo.  Generalmente están soldados de forma permanente y forman un conjunto armónico con el procesador y con las características de la placa (las placas modernas vienen bastante "desnudas", sin procesador, ni memoria, ni dispositivos periféricos, de forma que resultan altamente configurables, pero siempre incluyen este conjunto de integrados de apoyo). Los más significativos son los siguientes:

* Elementos de temporización [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Elementos de temporización).
* Controlador del bus externo [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Controlador del bus)
* Controladores de acceso directo a memoria **DMAC** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Controlador DMA)
* Controlador programable de interrupciones **PIC** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Controlador de interrupciones)
* Controladores de periféricos [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Controlador de periféricos)

##### §5.1 Elementos de temporización

Los ordenadores son máquinas síncronas, lo que significa que todas sus partes funcionan de forma acompasada.  Es frecuente poner el ejemplo de los remeros en las películas de galeras, donde cada remero no va por su cuenta, todos siguen el ritmo del gordo del tambor. En el PC ocurre otro tanto, aquí el ritmo no lo marca un tambor, sino un metrónomo (**reloj**) que genera una señal (**señal de reloj**), que marca el ritmo de la "música" que es capaz de ejecutar el sistema. Esta señal está presente en el bus de control [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Bus de control), la línea **CLK**.

El corazón del reloj es un oscilador controlado por un cristal de cuarzo que actúa de patrón (es un elemento fácilmente identificable con aspecto de almohadilla plateada). En los PC originales la frecuencia del cristal es de 14.31818 MHz. Esta señal está también presente en el bus de control, línea **OSC**, y la denominaremos **frecuencia del oscilador**.  En los primitivos PCs la señal del reloj era de 4.772720 MHz. Es decir, un tercio de la frecuencia del oscilador [[3](http://www.zator.com/Hardware/H2.htm#[3])].  A su vez, la señal del reloj es dividida por 262144 (218) para proporcionar una interrupción (tic) a una cadencia aproximada de 18.2 veces por segundo.

La señal **CLK** es importantísima, ya que sin ella sería imposible un funcionamiento armónico de los distintos elementos (incluyendo el procesador, una de cuyas patillas conecta directamente con ella). Todos los sucesos internos tienen lugar al compás de este tic-tac electrónico [[6](http://www.zator.com/Hardware/H2.htm#[6])].

**Nota**: Hemos señalado [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Ref-2), que a partir de la introducción del 80486 algunos elementos del ordenador no pueden funcionar al ritmo endiablado del reloj, que resulta demasiado rápido para el resto de chips instalados en la placa, por lo que en esta se utilizan frecuencias inferiores. Esto se consigue utilizando solo uno de cada tres, cuatro o cinco pulsos de **CLK**.   
  
En este sentido es particularmente significativa la denominada **velocidad del bus externo**.  Que es la que se utiliza en dicho bus. Por ejemplo, un procesador que esté trabajando a 200 MHz, puede que esté utilizando una frecuencia de 66 MHz en sus comunicaciones con el bus externo.  Exactamente la misma que un procesador de "solo" 66 MHz de reloj que utilice también 66 MHz para el bus externo. La situación actual es que las velocidades típicas de las placas-base son de 60- 66- 100- 133 MHz mientras que los procesadores funcionan a frecuencias múltiplos de aquellas; típicamente de 1x hasta 10x, es decir, desde 60 a 1330 MHz (lo que se denomina **velocidad del núcleo**).

##### §5.1.1 Chip del reloj

En el PC original, el oscilador antes mencionado está montado en un chip 8284A (88284 en el AT), que se conoce como **generador de reloj** (no confundir con el reloj programable [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2.htm#Temporizador programable)).  La fecha y hora, que había que introducir en la puesta en marcha del sistema, se perdía cada vez que se apagaba el equipo. Durante el funcionamiento, el sistema de fecha y hora era mantenido por una interrupción de alta prioridad (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.4](http://www.zator.com/Hardware/H2_4.htm)) que ocurría unas 18 veces cada segundo.

A partir de la introducción del PC AT, se instaló en la placa base el denominado **chip del reloj**, abreviadamente **RTC** ("Real Time Clock"). Es un auténtico reloj que suministra la fecha y hora al sistema. Como este reloj debía continuar su funcionamiento incluso estando el equipo desconectado, se le dotó de una pequeña pila o batería.

Aprovechando que este circuito disponía de su propia fuente de energía, se utilizó para añadirle cierta cantidad de memoria **CMOS** ("Complementary Metal-Oxide Semiconductor") de bajo consumo, la denominada **ROM del sistema**, que se utiliza para guardar información relativa a la configuración del equipo. Estos datos son accesibles pulsando una combinación de teclas en los momentos iniciales de arranque, lo que inicia un programa de configuración grabado en la BIOS conocido generalmente como Set-up.

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/batteryBackup.jpg  Fig. 2 |

**Nota**: La batería de backup de la placa base, es un elemento fácilmente identificable; tiene forma de pila de botón reemplazable, o de elementos soldados recubiertos de una funda plástica. Este último es el caso mostrado en la fig. 2. La batería es el elemento azul; por encima está el conector del teclado y sobre éste, junto al letrero BATTERY, se distinguen tres de los 4 pines del conector para la batería auxiliar [[7](http://www.zator.com/Hardware/H2.htm#[7])].   
  
Esta batería se agota con el tiempo, especialmente si el equipo permanece mucho tiempo apagado, en cuyo caso se pierde la fecha y (lo que es más importante) los datos de configuración establecidos con el programa de Set-up. Aunque la mayoría de equipos funcionan bien con la configuración por defecto, en caso de sustituir la batería es posible que sea necesario reconfigurar el Set-up.

Tradicionalmente este circuito **CMOS/RTC** era un Motorola MC146818.  En los modelos iniciales esta memoria era de 128 bytes. Actualmente es de 256 KB, y junto con algunos otros elementos están incluidos en un chipset de solo dos circuitos integrados. A su vez el programa de configuración es cada vez más sofisticado, de forma que simplifica la configuración del sistema. Por ejemplo, dispone de un sistema de protección ("Password") de acceso y de autodetección del tipo de disco instalado; algo que en los modelos antiguos debía hacerse manualmente.

##### §5.1.2 Temporizador programable

Junto con el anterior se encuentra un circuito que recibe varios nombres: **PIT** ("Programmable Interval Timer") temporizador programable, reloj programable, temporizador ("timer"), Etc. Constituido por un chip 8253  (8254 en el AT y máquinas con bus EISA/MCA) que dispone de tres temporizadores programables independientes numerados del 0 al 2.

En realidad son divisores de frecuencia que obtienen la señal del generador de reloj. Tienen cuatro modos de funcionamiento programables que pueden generar diversas señales:

* Pulsos de anchura variable.
* Generar una interrupción al final de la cuenta de una serie de pulsos de reloj, para lo que dispone de tres contadores independientes de 16 bits.
* Ondas cuadradas que sean una fracción de la frecuencia de entrada (reloj). Por ejemplo, podemos conseguir que generen una señal por cada 1000 señales de la frecuencia del reloj **RTC**, con lo que en un PC XT obtendríamos una frecuencia de 4.77 KHz.
* Medir el intervalo entre sucesos.

Estas capacidades se utilizan para cuestiones tales como producir señales de sincronización para refresco de la RAM, o tonos audibles de diversas frecuencias en el altavoz del PC. La distribución de tareas es como sigue:

* Canal 0: Produce una interrupción cada 54.94 ms en **IRQ0**, que activa la rutina de actualización del reloj de tiempo real **RTC** [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Chip del reloj).
* Canal 1: Produce una interrupción cada 15.12 microsegundos que marca el comienzo del ciclo de refresco de las memorias RAM dinámicas (http://www.zator.com/Hardware/images/Ico_hoja.gif [5.3.2](http://www.zator.com/Hardware/H5_3_2.htm#DRAM)).
* Canal 2: Este canal se destina a generar tonos para el altavoz del sistema.

##### §5.2 Controlador del bus

Es una especie de controlador de tráfico. En el PC XT es el chip 8288 y en el AT es el 82288. El procesador necesita del concurso de este chip para generar el juego completo de señales del bus de control.

##### §5.3 Controlador DMA

**DMA** es el acrónimo de "Direct Memory Access"; se trata de un mecanismo incluido en la arquitectura del PC, que permite intercambios de datos entre la memoria y ciertos dispositivos sin intervención del procesador (por esta razón se denomina precisamente **acceso directo a memoria**). Está controlado por un chip específico, el **DMAC** ("DMA Controller"). En los XT estaba integrado en un chip 8237A que proporcionaba 4 canales. Posteriormente en los AT se instalaron dos, ampliándose a 8 el número de canales. En el epígrafe dedicado al Acceso directo a memoria (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.3](http://www.zator.com/Hardware/H2_3.htm)), se expone con más detalle su funcionamiento.

##### §5.4 Controlador de interrupciones

Las interrupciones del PC, a las que nos referimos más detenidamente en el epígrafe correspondiente (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.4](http://www.zator.com/Hardware/H2_4.htm)), son manejadas por un chip conocido como **PIC** ("Programmable Interrupt Controller").

Originariamente era un 8259A, que puede manejar 8 peticiones de interrupción, pero pronto se hizo evidente que eran un número excesivamente corto, por lo que en 1984, IBM añadió un segundo controlador en el modelo AT. Este diseño, con dos controladores, ha permanecido invariable hasta nuestros días en lo que se refiere a su funcionalidad. Recientemente ha sido sustituido por el denominado **APIC** ("Advanced Programmable Interrupt Controller") en las máquinas Pentium y sucesoras, que está basado en un chip 82489DX.

##### §5.5 Controladores de periféricos

En la placa-base existen diversos chips destinados específicamente al control de periféricos serie y paralelo. A continuación reseñaremos brevemente algunos de los más conocidos.

El **PPI** ("Programmable peripheral interface"), también conocido como **PIO** ("Programmable Input/Output") era originariamente un 8255.  Se trata de un chip de 24 patillas que pueden ser programadas individualmente en dos grupos de 12, de las cuales 8 son de datos y 4 bits auxiliares incluyen señales de control (STROBE; ACKNOWLEDGE; BUFFER FULL, Etc.) Este chip tiene gran variedad de aplicaciones, entre ellas almacenar información sobre el estado de configuración de la placa-base y controlar otros chips, como el temporizador programable [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Temporizador programable)y la interfaz del teclado, pero su uso característico es como interfase paralelo bidireccional para periféricos que utilizan este tipo de comunicación (por ejemplo impresoras, escáneres, Etc.)  El altavoz del sistema está conectado a dos patillas de este integrado.

Desde el punto de vista lógico este dispositivo responde a las direcciones 060h-063h, que son sus números de **puerto** [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Ref-3)y dispone de tres registros de 1 byte conocidos como **A**, **B** y **C**, cuyas direcciones respectivas son 060h, 061h y 062h. La dirección 063h se utiliza para configurar el dispositivo.

**Nota**: Como hemos señalado anteriormente, a partir del AT, el estado de configuración de la placa-base se almacena en un integrado alimentado con batería que contiene también el reloj de tiempo real. En realidad el AT no tiene un 8255 como tal, pero emula su comportamiento. En este caso el controlador es un 8742 responde a las direcciones 060h-064h, y realiza otras funciones no directamente relacionadas con el teclado, como resetear la UCP, controlar la puerta A20 y manejar el ratón [[9](http://www.zator.com/Hardware/H2.htm#[9])].

El 8250 **ACE** ("Asynchronous communication equipment"), es una interfaz entre el bus paralelo del sistema y los periféricos que utilizan comunicación serie, como módems y ratones. Puede generar las señales de control de la popular interfaz RS-232 [[8](http://www.zator.com/Hardware/H2.htm#[8])]; dispone de un generador de velocidad de transmisión que permite generar cualquiera de las más frecuentes. Puede funcionar bajo control de programa o mediante interrupciones (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2a](http://www.zator.com/Hardware/H2_1.htm#Tipos de comunicación E/S)) y dispone de una cierta capacidad de comprobación y corrección de los errores típicos de las comunicaciones serie.

El PD765, que controlaba el disquete en los XT, más conocido como **FDC** ("Floppy Disc Controller"). Fue sustituido por el 82072A en el AT y por el 82077A en el PS/2.

##### §6 Memoria

El término "memoria" se refiere desde luego a cualquier dispositivo capaz de almacenar datos, pero aplicado a un ordenador actual es demasiado ambiguo, ya que existen varias clases. En principio, pueden clasificarse en dos grandes grupos: **Internas** y **externas** (también denominadas primarias y secundarias).

**§6.1** Las **memorias internas** están situadas en el procesador, la placa base, o en tarjetas insertas en zócalos de esta. En el capítulo correspondiente (http://www.zator.com/Hardware/images/Ico_hoja.gif [H5](http://www.zator.com/Hardware/H5.htm)), comentamos algunos de aspectos funcionales que son de especial interés para el programador.

**§6.2** La **memoria externa** está situada en dispositivos que se consideran "Externos" o "Periféricos" al ordenador, tales como disquetes, discos duros, cintas, CDs, DVDs, etc. Algunas de ellas son intercambiables, de forma que la capacidad de almacenamiento puede ser virtualmente ilimitada, aunque con el costo de tener que insertar manualmente la cinta, CD o DVD en la unidad correspondiente.

Los SOs utilizan la memoria externa para almacenar ficheros con información, pero los sistemas modernos, tipo Windows o Linux también utilizan la memoria externa en sustitución de la interna; un artificio conocido como **memoria virtual** (http://www.zator.com/Hardware/images/Ico_hoja.gif [H5.1](http://www.zator.com/Hardware/H5_1.htm#Memoria virtual)). La memoria virtual utiliza un fichero de intercambio ("Swap file") en memoria externa, que almacena la información que no cabe en RAM.

**Nota**: Actualmente ha surgido una nueva forma de memoria externa, que seguramente es el sistema del futuro. Nos referimos a Internet como fuente inagotable de almacenamiento de información. Ya han surgido incluso PSIs [[2](http://www.zator.com/Hardware/H2.htm#[2])] que ofrecen almacenamiento "ilimitado" a sus usuarios.

El estudio de las memorias externas, aunque muy interesante (apasionante, diría desde el punto de vista tecnológico) y de uso constante, en especial los discos duros, se sale totalmente de los márgenes que hemos impuesto a esta obra.

##### §7 BIOS

Básicamente es una memoria en la que se guardan datos fundamentales para el funcionamiento de la placa-base y sus periféricos principales. El funcionamiento de los IBM PCs y compatibles se basa en su existencia, y su importancia justifica que se le asigne un chip especialmente dedicado a contenerla.

En los equipos modernos, el chip de la **BIOS** es fácilmente identificable por una pegatina plateada con un holograma, en la que junto a la palabra BIOS, aparece el nombre del fabricante. Probablemente alguno de los siguientes: AMI; Phoenix; Award; AST; Compaq; Microid; Mylex; IBM; Quadtel; ACER, o ALR (los más conocidos). En atención a su importancia, le hemos dedicado un capítulo (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3](http://www.zator.com/Hardware/H4.htm)).

##### §8 Conectores auxiliares

La placa-base dispone generalmente de una serie de conectores para enchufar determinados dispositivos. Entre los más frecuentes están: Conector para la fuente de alimentación (http://www.zator.com/Hardware/images/Ico_hoja.gif [Conector](http://www.zator.com/Hardware/H2_w2.htm)); Conector DB25 de puerto paralelo; conector DB9 para puerto serie; conector USB; conectores mini-DIN para ratón y teclado, etc. Algunos de ellos se muestran en la figura (http://www.zator.com/Hardware/images/Ico_hoja.gif [Figura-1](http://www.zator.com/Hardware/H2_w1.htm)).

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H2.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H2.htm#TOP)

[1] Utilizamos el vocablo original inglés "Bus" porque ha tomado carta de naturaleza en nuestro idioma. Seguramente si usted dice el "autobús" del PC nadie sabrá a que se refiere.

[2] PSI: Proveedor de Servicios de Internet (ISP "Internet Service Provider").

[3] Posteriormente esta frecuencia fue elevada al 50%, es decir, 7.15 MHz en los ATs.  Aunque desde luego hoy nos parecen ridículas, muchos de los primeros usuarios sobrecargaban sus sistemas cambiando los cuarzos ("overclocking") a fin de conseguir algo más de velocidad, ya que los valores adoptados por IBM eran muy conservadores, y los equipos funcionaban sin dificultad a frecuencias algo más elevadas.

[4] A veces el **bus externo** es llamado impropiamente bus de datos ("Data bus"), cuando este es en realidad solo una parte del bus externo [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2.htm#Ref-1).

[5] El caso opuesto, cuando la información se transmite solo al dispositivo receptor se denomina "Multicast". Esta modalidad requiere que la red tenga cierta capacidad de direccionamiento o conmutación, de forma que pueda establecerse una vía exclusiva entre el transmisor y el receptor.

[6] Para que esta sincronía sea posible, se dispone incluso de un mecanismo que obliga al procesador a permanecer inactivo un número determinado de ciclos (de espera) cuando un periférico no está preparado para seguirlo (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3.2.1](http://www.zator.com/Hardware/H3_2_1.htm))

[7] En estos casos, la placa base tiene unos pequeños pines metálicos (generalmente 4 en línea) en los que puede abrocharse un pequeño conector para instalar una batería auxiliar externa que puede conseguirse en el comercio especializado; la única precaución es respetar la polaridad. El cable rojo que acompaña a la batería es el positivo; debe corresponder con el pin positivo que está claramente señalado con "+" o "1" en la placa-base.

[8] Durante muchos años esta ha sido la interfaz estándar para comunicaciones serie de la EIA ("Electronic Industries Association").

[9] Esta es la razón por la que, en los modelos PS/2, el conector mini-DIN pueda conectarse a un teclado o al ratón. Por su parte, la puerta ("Gate") A20 maneja el acceso a memoria en modo protegido, siendo por tanto es responsable de la memoria extendida (http://www.zator.com/Hardware/images/Ico_hoja.gif [5.1](http://www.zator.com/Hardware/H5_1.htm#Modo protegido)). Esta es también la razón por la que algunos test de diagnóstico de la placa base reporten errores de memoria por encima de 1 MByte cuando en realidad pueden deberse al 8742.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H2.htm#TOP)  Inicio](http://www.zator.com/Hardware/H2.htm#TOP)

**2.3  Acceso directo a memoria**

**§1  Presentación**

Puesto que la mayoría de datos que maneja el ordenador están antes o después en la memoria RAM, los intercambios entre esta y el resto de elementos son muy frecuentes.  En general este intercambio es conducido por el procesador, pero en determinados casos, la memoria pueda realizar intercambios directamente con los periféricos sin intervención del procesador.  Por ejemplo, el disco o una tarjeta de sonido, lo que conduce a un incremento del rendimiento del sistema.

**§2** **DMA**

Ya desde su concepción inicial, los diseñadores del PC dispusieron una arquitectura que permitiese este tipo de intercambios.  El mecanismo utilizado se conoce como **acceso directo a memoria** **DMA** ("Direct Memory Access"), e igual que ocurre con las excepciones, el sistema **DMA** dispone de algunos elementos hardware auxiliares que lo convierten en un subsistema autónomo dentro del bus externo.  Estos elementos son:

* Ciertas líneas dedicadas en el bus de control [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2_3.htm#Líneas de control).
* Un procesador específico, el **DMAC** ("DMA Controller"), que permite que puedan realizarse estos intercambios sin apenas intervención del procesador [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2_3.htm#Controlador DMA).
* Pequeñas zonas auxiliares de memoria, conocidas como **Registros de página** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2_3.htm#Registros de página).

**§3** **Líneas de control**

El bus de control (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2](http://www.zator.com/Hardware/H2.htm#Bus de control)) tiene líneas específicas para este tipo de intercambios, de forma que el **DMA** es un subsistema autónomo dentro del mecanismo general de intercambio de datos y control del bus.  Son las siguientes:

* Líneas **DRQ1** a **DRQ3** ("DMA request").  Utilizadas por los dispositivos que necesitan efectuar un acceso directo a memoria.
* Líneas **DACK1** a **DACK3** ("DMA acknowledge").  Se utilizan para acusar recibo de la petición DRQ correspondiente.
* **AEN** ("Access Enabled")  Cuando esta señal está alta, el controlador DMA tiene control sobre ciertas líneas del bus; precisamente las que gobiernan los intercambios con memoria y puertos (MEMR, MEMW, IOR, IOW, Etc).
* **MEMR** ("Memory Read")  Cuando se activa, esta señal indica a la memoria conectada al bus que escriba los datos en el bus de datos.
* **MEMW** ("Memory Write") Cuando se activa, indica a la memoria que almacene los datos situados en el bus de datos.
* Conexiones del controlador DMA con las patillas 30 y 31 de la UCP (http://www.zator.com/Hardware/images/Ico_hoja.gif [3.2.1](http://www.zator.com/Hardware/H3_2_1.htm#RQ/GT))

**§4** **Controlador DMA**

El mecanismo de acceso directo a memoria está controlado por un chip específico, el **DMAC** ("DMA Controller"), que permite realizar estos intercambios sin apenas intervención del procesador.  En los XT estaba integrado en un chip 8237A que proporcionaba 4 canales de 8 bits (puede mover solo 1 Byte cada vez); sus direcciones de puerto son 000-00Fh.  Posteriormente en los AT se instalaron dos de estos integrados y las correspondientes líneas auxiliares en el bus de control.

En contra de lo que podría parecer, el resultado no fue disponer de 8 canales, porque el segundo controlador se colgó en "Cascada" de la línea 4 del primero (más adelante se explica este concepto [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2_3.htm#Modos de operación)).  Los canales del segundo **DMAC** está asignado a las direcciones 0C0-0DFh y son de 16 bits.   Pueden mover 2 Bytes (de posiciones contiguas) cada vez.

Cada canal tiene asignada una prioridad para el caso de recibirse simultáneamente varias peticiones (los números más bajos tienen prioridad más alta).  Pueden ser utilizados por cualquier dispositivo que los necesite (suponiendo naturalmente que esté diseñado para soportar este modo de operación).   Cada sistema los asigna de forma arbitraria, pero hay algunos cuya asignación es estándar.

|  |  |
| --- | --- |
| **Canal** | **Uso** |
| **0** | En los primitivos PCs estaba dedicado al refresco de memoria.  En los sistemas actuales está libre. |
| **1** | Libre:  Generalmente a dispositivos SDLC [[3](http://www.zator.com/Hardware/H2_3.htm#[3])] (codificadores de audio) |
| **2** | Controlador de disquete |
| **3** | Controlador puerto de impresora ECP  [[2](http://www.zator.com/Hardware/H2_3.htm#[2])] |
| **4** | Libre en los XT. Ocupado en los AT (segundo controlador DMAC) |
| **5** | Libre (AT) |
| **6** | Libre (AT) |
| **7** | Libre (AT) |

**Nota**:  El Sistema Windows 98 incluye el programa MSINFO32.EXE con el que se pueden comprobar muchas características del sistema, incluyendo la asignación de canales DMA.  Los usuario de Linux pueden utilizar el comando dmesg | grep -i dma para obtener una relación de los dispositivos del sistema que, en principio, están habilitados para utilizar este tipo de transferencias.  Generalmente la relación contiene como mínimo los dos canales IDE (ide0 e ide1) y la unidad de disco (hda).

**§5** **Funcionamiento**

El mecanismo de acceso directo a memoria DMA es bastante complejo en sus detalles, y por supuesto, el movimiento de grandes volúmenes de datos entre memoria y un dispositivo requiere cierta intervención del procesador.  El movimiento se hace a ráfagas, y cada transferencia se inicia con una interrupción que obliga al procesador a suspender su tarea para permitir un nuevo intercambio.  A continuación intentaremos mostrar una visión general del proceso.

Empezaremos señalando que el DMA permite mover datos entre un puerto y memoria, o entre memoria y un puerto, pero no entre dos puertos o entre dos posiciones de memoria [[4](http://www.zator.com/Hardware/H2_3.htm#[4])].  Como veremos a continuación, cada transferencia DMA requiere cierta preparación previa;  conocer el volumen de datos a transferir (la más simple es de 1 byte) y la dirección de inicio del bufer de memoria involucrado (del que se leerán los datos o donde se escribirán).  Para esto dispone de dos registros para cada línea; el **contador** y el **registro de direcciones**.  Según se refieran a operaciones de lectura (memoria http://www.zator.com/Hardware/images/arrow_btt_R.gifdispositivo) o escritura (dispositivo http://www.zator.com/Hardware/images/arrow_btt_R.gifmemoria) reciben distintos nombres porque su significado difiere:

* Escritura:  Dirección de inicio  ("Write starting address").  Contador  ("Write starting word count")
* Lectura:  Dirección actual de lectura ("Read current address").  Contador  ("Read remaining word count")

Después de cada transferencia (de 1 byte) el registro de direcciones del DMAC es incrementado en una unidad, y el contador es disminuido en una unidad.  Cuando este último llega a cero, la transferencia ha concluido, el DMAC pone en nivel alto la línea T/C ("Terminal Count") en el bus de control (http://www.zator.com/Hardware/images/Ico_hoja.gif [2](http://www.zator.com/Hardware/H2.htm#Bus de control)) y procede a enviar al procesador la señal EOP ("End of Process").  A partir de este momento el controlador no puede realizar otra transferencia hasta que sea programado de nuevo por la UCP.  Aunque existen varios canales, el sistema de prioridades garantiza que solo uno de ellos puede estar en funcionamiento cada vez, de forma que sus funcionamientos no pueden solaparse, y la señal EOP se refiere forzosamente al canal activo en ese momento.

El sistema DMA tiene distintas formas de operación [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2_3.htm#Modos de operación), pero en el caso más simple, el proceso es como sigue:  cuando un dispositivo, que tiene asignado el canal **x** DMA, solicita una transferencia, activa la línea DRQx, avisando así al controlador DMAC de la petición.

Al recibir la señal, el controlador DMAC comprueba que dicha línea está programada y activada, y que no existe ninguna petición en otra línea DRQ de prioridad más alta.  A continuación solicita a la UCP que le ceda el control del bus enviándole una señal HRQ (Hold Request).

Cuando el procesador recibe la petición HRQ, termina la instrucción en curso, y una vez que está en condiciones de ceder el control del bus, deja "flotar" las señales relacionadas con su gobierno (MEMR, MEMW, IOR, IOW, Etc).  Es un estado denominado tri-state, en el que el procesador no coloca la patillas correspondientes en positivo ni en negativo.  A continuación envía una señal HLDA (Hold Acknowledge) al MDAC por la patilla correspondiente (http://www.zator.com/Hardware/images/Ico_hoja.gif [3.2.1](http://www.zator.com/Hardware/H3_2_1.htm#RQ/GT)), indicándole que debe tomar el control.

**Nota**:  Después de ceder el control del bus, la UCP puede seguir su proceso, pero en cuanto llega a una instrucción que necesita un dato de memoria que no esté previamente en su cache (si es un procesador moderno), debe detenerse y permanecer a la espera de ganar el control de nuevo.  Esta circunstancia es totalmente transparente para el Sistema Operativo (y desde luego para las aplicaciones que corren sobre él).  El único síntoma sería el tiempo inusitadamente largo que tarda el procesador en ejecutar la próxima instrucción.

A partir de este momento, el MDAC toma control del bus y monitoriza las señales que dejó flotar la UCP; activa las líneas MEMR, MEMW, IOR e IOW.  Si suponemos que la transferencia DMA programada consiste en transferir 1 byte a la dirección xxxxh (de memoria), el DMAC coloca la dirección en el bus de direcciones y envía una señal al dispositivo que solicitó la transferencia activando la línea DACKx correspondiente, lo que indica al dispositivo que debe depositar el dato en el bus de datos.  A continuación de la señal DACKx el controlador DMA espera un ciclo y, suponiendo que el dispositivo no haya solicitado tiempo adicional mediante la señal CHRDY del bus (http://www.zator.com/Hardware/images/Ico_hoja.gif [2](http://www.zator.com/Hardware/H2.htm#Bus de control)), desactiva las líneas MEMW e IOR, lo que permite que se complete la operación ("Latch").  Los datos del bus de datos son escritos en la dirección contenida en el bus de direcciones.

Suponemos que se trata de un dispositivo lento, por lo que desactiva la señal DRQx anunciando al DMAC que no lo necesita por el momento.  El DMAC responde desactivando la señal DACKx que recuerda al dispositivo que no puede transmitir más, y comprueba si existen otras peticiones que atender.  A continuación coloca en tri-state las señales del bus y envía al procesador la señal EOP.  El procesador contesta desactivando a su vez la señal HLDA; deja de flotar las líneas de bus y continua con el proceso que venía ejecutando.

**§6** **Modos de operación**

Aunque el mecanismo de transferencia se ajusta en lo sustancial al proceso anteriormente descrito, en realidad el controlador MDAC permite varios modos de operación:

**§6.1  Sencillo ("Single")**

Este modo transfiere solo un byte cada vez.  Después de cada transferencia el sistema cede el control del bus y debe adquirirlo de nuevo para transmitir el siguiente.  Es utilizada por dispositivos que solo pueden transmitir 1 byte cada vez a intervalos comparativamente muy largos (periféricos lentos).  Por ejemplo, el primitivo controlador de disquete del PC utilizaba este modo porque su bufer era de un byte.  El ciclo solicitud http://www.zator.com/Hardware/images/arrow_btt_R.gifadquisición del bus http://www.zator.com/Hardware/images/arrow_btt_R.giftransferencia http://www.zator.com/Hardware/images/arrow_btt_R.gifcesión del bus, se repite cuantas veces que sean necesarias.

**§6.2  Bloque ("Block")**

Las transferencias se realizan en bloques (un máximo de 64 KB).  Se supone que el periférico es capaz de escribir/leer los datos a velocidad sostenida, porque una vez iniciada la transferencia, continúa hasta que se completa.  En caso necesario el periférico puede solicitar una pausa momentánea mediante la línea CHRDY del bus (http://www.zator.com/Hardware/images/Ico_hoja.gif [2](http://www.zator.com/Hardware/H2.htm#Bus de control)), pero en general los dispositivos lentos utilizan el modo anterior.

**§6.3  Demanda ("Demand")**

Como en el caso anterior, la transferencia se realiza en bloques, pero solo tiene lugar mientras el dispositivo mantiene activada la línea de "Petición" correspondiente (DRQ).  En cuanto la desactiva, el DMAC cesa la transferencia y devuelve el control del bus al procesador.  Si el dispositivo necesita transferir más datos debe solicitar otra transferencia.

**§6.4  Cascada ("Cascade")**

El procedimiento de transferencia descrito anteriormente, corresponde al primitivo sistema **DMA** ISA.  Es conocido como DMA de tercer elemento ("Third party"), en relación a que en el proceso intervienen tres miembros:  El dispositivo que envía los datos, el que los recibe y el controlador DMAC que sería el tercero (observe que en realidad, los datos NO pasan por él).  Este enfoque pronto se mostró insuficiente para controlar los intercambios con dispositivos de alta velocidad, de forma que se arbitraron nuevas soluciones.  El **bus mastering** o DMA de primer elemento ("First party"), llamado así en relación a que es el dispositivo que realiza el intercambio de datos (con la memoria) el que toma directamente el control del bus en lugar de hacerlo el controlador DMAC.  En consecuencia, debe controlar las señales necesarias del bus (MEMR, MEMW, IOR, IOW, Etc).

Este procedimiento permite encadenar entre sí varios dispositivos DMA, y fue precisamente esta capacidad, la utilizada en el diseño del AT para añadir el segundo controlador.  DMAC#2 se colocó en cascada con la línea 4 del DMAC#1.  Cuando el DAMC#1 recibe una petición por el canal 4, cede el control del bus al dispositivo peticionario, el DMAC#2.

**§6.5  Autoinicialización ("Autoinicialize")**

En esta forma las transferencias se realizan al modo Sencillo o Demanda, pero cuando la UCP vuelve a tomar el control y el dispositivo está listo para enviar o recibir nuevos datos, no es necesario reprogramar la siguiente transferencia.  Si se estaban transfiriendo datos desde el dispositivo a un bufer de memoria, la UCP puede seguir añadiendo datos al bufer a continuación de los últimos transmitidos.  Si era una transferencia de de datos desde un bufer hacia el dispositivo, la UCP puede seguir leyendo datos desde la última posición de leída y escribiéndolos en el dispositivo.

Esta técnica se utiliza con dispositivos que tienen bufers pequeños. Por ejemplo, dispositivos de audio.  Supone cierta sobrecarga para la UCP, pero es la única forma de eliminar el retardo existente entre el momento en que termina una transferencia y se reprograma la siguiente.

**§7** **Registros de página**

Los chips 8237A y 8259A, elegidos por los diseñadores de IBM para los controladores **DMAC** y **PIC** del PC, en realidad habían sido diseñados para el 8085, un procesador de registros de 8 bits y bus de direcciones de 16 bits [[1](http://www.zator.com/Hardware/H2_3.htm#[1])] que le proporcionaban un espacio de direccionamiento de 64 KB.  (216 = 65.536); muy por debajo de las 1.048.576 (220) posiciones que podía direccionar el PC (http://www.zator.com/Hardware/images/Ico_hoja.gif [2](http://www.zator.com/Hardware/H2.htm#Bus de direcciones)).

Para que el **DMAC** pudiera seleccionar direcciones por encima del límite de 64 KB impuesto por su propio diseño, agregaron una pequeña memoria que pudiera contener la parte alta de las direcciones a manejar (desde las que leer, o en las que escribir, durante las operaciones DMA).  IBM las denominó **registros de página** y eran accesibles en e puerto 081-08Fh según el mapa adjunto (todas las direcciones son de lectura/escritura).

Puerto 0080-008Fh - Registros de página DMA

0081 DMA channel 2 address byte 2

0082 DMA channel 3 address byte 2

0083 DMA channel 1 address byte 2

0084 extra page register

0085 extra page register

0086 extra page register

0087 DMA channel 0 address byte 2

0088 extra page register

0089 DMA channel 6 address byte 2

008A DMA channel 7 address byte 2

008B DMA channel 5 address byte 2

008C extra page register

008D extra page register

008E extra page register

008F DMA refresh page register

La idea era componer la dirección (**Dt**) de la operación DMA, yuxtaponiendo dos direcciones (una especie de direccionamiento segmentado http://www.zator.com/Hardware/images/Ico_hoja.gif[H5.1](http://www.zator.com/Hardware/H5_1.htm#Direccionamiento segmentado), aunque no exactamente).  La parte alta (**D2**) se guardaría en el registro de página y la parte baja (**D1**) en el registro de direcciones del DMAC, de modo que Dt = D2 # D1. Por ejemplo, si la dirección de destino de la transferencia es 00123456h, el registro de direcciones contendrá 3456h, y el registro de página del canal correspondiente el valor 0012h.   Como puede verse, la forma de componer la dirección resultante **Dt** a partir de la información de cada segmento (en hexadecimal) sería:  Dt = D1 + FFFF + D2.

Es fácil comprobar que el sistema presenta algunos inconvenientes.  El valor más alto posible para **D1** es FFFFh, porque solo se dispone de 16 bits para su almacenamiento, y como puede verse en el mapa del puerto, el registro de página de cada canal es solo de 1 Byte, de forma que el valor más alto para **D2** es FFh.  En conjunto solo se dispone de 8 + 16 = 24 bits para la dirección, y como resultado, el sistema DMA del PC estándar no puede direccionar memoria por encima de 16 MB (224) [[5](http://www.zator.com/Hardware/H2_3.htm#[5])].

El diseño del sistema impone la limitación adicional de que el bloque máximo que puede manejarse en una sola transferencia es de 64 KB.  Está motivado porque el registro de página es totalmente independiente del DMAC y su funcionamiento no está sincronizado.  Cuando se inicia una operación, el valor **D2** del registro de página permanece invariable, mientras que, como hemos visto, el valor **D1** del registro de direcciones es incrementado con cada byte transferido.  Cualquiera que sea la dirección inicial XXh en **D2**, el rango de direcciones máximo que puede cubrirse corresponde al intervalo 0000-FFFFh en **D1**.  Después del último valor debería sumarse 1 a XXh y volver a comenzar por el valor **D1** = 0000h, cosa que no está contemplada en el diseño.

**§8  Evolución**

Como todo lo relacionado con la arquitectura del ordenador, el sistema de acceso directo a memoria ha sufrido una considerable evolución desde sus inicios.  Además de las anteriores, han aparecido nuevas formas de funcionamiento, como el Ultra DMA o DMA-33.

En 1996 Intel introdujo el chip 82374 **ESC** (EISA System Component) que entre otras características incluye un superset de la funcionalidad del primitivo 8237A y que servía tanto para los buses EISA como para los (entonces emergentes) PCIs.  Respecto al mecanismo DMA tradicional, presenta la mejora de que permite acceso directo a un espacio de direcciones de 32 bits.

La introducción del "Bus mástering" permitió que determinados dispositivos controlaran la gestión del bus en sustitución del controlador DMA.  Esta es la forma de funcionamiento de los dispositivos rápidos en los buses PCI.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H2_3.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H2_3.htm#TOP)

[1]  Este micro, lanzado por Intel en 1976, era en realidad una versión mejorada del 8080, que descendía a su vez del 8008...

[2]  **ECP** ("Enhanced Capabilities Port").  Este tipo de puerto paralelo bidireccional, desarrollado por HP y Microsoft, para conectar de forma económica periféricos de alta velocidad al PC (como impresoras de alto rendimiento y escáners), requiere acceso directo a memoria, algo que no era necesario en los puertos LPT normales **SPP** ("Standard Parallel Port") ni con los **EPP** ("Enhanced Parellel Port").  Actualmente la mayoría de PCs incluyen un chipset que puede manejar ambos tipos de puerto.  En estos casos, el programa de Setup de la BIOS permite seleccionar el puerto paralelo como ECP, EPP, o ECP+EPP.  En caso de utilizar ECP, la BIOS también puede permitir asignar el canal DMA correspondiente (generalmente el 3), aunque en los sistemas PnP es mejor confiar esta tarea a la propia BIOS.

[3]  Controlador síncrono serie de alta velocidad.  Generalmente utilizado en dispositivos de audio y comunicaciones.

[4]  Los movimientos de datos entre posiciones de memoria se realizan suficientemente bien mediante instrucciones de la UCP.

[5]  Esta cantidad de memoria RAM era inimaginable cuando se diseñó el PC, pero actualmente (2005) está dentro de lo posible.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H2_3.htm#TOP)  Inicio](http://www.zator.com/Hardware/H2_3.htm#TOP)

#### 2.4.1  Servicios BIOS

##### §1  Sinopsis

La **tabla de vectores de interrupción** **IDT** ("Interrupt Description Table") del PC es una zona de 1024 bytes en la memoria baja, que contiene un máximo de 256 vectores (punteros) a los programas que atienden las interrupciones **ISR**'s ("Interrupt service routine").  Estos punteros son de 4 bytes y están en forma segmentada (http://www.zator.com/Hardware/images/Ico_hoja.gif [5.1](http://www.zator.com/Hardware/H5_1.htm#Direccionamiento segmentado)); los dos primeros bytes indican el **desplazamiento** y los dos siguientes el **segmento**.

Cuando el procesador necesita encontrar alguna **ISR**, el dato de inicio es su número de servicio (también conocido como número de interrupción); a partir de este dato, multiplicando por 4 se obtiene la dirección absoluta del puntero (la **IDT** comienza en la dirección 0).

Por ejemplo:  Cuando se pulsa una tecla, se genera una interrupción hardware (http://www.zator.com/Hardware/images/Ico_hoja.gif [2.4](http://www.zator.com/Hardware/H2_4.htm#Tipos de interrupción)) con el número 9; su dirección es 9 x 4 = 36d = 24h, que en forma segmentada sería 0000:0024.

**Nota**:  Para obtener el contenido de esta dirección (que es la dirección de inicio del ISR del teclado) puede utilizarse el programa DEBUG (http://www.zator.com/Hardware/images/Ico_hoja.gif [1.7.1w2](http://www.zator.com/Hardware/H1_7_1.htm)).  La forma es la siguiente:

* Invocar  DEBUG desde una ventana DOS (suponemos que estamos en Windows; el "prompt" es un guión "-")
* introducir el comando D 0000:0024 L 4  (pedimos que nos muestre 4 bytes a partir de la dirección indicada).
* Salir de Debug con Q

En mi PCm la respuesta al comando es:

0000:0020           D2 08 DB 0A

El grupo a la derecha indica el contenido hexadecimal de los 4 bytes solicitados, en el orden creciente de posiciones de memoria.  Son por tanto, los contenidos de las posiciones 0000:0024; 0000:0025; 0000:0026 y 0000:0027.  Teniendo en cuenta lo señalado respecto al almacenamiento (back-word http://www.zator.com/Hardware/images/Ico_hoja.gif[5](http://www.zator.com/Hardware/H5.htm#Forma de almacenamiento)), en mi PC, la dirección segmentada de la rutina de manejo del teclado es  0ADB:08D2.  Como puede verse, está en el segmento 0ADB, correspondiente a la zona baja de la memoria de usuario.

**Nota**:  El valor 0000:0020 a la izquierda del resultado podría conducir a error, dado que la dirección solicitada no es esta.  La justificación es que el inicio de cada línea no puede ser una dirección cualquiera, sino un múltiplo de 16 (Fh).  En nuestro caso 0024h (36d) **no** es divisible por 16.  **debug** comienza la línea en la posición señalada, pero elimina las posiciones anteriores, dejando solo las que hemos pedido.  Para ver el párrafo completo, puede hacerse:

-D 0000:0020 L F

Aquí le hemos pedido que que muestre 16 bytes (Fh) a partir de la dirección indicada, y se obtiene el párrafo que sigue (hemos resaltado los resultados del caso anterior):

0000:0020 00 00 00 **D2 08 DB 0A** 0C-3A 00 BE 09 52 00 BE

##### §2  Servicios BIOS

Durante la secuencia de arranque del sistema, el programa de inicio se ocupa de iniciar algunos de estos punteros, que corresponden con programas (aquí se llaman "servicios"), que están en la propia BIOS [[1](http://www.zator.com/Hardware/H2_4_1.htm#[1])];  son los denominados **servicios BIOS**, a los que dedicamos el presente epígrafe.

##### §3  Servicios del sistema

Además de los anteriores, en la IDT se situan otra serie de vectores que apuntan a rutinas del propio Sistema Operativo.  Estos vectores son situados en el proceso de carga del Sistema, y apuntan siempre a zonas de memoria RAM.

##### §4 Resumen de servicios del BIOS

Los servicios proporcionados por el BIOS pueden agruparse en cuatro grandes grupos:

* Servicios a periféricos (incluyen video, disquete, teclado, puertos de comunicaciones, impresora y pantalla).
* Servicios de estado del equipo (incluyen conocer los dispositivos presentes, incluyendo el tamaño de la memoria disponible).
* Servicio de hora (posibilidad de leer y escribir el número de "ticks de reloj pasados desde media noche)
* Servicios adicionales (permiten activar el BASIC y reiniciar el sistema)

Estos servicios son atendidos por doce vectores que se detallan un poco más ampliamente en el siguiente epígrafe, pero es importante conocer que el BIOS proporciona varios subservicios dentro de cada interrupción (por ejemplo, la  16, de video, incluye 19 subservicios, cada uno de los cuales tiene una funcionalidad específica).  La invocación de estas rutinas debe hacerse en lenguaje ensamblador, y los diferentes subservicios son invocados mediante la colocación de determinados valores en ciertos registros del procesador [[2](http://www.zator.com/Hardware/H2_4_1.htm#[2])].  En cierto sentido este método es equivalente a la invocación de una función a la que se pasan distintos parámetros en función del comportamiento (respuesta) que deseamos obtener.

##### §5  Detalle de servicios del BIOS

La tabla de vectores de interrupción del PC tiene posiciones que son estándar, esto significa que algunos números de interrupción corresponden a un mismo servicio en todas las máquinas.  En la tabla adjunta se han señalado aquellos servicios que corresponden a la BIOS, indicando el número de interrupción (decimal), la dirección absoluta del vector (en hexadecimal) y el uso de la **ISR** correspondiente.

|  |  |  |
| --- | --- | --- |
| **Numero** | **Dirección** | **Uso** |
| **5** | 0014h | Servicio de impresión del contenido de la pantalla.  La invocación de este servicio provoca el mismo resultado que la combinación de teclas  SHIFT+CTRL (que utilizan este servicio). |
| **8** | 0020h | El reloj del sistema genera interrupciones con este número con una frecuencia de 18.21 veces por segundo aproximadamente (se conocen como "ticks") [[3](http://www.zator.com/Hardware/H2_4_1.htm#[3])]. Este servicio incrementa en una unidad la cuenta del reloj (almacenada en la memoria de datos de la ROM-BIOS http://www.zator.com/Hardware/images/Ico_hoja.gif[0.3.3](http://www.zator.com/Hardware/H4.htm#Inicialización)).  Del valor de este contador, que es puesto a cero cada 24 horas, se basan los servicios de hora del sistema (servicio 26 [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H2_4_1.htm#26)). |
| **16** | 0040h | Servicios de video.  Contiene 16 subservicios tales como ajuste del tamaño y desplazamiento del cursor, escritura y lectura de un carácter o de un píxel, desplazamiento vertical ("Scroll"), ajuste y lectura del modo de video, etc. |
| **17** | 0044h | Servicio de componentes del equipo.  Proporciona una palabra de 16 bits que contiene información básica sobre los componentes instalados en el ordenador. |
| **18** | 0048h | Este servicio informa del tamaño de la memoria instalada en el sistema. |
| **19** | 004Ch | La ROM BIOS proporciona seis servicios estándar de disquete: Reinicialización; obtención del estado; lectura; escritura y verificación de sectores, y formateo de pistas.  A partir de la introducción del IBM PC AT, se introdujeron 12 nuevos servicios para disco, incluyendo recalibrado de la unidad, diagnóstico del controlador, etc. |
| **20** | 0050h | En esta interrupción se agrupan diversos subservicios de comunicaciones para puertos serie proporcionados por la BIOS:  Inicializar los parámetros de inicio del puerto; enviar un carácter; recibir un carácter; obtener el estado del puerto. |
| **21** | 0054h | Incluye los servicios de cassette.  En realidad es una reliquia prehistórica, ya que el PC tenía posibilidad de una cassette opcional. |
| **22** | 0058h | Incluye los servicios de teclado que incluye tres subservicios:  Lectura de un carácter del buffer del teclado; Informar si hay algún carácter en el buffer; selección mayúsculas/minúsculas (estado del "Shift"). |
| **23** | 005Ch | Incluye tres subservicios para impresora:  Enviar un carácter al puerto paralelo; Inicializar el puerto;  obtener un informe esquemático del estado de la impresora. |
| **24** | 0060h | Esta es otra reliquia prehistórica.  Se trata del cargador del intérprete BASIC; este lenguaje que estaba presente en la ROM de los primeros PC's. |
| **25** | 0064h | Este servicio es la rutina de puesta en marcha ("bootstrap") del equipo.  El resultado de su activación es equivalente a su puesta en marcha.  Se denomina reinicio en caliente, y es parecido al efecto que se consigue por el teclado con la combinación CTRL+ALT+DEL. |
| **26** | 0068h | Este servicio suministra la hora del sistema mediante la inspección del contador de ticks de reloj pasados desde media noche (ver servicio 8 [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H2_4_1.htm#8)).  Incluye dos subservicios, que permiten respectivamente leer y escribir el valor del contador de ticks.  **Nota**:  La librería estándar C++ dispone de una función **clock**() que devuelve el valor del tiempo transcurrido desde el inicio del programa.  Esta función se basa en la inspección del contador de ticks. |

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H2_4_1.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H2_4_1.htm#TOP)

[1]  El proceso de carga puede copiar estos programas en en otros puntos de la memoria RAM (lo que ocurre si se utiliza una shadow-BIOS por ejemplo, http://www.zator.com/Hardware/images/Ico_hoja.gif[H4](http://www.zator.com/Hardware/H4.htm#Tipos de BIOS)).

[2]  La mejor referencia al respecto que conozco es:  Peter Norton "The Peter Norton Programer's Guide to the IBM PC" 1985 Microsoft Press.

[3]  El valor resulta de dividir la la frecuencia 4.772720 MHz del reloj del IBM PC original (http://www.zator.com/Hardware/images/Ico_hoja.gif [2](http://www.zator.com/Hardware/H2.htm#Elementos de temporización)) por 218.  El resultado: 4.77272 · 106 / 262144 = 18.206.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H2_4_1.htm#TOP)  Inicio](http://www.zator.com/Hardware/H2_4_1.htm#TOP)

## 5  Memoria interna

##### §1  Sinopsis

Al tratar de la placa-base, comentamos que la **memoria interna** es la que se encuentra físicamente dentro del sistema constituido por la placa-base, o en tarjetas de circuito impreso directamente conectadas a ella.  Dentro de este tipo de memorias nos interesa distinguir las siguientes:

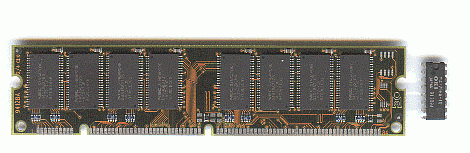
* Los registros del procesador (http://www.zator.com/Hardware/images/Ico_hoja.gif [H3.2](http://www.zator.com/Hardware/H3_2.htm#Los registros))
* Las cachés interna y externa (http://www.zator.com/Hardware/images/Ico_hoja.gif [H5.2](http://www.zator.com/Hardware/H5_2.htm))
* La memoria BIOS (http://www.zator.com/Hardware/images/Ico_hoja.gif [H4](http://www.zator.com/Hardware/H4.htm))
* La memoria RAM

##### §2  La memoria RAM

Aunque la clasificamos como "Interna", la memoria **RAM** ("Random access memory") es también en alguna forma una memoria "externa"; en el sentido que está situada fuera del procesador (el "Cerebro" del ordenador); es como su bloc de notas.  El procesador tiene una memoria raquítica (se reduce a sus registros), pero una gran facilidad para manejar este almacenamiento auxiliar. De hecho, gran parte del trabajo del procesador se concreta en traer y llevar datos desde RAM hasta sus propios registros.

Atendiendo a sus características físicas, las memorias RAM se dividen en dos grandes grupos:  estáticas **SRAM** ("Static RAM"), y dinámicas **DRAM** ("Dynamic RAM").  Ambas comparten la característica de perder su contenido cuando se apaga el sistema. Pero las DRAM tienen además la necesidad de que su contenido sea constantemente actualizado.

Como el resto de elementos hardware, el desarrollo de la tecnología de memorias ha sido incesante; diríamos meteórico.  La constelación de siglas es inacabable: FPM DRAM ("Fast Page Mode");  EDO RAM ("Extended Data Out");  BEDO RAM ("Burst EDO RAM"); SDRAM ("Synchronous DRAM");  RDRAM ("Rambus DRAM");  DDR-RAM ("Double data rate RAM");  SLDRAM ("SyncLink DRAM");  VRAM ("Video RAM");  SGRAM ("Sinchronous Graphics RAM"); etc.

Los primeros PCs no llegaron a conocer las memorias de núcleos de ferrita, puesto que ya montaban varias decenas módulos de DRAM encapsulados en chips DIP ("Dual In-line Package") de 16 contactos sobre zócalos [[3](http://www.zator.com/Hardware/H5.htm#[3])].  Actualmente (2001) coexisten dos tipos principales de encapsulado:  El **SIMM** ("Single In-line Memory Module") de 72 contactos, y el **DIMM** ("Dual In-line Memory Module") de 168 contactos. 

En le imagen, un módulo DIMM de 168 contactos con 16 MB de SDRAM junto con un antiguo chip de 16 contactos con 2 KB de DRAM.

##### §2.1  Estructura lógica

Desde el punto de vista lógico, la memoria RAM puede considerarse como una serie de varios miles (o millones) de bits que pueden ser accedidos para lectura y escritura en grupos de 8 (en Bytes) mediante una dirección.  Pero hay dos aspectos que pueden interesar al programador:  uno se refiere a como se guardan algunos datos; otro es relativo a como se almacenan los ejecutables.

##### §2.2  Forma de almacenamiento

Respecto al primer punto, tener en cuenta que, aunque la arquitectura de PC permite manejar la memoria en bytes individuales.  Muchas operaciones implican guardar palabras de 16 bits.  De estos 2 octetos adyacentes, el de la izquierda es el más significativo y el de la derecha el menos.  En estos casos, el byte menos significativo se guarda en la posición más baja y el más significativo a continuación, en la posición más alta.  Esta forma de almacenamiento se denomina de **palabras invertidas** ("Back-words") o little endian (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.6a](http://www.zator.com/Cpp/E2_2_6a.htm)).

Esta forma especial de almacenar información se refiere a números, tanto enteros como decimales (de punto flotante), y no solo a los de 16 bits.  También a los demás, de 32 y 64 bytes (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.4](http://www.zator.com/Cpp/E2_2_4.htm)).

##### §2.3  Carga de ejecutables

Respecto al segundo punto, ni que decir tiene que para ejecutar un programa, éste debe ser previamente cargado en memoria. Pero en la mayoría de los casos [[1](http://www.zator.com/Hardware/H5.htm#[1])] no se trata de una carga del fichero tal cual se encuentra en el disco, sino que requiere un "acomodo" especial.  De este trabajo se encarga un programa especial (de carga), y se exige que la primera parte del contenido de un fichero .EXE contenga precisamente la información sobre "como" se realizará la acomodación antes aludida.  En el caso de Windows, los ejecutables deben contener esta información en un formato específico, denominado **nuevo formato de fichero ejecutable** ("New Executable file format"); una especificación de MS para las aplicaciones que deban correr bajo sus Sistemas [[2](http://www.zator.com/Hardware/H5.htm#[2])].

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H5.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H5.htm#TOP)

[1]   Es el caso de los ficheros .EXE, pero en los ejecutables .COM sí existe concordancia entre la imagen del programa, tal como se almacena en memoria y el contenido del fichero grabado en disco.  Este último formato no puede utilizarse con todos los ejecutable, pero es la razón por la que los .COM son de carga más rápida que los .EXE.

[2]  Gran parte de esta información está contenida en el **fichero de definición** (http://www.zator.com/Hardware/images/Ico_hoja.gif [E1.4.4a](http://www.zator.com/Cpp/E1_4_4a.htm)).

[3]  Las dilataciones y contracciones térmicas de la placa, originadas por los periodos de funcionamiento y reposo, tendían a expulsar estos chips de sus alojamientos, con lo que eran frecuentes los errores de paridad y tener que reapretarlas empujándolas con el dedo.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H5.htm#TOP)  Inicio](http://www.zator.com/Hardware/H5.htm#TOP)

**5.2  Memoria CACHE**

**§1  Sinopsis**

Aunque de origen inglés [[4](http://www.zator.com/Hardware/H5_2.htm#[4])], la palabra **cache** ha tomado carta de naturaleza en Español (no se si los académicos se habrán enterado de ello, por si acaso, la españolizamos añadiéndole un acento).  Utilizada en informática significa memoria temporal; generalmente de existencia oculta y automática para el usuario, que proporciona acceso rápido a los datos de uso más frecuente o previsible.  Por ejemplo, el "Caché" de disco [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H5_2.htm#Caché de disco)es un área de memoria donde el Sistema transfiere los datos que supuestamente serán accedidos de inmediato.  Si leemos un "cluster" [[1](http://www.zator.com/Hardware/H5_2.htm#[1])] el sistema puede disponer en esta memoria "cache" los clusters que siguen en la estructura lógica, de forma que, si seguimos efectuando lecturas, lo más probable es que los próximos datos estén ya en memoria y puedan ser accedidos de forma inmediata [[2](http://www.zator.com/Hardware/H5_2.htm#[2])].

La utilización de este tipo de memorias no es sino la generalización de un principio de uso común en la vida diaria; poner más a mano las cosas de uso más frecuente.  Se basa en dos suposiciones que generalmente resultan ciertas:

* Los ordenadores tienden a utilizar las mismas instrucciones y (en menor medida), los mismos datos repetidamente.
* La información necesitada se encuentra almacenada de forma adyacente, o cuando menos muy cercana, en memoria o disco.

**§2** **Tipos de cache**

Desde el punto de vista del hardware, existen dos tipos de memoria cache; **interna** y **externa**.  La primera, denominada también **cache primaria**, **caché de nivel 1** o simplemente **caché L1** (Level one) [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H5_2.htm#Caché interna).   La segunda se conoce también como **cache secundaria**, **cache de nivel 2** o **cache L2 [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H5_2.htm#Caché externa)**.

Desde el punto de vista funcional, existen cachés específicas de algunos dispositivos, por ejemplo, de disco. También se distingue entre caché de lectura y de escritura.

**§2.1** **Caché interna**

Es una innovación relativamente reciente [[3](http://www.zator.com/Hardware/H5_2.htm#[3])]; en realidad son dos, cada una con una misión específica:  Una para datos y otra para instrucciones.  Están incluidas en el procesador junto con su circuitería de control, lo que significa tres cosas:  comparativamente es muy cara; extremadamente rápida, y limitada en tamaño (en cada una de las cachés internas, los 386 tenían 8 KB; el 486 DX4 16 KB, y los primeros Pentium 8 KB).  Como puede suponerse, su velocidad de acceso es comparable a la de los registros, es decir, centenares de veces más rápida que la RAM.

**§2.2** **Caché externa**

Es más antigua que la interna, dado que hasta fecha "relativamente" reciente estas últimas eran impracticables.   Es una memoria de acceso rápido incluida en la placa base, que dispone de su propio bus y controlador independiente que intercepta las llamadas a memoria antes que sean enviadas a la RAM (http://www.zator.com/Hardware/images/Ico_hoja.gif [H2.2](http://www.zator.com/Hardware/H2_2.htm#Backside bus)  Buses locales).

La caché externa típica es un banco SRAM ("Static Random Access Memory") de entre 128 y 256 KB. Esta memoria es considerablemente más rápida que la DRAM ("Dynamic Random Access Memory") convencional, aunque también mucho más cara [[5](http://www.zator.com/Hardware/H5_2.htm#[5])] (tenga en cuenta que un aumento de tamaño sobre los valores anteriores no incrementa proporcionalmente la eficacia de la memoria caché).  Actualmente (2004) la tendencia es incluir esta caché en el procesador.  Los tamaños típicos oscilan entre 256 KB y 1 MB.

**Nota**:  En 1997, con la introducción del procesador Pentium II, Intel abandonó el denominado **zócalo 7** utilizado hasta entonces en sus procesadores, en favor del denominado **Slot-1**.  La razón argüida era precisamente la inclusión de la caché **L2** en la cápsula del procesador.

**§3** **Caché de disco**

Además de las anteriores, que son de propósito general, existe una caché de funcionalidad específica que se aloja en memoria RAM estándar. Es la **caché de disco** (nos hemos referido a ella en la introducción de este epígrafe), destinada a contener los datos de disco que probablemente sean necesitados en un futuro próximo y los que deben ser escritos.  Si la información requerida está en chaché, se ahorra un acceso a disco, lo que es centenares de veces más rápido (recuerde que los tiempos de acceso a RAM se miden en nanosegundos y los de disco en milisegundos http://www.zator.com/Hardware/images/Ico_hoja.gif[E1.7.1](http://www.zator.com/Cpp/E1_7_1.htm#Unidades de medida)  Unidades de medida).

**Nota**:  Existe un mecanismo parecido al de caché de disco que aquí se describe pero que funciona en sentido inverso.  Es decir, aloja en disco los datos que no pueden ser almacenados en la memoria RAM.  Es el sistema de **memoria virtual**, al que nos referiremos al tratar de la memoria.

El funcionamiento de la caché de disco se basa en dos esquemas de operación.  La **lectura adelantada** ("Read-ahead") y la **escritura retrasada** ("Write-behind").  La primera consiste en anticipar lo que se necesitará de forma inmediata y traerlo a la caché.  Por su parte, la escritura retrasada consiste en mantener los datos en caché hasta que se producen momentos de desocupación del sistema de disco.  En este caso la caché actúa como memoria tampón o "buffer" intermedio, y no se obliga al subsistema a realizar físicamente ninguna escritura, con lo que las cabezas quedan libres para nuevas lecturas.

http://www.zator.com/Hardware/images/Bombilla_.gif  Puesto que los cachés de disco de escritura retrasada mantienen los datos en memoria volátil después que "supuestamente" se han escrito en el dispositivo,  una caída accidental del sistema, por fallo de energía o apagado intempestivo, puede producir pérdidas de los datos alojados en la caché en ese momento (es esta una de las razones por las que los sistemas Windows y Linux exigen un proceso especial de apagado, que a veces tarda unos segundos, en los que observamos una intensa actividad del sistema de disco).

**Nota**:  La mayoría de los lenguajes disponen de métodos para forzar una escritura "real" de los datos vaciando la caché de disco; suelen ser sentencias del tipo commit, flush etc.  Es una práctica de seguridad aconsejable, y señal de programación cuidadosa, realizar un vaciado de "buffers" después de cada transacción importante siempre que las circunstancias lo permitan [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H5_2.htm#Vaciado de buffers en C++).

**§3.1  Caché de disco en MS DOS y  Windows**

La cache de los sistemas MS DOS y de los primeros sistemas Windows se denominaba **SmartDrive**.  Por su parte, los nuevos Sistemas de 32 bits disponen de un controlador virtual denominado **VCACHE** que utiliza un esquema de funcionamiento de **lectura adelantada** y **escritura atrasada** [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H5_2.htm#Caché de disco)para proporcionar servicios de cache a las máquinas virtuales (http://www.zator.com/Hardware/images/Ico_hoja.gif [E0.2](http://www.zator.com/Cpp/E0_2.htm#La máquina virtual)).

VCACHE tiene la ventaja cachear ficheros en discos de red, y de permitir cambiar en tiempo de ejecución la cantidad de memoria destinada a este menester.  Cuando la actividad del disco es elevada pero la ocupación de memoria es baja, VCACHE incrementa su tamaño para realizar la mayor cantidad de operación en RAM, evitando de este modo accesos a disco.  Por ejemplo, si la aplicación abre un fichero para lectura/escritura, es posible que VCACHE vuelque la totalidad del fichero a memoria; posteriormente, quizás cuando se cierre el fichero, la imagen de memoria sea volcada de nuevo al disco.  Si por el contrario la actividad de disco es pequeña y la ocupación de memoria es alta, VCACHE disminuye su propio tamaño con objeto de aumentar la RAM disponible para las aplicaciones.

**§3.2** **Vaciado de buffers en C++**

En la terminología C++ los flujos que son cacheados se denominan "buffered".  A este respecto, los compiladores C/C++ disponen de su propio sistema de caché para ficheros de disco.  Esta caché se denomina **de ejecución** (runtime), para distinguirla de la caché **del Sistema**.  Así mismo, disponen de recursos en la Librería Estándar para forzar su vaciado en caso necesario; para esto se recurre a la funciones **fflush** (para ficheros abiertos con **fopen**) y **flush** (para los flujos de salida, "ostreams").

Sin embargo, no olvide que el vaciado de la caché del compilador se realiza sobre la del Sistema, que está por debajo (recuerde que el Software tiene una estructura de capas http://www.zator.com/Hardware/images/Ico_hojaNw.gif[E1.7w1](JavaScript:pUp('../Cpp/E1_7W1.htm'))), y que el SO decide por su cuenta cuando es el momento oportuno para realizar físicamente la escritura de los discos.  Esto significa que una seguridad total solo se alcanza forzando la escritura de la caché del Sistema, y esto naturalmente depende de la plataforma utilizada.

**Nota**:  Además de las posibilidades ofrecidas en la Librería Estándar, el compilador MS Visual C++ para Windows ofrece la función **\_flushall**, que fuerza el vaciado de la caché de ejecución de todos los ficheros abiertos.  También dispone de la librería COMMODE.OBJ, que enlazada con la aplicación, fuerza que las llamadas a **fflush** y a **\_flushall** escriban directamente los buffers al disco en vez de a la caché del Sistema.

El siguiente **ejemplo** muestra algunas formas de vaciado de los buffers del compilador y de la caché del Sistema

#include <stdio.h>

#include <ofstream.h>

void funcES1() {

  FILE\* Fichero1;               // L.5

  fflush(Fichero1);             // L.6

  \_commit(\_filenum(Fichero1));  // L.7

}

void funcES2() {

  ofstream Fichero2;                // L.11

  Fichero2.flush();                 // L.12

  \_commit(Fichero2.rdbuf()->fd());  // L:13

}

void main() {                   // ========

  funcES1();

  funcES2();

}

**Comentario**

L.5 abre el fichero Fichero1 para lectura y escritura.

La llamada a **fflush** en L.6 fuerza al programa a vaciar al SO la caché de ejecución asociados al fichero1.

La llamada a **\_commit** en L.7 obliga al cache de disco del Sistema vaciar sus buffers.

L.11 se instancia un objeto Fichero2 de la plantilla basic\_ofstream<char>, para realizar escritura.

La línea L.12 invoca el método **flush** para dicho objeto.  Esta invocación fuerza al programa a vaciar al SO los buffers de ejecución asociados con Fichero2.

La invocación a **\_commit** en L.13 obliga al caché de disco del Sistema a vaciar los buffers al disco.   Esta función requiere un manejador "handle" referido al fichero; en este caso, el manejador se obtiene mediante una invocación al método **rdbuf**.  ofstream.rdbuf()->fd().

**Nota**:   **\_commit** es una función de MS Visual C++ que obliga al Sistema a vaciar los buffers de un fichero determinado.  No es estándar C++ y solo funciona en Sistemas MS de 16 bits.   En los de 32 bits se recurre a la librería **Commode.obj** según se ha indicado [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H5_2.htm#Nota-1).

**§4  Rendimiento de la caché**

El funcionamiento de la caché de lectura se parece al de un adivino; debe anticipar lo que ocurrirá en el futuro.   Si el dispositivo que está siendo cacheado encuentra los datos en la caché, habrá un éxito ("hit"), en caso contrario, un fracaso ("miss").  Los sistemas de caché actuales son capaces de proporcionar una tasa de éxitos superior al 90%.

Como puede figurarse el lector, construir un mecanismo de caché no es una tarea baladí.  Se requieren esquemas de funcionamiento que atiendan de forma simultanea y balanceada diversos factores:

* Discriminar que información debe ser almacenada y cual descartada.
* Decidir la organización interna de este almacenamiento.
* Manejar las peticiones de lectura.  Esto exige disponer de un mecanismo de intercepción de las peticiones del dispositivo que está siendo cacheado.
* Manejar las peticiones de escritura.  Interceptar las peticiones de escritura del dispositivo a cachear.

**§5  Caché oportunista**

Existe un tipo especial que podríamos considerar "de aplicación", denominada **caché oportunista** ("Opportunistic cache").  Está relacionada con los problemas de bloqueos de ficheros en entornos multiusuario en los que distintas aplicaciones pueden acceder a los mismos datos.

En estos casos, los Sistemas Operativos disponen de mecanismos para que un usuario (programa de aplicación) obtenga el bloqueo de todo un fichero o parte de él.  La teoría es que mientras se mantenga el bloqueo, ningún otro usuario puede modificar el fichero (tal vez si leerlo), y que una vez finalizadas las modificaciones, el usuario desbloquea el fichero para que otros puedan utilizarlo.  Sin embargo, en determinadas aplicaciones de red, y con objeto de aumentar el rendimiento, se utiliza un sistema mixto denominado bloqueo oportunista **oplock** ("Opportunistic locking"), en el que el usuario comunica al Sistema que utilizará esta modalidad [[6](http://www.zator.com/Hardware/H5_2.htm#[6])].  Para ello, obtiene una copia de la totalidad del fichero, que almacena un una caché local oportunista.  De esta forma, las operaciones son más rápidas que si tiene que realizarse a través de la red las peticiones de distintos trozos, junto con las correspondientes solicitudes de bloqueo/desbloqueo.  Finalmente, cuando el usuario ha finalizado las operaciones con el fichero, devuelve al servidor una copia actualizada.

El problema se presenta cuando, en el intermedio, otro usuario solicita utilizar el mismo fichero.  La incidencia es especialmente frecuente cuando el fichero a manejar es muy grande.  Porque entonces, incluso para una pequeña modificación, el primer usuario puede demorarse bastante en devolver la versión modificada al servidor.  La solución adoptada para evitar demoras excesivas, consiste en que, al recibir la petición del segundo usuario, el Sistema envía al primero una orden de interrumpir el **oplock** y devolver el fichero tal como está en ese momento para que el segundo usuario pueda utilizarlo.

Aunque no exento de problemas, especialmente en redes poco fiables, el sistema permite aumentos del rendimiento del orden del 30%. No tanto por el sistema de bloqueo utilizado, como por el hecho de que los datos hayan sido previamente cacheados por el usuario.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H5_2.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H5_2.htm#TOP)

[1]  También denominado "Allocation Unit" o **unidad de asignación** en Español.  Es el conjunto de sectores de disco que son tratados por el sistema como una sola unidad de almacenamiento.  Su tamaño (número de sectores de disco) depende del SO y del tamaño de la partición lógica (MS DOS y Windows utilizan un cluster de 1 sector en los disquetes de 3.5").  Un "Cluster" puede estar ocupado o disponible, pero no parcialmente ocupado.  El **Sistema de Archivo** de un Sistema Operativo es precisamente una forma de controlar, asignar, desasignar y acceder clusters para alojar ficheros en disco (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2](http://www.zator.com/Hardware/H8_1_2.htm#Pistas, sectores y clusters)).

"Cluster" tiene también otra acepción en inglés:  Designar un almacenamiento redundante de datos en sistemas distintos.

[2]  Hasta hace pocos años, en que la tecnología de discos ha producido unidades extremadamente silenciosas, a un oído experimentado le bastaba escuchar la "música" de los servos para saber si el sistema de "cache" de disco estaba instalado o no.

[3]  Introducida con el Intel 386 SLC que tenía una cache interna para instrucciones y datos, mientras que en los Pentinum® instrucciones y datos disponen de cachés separadas.

[4]  Bueno, eso es lo que yo creía porque aprendí la palabreja de textos ingleses, pero D. Julián Cirielli, desde Bs. As. Argentina, me aclara amablemente que en realidad, la palabra proviene del francés, y significa escondite/escondida.  He preferido mantener intacta mi errónea redacción anterior e incluir aquí la corrección junto con mi agradecimiento.

[5]  El motivo es la propia construcción de la memoria estática, constituida por conjuntos de seis transistores por cada bit almacenado, lo que las hace mucho más voluminosas (y caras) que las memorias dinámicas de capacidad comparable; estas últimas están consituidas por un conjunto de 1 transistor y un condensador por cada bit.  La descarga del condensador es lo que hace que requieran una actualización (refresco) cada 15 µs (microsegundos) aproximadamente, lo que añade una dificultad adicional, pues durante la actualización el sistema queda paralizado (en el bus hay una línea específica, DACK-0, para indicar que se está produciendo este refresco de la memoria dinámica http://www.zator.com/Hardware/images/Ico_hoja.gif[H2](http://www.zator.com/Hardware/H2.htm#Bus)).

Actualmente (2001), las SRAM tienen tiempos de acceso del orden de 2 a 15 ns (nanosegundos), mientras que en las DRAM es del orden de 60 ns.

[6]  Por ejemplo, Samba el popular sistema Linux para compartir recursos sobre redes TCP/IP, puede utilizar este tipo de caché oportunista con sus clientes.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H5_2.htm#TOP)  Inicio](http://www.zator.com/Hardware/H5_2.htm#TOP)

## 8  Almacenamiento externo

##### §1  Sinopsis

En éste capítulo, abordaremos las características de los **dispositivos de almacenamiento externo** más usuales en los ordenadores personales. Principalmente unidades de disco en sus distintas versiones (de plato giratorio y de estado sólido). Así como unidades ópticas (CDs y DVDs) y otras unidades de almacenamiento magnético, como cintas y "streamers".

Como todo lo que implique un principio de clasificación, al referirnos al ordenador, y más concretamente al PC, los conceptos interno/externo pueden ser más que discutibles. Para nuestro propósito, consideramos dispositivos **almacenamiento interno** la memoria RAM (http://www.zator.com/Hardware/images/Ico_hoja.gif [5](http://www.zator.com/Hardware/H5.htm)) y los registros del procesador (http://www.zator.com/Hardware/images/Ico_hoja.gif [3.2](http://www.zator.com/Hardware/H3_2.htm#Los registros)). Todos los demás los consideraremos "externos". Esto, con independencia de que el dispositivo concreto, por ejemplo un disco, pueda estar alojado físicamente en el mismo chasis que los primeros (junto con la placa-base  (http://www.zator.com/Hardware/images/Ico_hoja.gif [2](http://www.zator.com/Hardware/H2.htm)), o en un chasis independiente. Que en estas fechas, puede estar sobre la misma mesa que el resto de componentes del PC, o a miles de Kilómetros de distancia.

Como en el resto de capítulos de esta obra, no pretendemos aquí una descripción exhaustiva de sus características técnicas.  Solo proporcionar algunos detalles interesantes para el usuario que desee saber un poco más sobre estos dispositivos tan cotidianos en nuestras vidas. Dado que me está leyendo, doy por supuesto que tiene un cierto interés en el tema. También que posiblemente tenga que convivir con ellos a menudo. En estas circunstancias, estoy seguro que conocer ciertos detalles de su estructura interna, y de los principios de su funcionamiento, le ayudarán a desenvolverse mejor con estos cacharos, e incluso a salir airoso cuando se presenten pequeños problemas (si trata con equipos informáticos, es seguro que los tendrá con más frecuencia de lo que sería deseable).

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8.htm#TOP)  Inicio](http://www.zator.com/Hardware/H8.htm#TOP)

#### 8.1.2  Discos:  Estructura lógica

##### §1  Generalidades

Una vez conocidas las características físicas de los discos duros, acometeremos aquí una descripción de cómo están organizados los datos en su interior;  cómo es la estructura lógica que los alberga.  Empezaremos advirtiendo que en éste, como en otros aspectos de la informática, la estructura está organizada en niveles.  En este caso, la estructura de más alto nivel es lo que conocemos como **Sistema de Ficheros** ("File System") que será tratada con mayor detalle en el siguiente capítulo (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2a](http://www.zator.com/Hardware/H8_1_2a.htm)).  Antes debemos introducir algunos conceptos imprescindibles para manejarnos en el tema con un mínimo de vocabulario.

El uso y la experiencia ha consagrado distintos sistemas de ficheros en los distintos Sistemas Operativos y disposiciones hardware.  Pero en todo caso, su fin último es siempre el mismo: conseguir un mecanismo que proporcione la forma más rápida, segura y flexible de almacenar y recuperar los datos en los sistemas de disco.  En nuestra descripción seguiremos el orden que creemos más didáctico para el interesado que desee una visión general sobre el asunto, sin otro requisito previo que la necesaria dosis de curiosidad.

Adelantaremos que la forma de organización de los niveles altos, en especial el Sistema de Ficheros en su conjunto, es característico de cada SO particular.  Aunque las interioridades de su funcionamiento son propias de cada disco (dependen del fabricante) las interfaces de bajo nivel son universales.  Ya que las unidades físicas pueden ser montadas en equipos muy distintos y deben ser utilizadas con cualquier sistema operativo.

**Un poco de historia**:  la primera unidad de disco duro (almacenamiento magnético) que pudiera considerarse como tal, con capacidad de acceso aleatorio ("random access"), fue el famoso RAMAC. Un equipo de IBM que apareció comercialmente en Septiembre de 1956. El sistema, dotado con 50 platos de 24 pulgadas de diámetro (unos 60 cm) recubiertos de óxido de hierro, ocupaba el espacio de un refrigerador grande, y generaba tanto calor que disponía de un compresor de aire para refrigerar las cabezas de lectura/escritura.  A pesar de su aparatosidad, la capacidad era de solo 5 MB. El elevado número de platos y su desmesurado tamaño se debían a que su densidad de grabación era de 2.000 bits por pulgada cuadrada. Es decir, de unos 3 bits por mm2. A los estándares actuales (2006) de 100 Gigabits por pulgada cuadrada, podemos considerar que la distancias entre bits en las superficie de grabación eran interestelares.  De acuerdo con los expertos, también para los discos duros existe versión de la Ley de Moore, según la cual, la capacidad de las unidades se duplica aproximadamente cada 23 meses. Aunque en tecnología es muy difícil ejercer de adivino, todos los indicios apuntan a que el fin del disco duro giratorio está próximo. La disminución de precios en las memorias flash apunta en el sentido que los próximos discos serán de estado sólido (estáticos, http://www.zator.com/Hardware/images/Ico_hoja.gif  [8.2.1a](http://www.zator.com/Hardware/H8_2_1a.htm)).

En realidad, IBM no vendía la unidad, sino que la alquilaba con servicio de mantenimiento incluido -algo muy normal en la época- a razón de 35.000 Dólares anuales (el equivalente a 250.000 dólares actuales).

Puede obtener información adicional en "Happy Birthday to the Hard Drive" un artículo de Terri Wells aparecido en Developer Shed http://www.zator.com/Hardware/images/Internet3.gif[www.devhardware.com](http://www.devhardware.com/c/a/Opinions/Happy-Birthday-to-the-Hard-Drive/)

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/H8_1_2-F1.jpg |
| http://www.zator.com/Hardware/Images_esp/H8_1_2-F2.jpg |

##### §2  Pistas, sectores y clusters

Los partículas magnéticas de la superficie del disco se utilizan en grupos dispuestos en círculos concéntricos denominados **pistas** ("Tracks"), que se numeran desde el exterior al interior comenzando por la cero [[1](http://www.zator.com/Hardware/H8_1_2.htm#[1])].

En las unidades modernas se suele utilizar ambas superficies de cada **plato** ("Platter"), de los que puede haber varios en la misma unidad [[2](http://www.zator.com/Hardware/H8_1_2.htm#[2])].  Las pistas de todos los platos equidistantes del eje conforman un **cilindro** ("Cylinder") y las cabezas de lectura/escritura están construidas de forma que todas están situadas sobre el mismo cilindro.

Recordemos que existen dos mecanismos; uno mecánico y otro eléctrico, que permiten a las cabezas leer/escribir sobre cualquier punto: el movimiento ("Seeking") hace que puedan situarse rápidamente sobre cualquier cilindro.  Por su parte el mecanismo de conmutación hace que sea una u otra cabeza la que lee/escribe en cada momento.  Recordemos también que hay tantas pistas como número de cilindros x número de platos x 2, y que en los discos duros, la velocidad de rotación angular es constante [[3](http://www.zator.com/Hardware/H8_1_2.htm#[3])].  En las unidades actuales oscila entre 4.200, 15.000 rpm.

Además de las pistas concéntricas, la superficie del disco se supone dividida en sectores circulares.  La intersección de cada sector sobre las pistas se conoce como **sector**.  La capacidad de cada sector es de  512 bytes.  Naturalmente, el número de sectores configura la capacidad total de la unidad.  Esta puede variar desde un par de miles de sectores en un humilde disquete, a cientos de millones en una unidad de gran capacidad.

**Nota**:  Observe que al referirse a discos duros, un **sector** no se refiere a la cuña completa, sino al trozo de pista intersectado por los radios que delimitan lo que en geometría es el sector circular.  También que los los sectores tienen distinta longitud según su distancia al centro.   Circunstancia esta que tiene importantes consecuencias prácticas.

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/H8_1_2-F5.jpg |

Dado que las pistas exteriores son más largas que las interiores, y con objeto de no aumentar la densidad lineal de los datos, es práctica común en los discos modernos (no así en los disquetes) que el número de sectores sea menor en las pistas interiores que en las exteriores.  O lo que es lo mismo: que el valor **SPT** ("Sectors per track") no sea igual en todos los cilindros (algunos programas de diagnóstico que proporcionan el valor SPT de la unidad, se refieren -en su caso- a las pistas exteriores).  Esta técnica de grabación se conoce como zoned-bit recording y señala que no todas las pistas tienen igual número de sectores.

##### §3  Valores CHS

Hemos señalado (http://www.zator.com/Hardware/images/Ico_hoja.gif [6.2](http://www.zator.com/Hardware/H6_2.htm#Direccionamiento LBA)) que en las unidades modernas, la identificación de cada sector viene determinada por su número **n** dentro del total **N** de sectores disponibles en la unidad.  Lo que se conoce como direccionamiento **LBA** ("Logical Block Addressing).  Sin embargo, por razones históricas, es costumbre designarlos mediante tres valores (que se representan por sus iniciales inglesas): número de cilindro **C**;  número de cabeza **H** y número de sector dentro de la pista **S**.  Así que en la literatura especializada encontrará repetidamente el acrónimo **CHS** para referirse a esta terna de valores que corresponden con las características "lógicas" (que no físicas) de la unidad.

En la época de los primeros PCs, la designación **CHS** correspondía efectivamente con las características físicas de las unidades de disco.  Sin embargo, en las unidades modernas estos valores no guardan ninguna relación con la física del dispositivo.  Existe un algoritmo que traslada estos tres valores a los parámetros internos del sistema.  Por ejemplo.  El portátil en que escribo, dispone de una unidad Toshiba MK6025GAS con 117.210.240 sectores, lo que hace un total de 60.011.642.880 bytes (60.01 GB según el fabricante [[8](http://www.zator.com/Hardware/H8_1_2.htm#[8])]).  Sus especificaciones técnicas señalan que la unidad tiene dos platos y 4 cabezas a las que corresponden las siguientes "características lógicas":

Cabezas H = 16

Cilindros C = 16,383

Sectores por pista en la zona cero [[9](http://www.zator.com/Hardware/H8_1_2.htm#[9])]  S = 63

##### §4  Formateo de bajo nivel

El número de sectores y pistas viene determinado por el fabricante de la unidad y su organización se realiza en un proceso denominado **formateo de bajo nivel**, o formateo físico, que incluye ciertas operaciones de verificación para comprobar que las cabezas son capaces de leer/escribir sin error, cualquier patrón de bits en todos y cada uno de los bytes de las superficies.  En caso de haber alguno erróneo (lo que es relativamente frecuente), estos bytes malos se apuntan en una tabla interna de la propia unidad y no vuelven a utilizarse.

**Nota**:  Algunos sistemas de fichero.  Por ejemplo NTFS (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2a](http://www.zator.com/Hardware/H8_1_2a.htm#Sistemas Windows)) tienen la capacidad de realizar esta anotación en run-time ("Hot fixing"), de forma que si detectan un byte defectuoso, es anotado para que no vuelva a ser utilizado por la unidad.

Actualmente los discos vienen formateados de bajo nivel y probados, por lo que el formateo físico es ráramente necesario.  Hasta fechas relativamente recientes, era frecuente encontrar la tabla de sectores defectuosos anotada por el fabricante en la propia carátula de la unidad.  En cambio, antaño era imprescindible realizar este formateo sobre las unidades nuevas mediante herramientas software especiales.  Entonces era frecuente que el formateo de bajo nivel y comprobación de un disco Winchester de 5+5 MB tardara varias horas!! [[5](http://www.zator.com/Hardware/H8_1_2.htm#[5])].

**Nota**:  En épocas en que las unidades de disco eran terriblemente caras (un disco Winchester podía costar el equivalente a un automovil actual de tipo medio), si el disco empezaba a dar problemas de lectura/escritura, la terapia consistía en volver a realizar un formateo de bajo nivel y recomprobar el estado de la superficie (volver a pasar el "dkinit"). Si existían nuevos bits defectuosos, eran añadidos a la tabla de defectos y la unidad volvía a ser puesta en explotación cruzando los dedos.  En cambio, el costo de las unidades actuales hace que excepcionalmente haya que reparar o recuperar un disco [[6](http://www.zator.com/Hardware/H8_1_2.htm#[6])].  Es más, el mantenimiento actual es tipo proactivo, en el sentido de que se monitorizan determinados parámetros de funcionamiento y la unidad es desechada a la más mínima señal de que pueda presentar un fallo.

##### §5  Unidad de asignación

Los sectores constituyen la menor cantidad de información que las cabezas pueden leer/escribir cada vez, lo que significa que la unidad puede leer/escribir como mínimo la información contenida en un sector (que para el sistema estará libre u ocupado, pero nunca a medias).  A este respecto es importante señalar que el SO ráramente utiliza esta cantidad directamente.  Esto significaría, sobre todo en unidades grandes, tener la información demasiado fragmentada (contenedores unitarios demasiado pequeños).  Por esta razón, se utilizan múltiplos de esta cantidad en forma de agrupaciones de sectores contiguos denominadas **unidad de asignación** ("Cluster") [[4](http://www.zator.com/Hardware/H8_1_2.htm#[4])].

**Nota**:  Los sectores por cluster utilizados por Microsoft en sus sistemas de ficheros son:  1, 2, 4, 8, 16, 32, 64 y 128.

##### §6  Formateo de alto nivel

El tamaño del cluster (número de sectores), que conforma la unidad mínima de lectura/escritura,  depende del sistema de ficheros utilizado y de la capacidad de la unidad (puede ser distinto para cada disco).  Su determinación, junto con la creación de determinadas estructuras auxiliares, que comentaremos más adelante (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2a](http://www.zator.com/Hardware/H8_1_2a.htm#Datos y metadatos)), se realiza mediante el denominado **formateo de alto nivel**, o formateo lógico, que es realizado por una utilidad del SO.   En los sistemas ext2 (Linux) es la utilidad **mke2fs**; en los sistemas FAT (MS-DOS/Windows), es la utilidad **format.com**.  El cuadro adjunto muestra las respuestas obtenidas al ejecutar esta última sobre una unidad de 4 GBytes.

Comprobando el formato del disco  
Grabando los clústeres no válidos actuales  
Completado.  
Comprobando 4.001,45 MB  
Formato completado.  
Escribiendo la tabla de asignación de archivos  
Completado.  
Calculando espacio libre (esto puede tardar algunos minutos)...  
Completado  
  
¿Nombre del volumen? (11 caracteres, Entrar para ninguno)?  
  
4.187.631.616 bytes de espacio total en disco  
4.187.631.616 bytes disponibles en disco  
  
4.096 bytes en cada unidad de asignación.  
1.022.370 unidades de asignación libres en disco  
  
El número de serie del volumen es 304A-0FEF

**Nota**:  El caso de los disquetes es una excepción sobre lo comentado.  Aunque estas unidades pueden venir del fabricante con un formateo de alto nivel al estilo MS-DOS, **format.com** realiza en ellas los formateos de alto y bajo nivel simultáneamente.  Por supuesto, un disquete HD de 3.5" puede sufrir distintos formatos en una máquina DOS/Windows, en una Linux o en un Mac.  La tabla adjunta muestra la distribución de los disquetes formateados bajo MS-DOS.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Capacidad nominal | 360 KB. | 1.2 MB. | 720 KB. | 1.44 MB. | 2.88 MB. |
| Tamaño (diámetro) | 5.25" | 5.25" | 3.5" | 3.5" | 3.5" |
| Designación | **DD** | **HD** | **DD** | **HD** | **EHD** |
| Sectores por cluster | 2 | 1 | 2 | 1 | 2 |
| Pistas por cara | 40 | 80 | 80 | 80 | 80 |
| Sectores por pista | 9 | 15 | 9 | 18 | 36 |

|  |
| --- |
| http://www.zator.com/Hardware/Images_esp/H8_1_2-F3.gif  Fig. 3 |
| http://www.zator.com/Hardware/Images_esp/H8_1_2-F4.gif  Fig. 4 |

La figura 3 muestra el resultado después del formateo de alto nivel de un disquete 3.5" HD en MS-DOS/Windows (la unidad está totalmente vacía).  En este caso el cluster es de un sector y la capacidad total coincide con el producto: 2847 clusters x 512 bytes/cluster = 1.457.664 bytes.  Sin embargo, las unidades de asignación anunciadas no corresponden con el total teórico de la tabla anterior, que debería ser:  80 pistas/cara x 18 sectores/pista x 2 caras = 2880 clusters.  Los 33 clusters perdidos corresponden a distintas estructuras de bajo nivel, no utilizables directamente por el usuario, a las que tendremos ocasión de referirnos más adelante, que son creadas durante el formateo de alto nivel (los resultados mostrados en el cuadro se refieren a capacidades utilizables por el usuario).

La figura 4 muestra el resultado de la utilidad **scandisk** [[7](http://www.zator.com/Hardware/H8_1_2.htm#[7])] en una unidad de 4 GB. formateada con la misma utilidad que el disquete anterior (la unidad contiene algunos datos de usuario).  Puede verse que en esta ocasión el cluster es de 8 sectores (4.096 bytes) y que la capacidad total coincide con el producto del más de un millón de clusters existentes en la unidad por el tamaño de cada uno (4.187.623.424 = 4.096 x 1.022.369).  También en este caso los valores se refieren a capacidad útil, una vez deducido el espacio requerido por las estructuras del formato lógico.

En los sistemas Windows y MS-DOS, puede utilizar la utilidad **chkdsk.exe** seguida del nombre de la unidad **A:**; **C:**; **D:**; etc. para obtener información sobre sus discos (unidades lógicas) [[10](http://www.zator.com/Hardware/H8_1_2.htm#[10])].  Por ejemplo, en una unidad de 300 GB nominales, el formateo lógico con **format.com**, ha construido un Sistema de Ficheros FAT32 con las siguientes características (en el momento del análisis la unidad contiene algunos datos de usuario):

292.954.048 KB de espacio total en disco  
277.057.632 KB libres  
  
32.768 bytes en cada unidad de asignación  
9.154.814 total de unidades de asignación en el disco  
8.658.051 unidades de asignación libres en disco

Como puede verse, el cluster es enorme, 64 sectores, y proporcionado al tamaño de la unidad.  http://www.zator.com/Hardware/images/Bombilla_.gifEs significativo que en estas unidades muy grandes, el almacenamiento de gran número de ficheros pequeños (de menos de 32 KB) puede suponer un terrible desaprovechamiento del espacio.  Por lo que dependiendo de su uso, es recomendable dividirlas en varias unidades lógicas (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2c1](http://www.zator.com/Hardware/H8_1_2c1.htm#Particiones)).  Aunque la capacidad de esta unidad es unas 70 veces mayor que la del ejemplo anterior, solo tiene 8.9 veces más clusters que aquel.  En esta ocasión es un Seagate Barracuda ST3300831AS, que según especificaciones del fabricante, tiene 586,072,368 sectores en modo LBA. Es decir, 300,069,052,416 Bytes.  Puede verse que las estructuras necesarias para el formateo de alto nivel suponen un espacio considerable.  Aún considerando KBytes de 1024 bytes, el espacio total (utilizable) del disco resulta ser 84 MB menor que su capacidad teórica!!.  La diferencia corresponde al espacio ocupado por los metadatos (datos sobre los datos).

**Nota**:  En realidad, la elección del tamaño de la partición, que como hemos visto, influye en el tamaño del cluster y en el aprovechamiento del espacio, es una cuestión de equilibrio.  Tamaños de cluster pequeños aprovechan mejor el espacio, pero conducen a muchas unidades de asignación en la unidad, y por ende, a grandes estructuras de metadatos (por ejemplo tablas FAT http://www.zator.com/Hardware/images/Ico_hoja.gif[8.1.2a1](http://www.zator.com/Hardware/H8_1_2a1.htm))  Por contra, tamaños grandes de cluster tienden a desaprovechar el espacio, pero conducen a estructuras más reducidas.

##### §7  Principio de operación

Los discos comienzan a escribirse del exterior al interior, siendo los datos que primero se graban los que más rápido se acceden y más velocidad de transferencia poseen, ya que la velocidad lineal de las pistas exteriores es mayor que en las interiores.

Contra lo que podría parecer, el llenado no se produce comenzando por una superficie y continuando en la siguiente, sino por cilindros. De forma que una vez llenados los clusters del cilindro exterior, se continúa con el adyacente.  Este procedimiento tiende a minimizar los movimientos de cabeza ("seeking"), de forma que para leer una serie de sectores sucesivos, basta la conmutación de cabezas mientras estas se mantienen sobre el mismo cilindro (la "conmutación" es un proceso electrónico incomparablemente más rápido que el "seeking" que implica movimiento mecánico).

##### §8  Fragmentación

La escritura secuencial de ficheros no produce huecos en la estructura de datos, ya que cada nuevo fichero supone la utilización de una serie de clusters sucesivos después del último ocupado.  El problema comienza cuando se borran ficheros intermedios.  Entonces aparecen zonas intermedias de clusters desocupados que tienden a rellenarse la próxima vez que se escribe un fichero.  Pero entonces, los clusters que componen el nuevo fichero no son contiguos y las operaciones de lectura/escritura tienen que proceder a saltos, con el consiguiente perjuicio para el rendimiento del sistema.  El problema, conocido como **fragmentación externa**, afecta a unos sistemas de ficheros más que a otros, pero empieza a ser especialmente acusado cuando la unidad está casi llena (por encima del 80% de su capacidad).

La **fragmentación** tiende a ser especialmente acusada en unidades en las que se escriben y borran gran cantidad de ficheros o cuando estos cambian su tamaño.  El problema es similar al que ocurre con la memoria en las aplicaciones que la utilizan extensivamente.  Para evitar sus efectos indeseables, existen herramientas como el conocido (y denostado) defragmentador de Windows (**defrag.exe**) que reacomodan sistemáticamente los datos reuniendo los cluster dispersos de un mismo fichero.

El problema de la fragmentación externa, junto con algunas otras necesidades de los grandes sistemas, que deben funcionar ininterrumpidamente durante semanas o meses con grandísimas cargas de trabajo en condiciones críticas, ha obligado a algunos sistemas de ficheros a dotarse de ciertas habilidades especiales.  Por ejemplo, conocer rápidamente si existe suficiente espacio contiguo en algún sitio de la unidad para colocar un fichero, en lugar de instalarlo (seguramente fragmentado) empezando en el primer sitio libre.

Una extensión ("extent") es un conjunto grande de clusters contiguos en un fichero.  Se define mediante una terna de valores:  el desplazamiento inicial respecto al principio del fichero;  el cluster de comienzo y la longitud (número) de clusters contiguos.  El sistema de ficheros mantiene una tabla de extensiones que le permite encontrar rápidamente el espacio adecuado para un nuevo fichero.

Existe el malentendido generalizado, de que la fragmentación es un problema exclusivo de los Sistemas DOS/Windows que **no** afecta a los sistemas Unix/Linux. Cuando en realidad no es cierto, ya que todos los sistemas de ficheros conocidos padecen del mismo defecto cuando se utilizan en las mismas circunstancias.

Probablemente el malentendido tiene su origen en que en sistemas mono usuario, tales como DOS/Windows, o incluso un sistema Linux de escritorio, la utilización de recursos se debe a la acción de un solo usuario, y por lo que se refiere al disco, los accesos suelen ser secuenciales sobre el mismo fichero.  En estas condiciones, la fragmentación puede ser un elemento determinate en el rendimiento global del proceso.  Por contra, en los sistemas multiusuario, especialmente si están muy cargados (muchos usuarios solicitando recursos distintos al mismo tiempo), son las peticiones al sistema de ficheros las que son de naturaleza fragmentada.  En estas condiciones el disco tiene que atender secuencialmente peticiones de trozos de ficheros distintos, lo que supone una penalización del rendimiento con independencia de que tales ficheros sean contiguos o no (aunque evidentemente, el hecho de que sean discontinuos no facilita las cosas).

Los efectos negativos de esta forma de utilización, que podíamos denominar de **demanda fragmentada**, se ven atenuados por el hecho de que las unidades modernas disponen de la capacidad **NCQ** ("Native Command Queing"), que permite que un dispositivo reciba un conjunto de órdenes y las reordene para conseguir el máximo rendimiento con el mínimo movimiento de cabezas.  Otras características propias de sistemas avanzados son el cacheado de disco, la lectura adelantada y la escritura retrasada (http://www.zator.com/Hardware/images/Ico_hoja.gif [5.2](http://www.zator.com/Hardware/H5_2.htm#Caché de disco)).

Según los expertos, el argumento final es que, en un sistema multiusuario-multitaréa, la sobrecarga inherente a la fragmentación de ficheros es despreciable frente a la derivada de la fragmentación de la demanda.  Por lo que es muy raro que se requiera defragmentar las unidades lógicas.  Máxime si se utiliza el sistema de ficheros ext2/ext3, que son inherentemente más resistente a la fragmentación que los sistemas FAT o NTFS.  A pesar de todo, aunque raras, existen herramientas para defragmentar discos Linux.  En cualquier caso, la recomendación de los expertos para estas situaciones es no ocupar las unidades por encima del 80% de su capacidad máxima, y en caso necesario, copiar el contenido de la partición en una unidad externa, borrar su contenido, y volverlo a copiar en su localización original.

http://www.zator.com/Hardware/images/marca_.gif  Tema relacionado:  Fragmentación interna (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2a1](http://www.zator.com/Hardware/H8_1_2a1.htm#[1])).

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8_1_2.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H8_1_2.htm#TOP)

[1]  En este sentido, la organización de los discos es diferente de la adoptada en los dispositivos ópticos (CDs y DVDs), donde la distribución no adopta la forma de pistas concéntricas, sino de una espiral continua.

[2]  En la década de los 90 eran frecuentes unidades con gran números de platos.  Actualmente (2005) el aumento de la densidad de grabación ha permitido construir unidades más delgadas, por lo que son relativamente raras las unidades con más de 4.  Como dato anecdótico puedo deciros que conservo un Maxtor XT 2190.  Un "ladrillo" de nada menos que 8 platos y 15 cabezas (una cara se reservaba para uso de la propia unidad) con la espeluznante (de entonces) capacidad de 190 MB!!.

[3]  Esta es otra diferencia importante de los discos duros respecto a los dispositivos ópticos como CDs y DVDs, en los que la velocidad angular del plato es variable.  De esta forma se consigue que la velocidad lineal del láser de lectura sobre la pista se mantenga constante, tanto en las zonas cercanas al eje, como en las más alejadas (lo que no ocurriría de mantenerse la velocidad angular constante).

[4] La palabra cluster se utiliza también con otro significado:  La agrupación de varios equipos de cómputo, unidos por redes de alta velocidad, para realizar determinada tarea de forma cooperativa (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2a](http://www.zator.com/Hardware/H8_1_2a.htm#[2])).

[5]  La designación tiene su origen el modelo 3340, lanzado originariamente por IBM en 1973, en el que las cabezas "vuelan" sobre un colchón de aire sobre la superficie del disco (sistema que sigue utilizándose actualmente).  Estas unidades tenían dos discos de 30 MB colocados uno sobre otro, de forma que el superior era removible; podía cambiarse mediante una operación de carga superior ("Top loader").  Los discos tenían una carcasa plástica de protección en forma de gigantesca tartera, cuya tapa podía quitarse para introducir el disco en la unidad.  La designación 30-30 derivó en "Winchester", en recuerdo del famoso modelo del mismo nombre del conocido fabricante de armas.

[6]  En 1991 el precio de un subsistema de disco Data General de 24 Gigabytes (una capacidad monstruosa en aquellos días) era de 25.000 dólares americanos (de entonces); en 2005 he adquirido una unidad Seagate de 300 GB por 135 Euros.  Como consecuencia, las operaciones de recuperación de discos no están orientadas a volver a utilizar la unidad, sino a la recuperación de información en unidades averiadas de las que no existe copia de seguridad.

[7]  Los sistemas Windows disponen de dos versiones de este programa: **scandisk.exe** y **scandskw.exe**.  El primero es una versión MS-DOS; el segundo es una versión Windows con interfaz gráfica.  En ambos casos es una utilidad de chequeo de las estructuras creadas durante el formato de alto nivel; de la consistencia del sistema de ficheros, y de la idoneidad de la superficie útil para escribir datos.

[8]  La documentación del fabricante señala que para Toshiba, 1 MB equivale a 1.000.000 bytes y GB a 1.000.000.000 Bytes (lo que es de agradecer http://www.zator.com/Hardware/images/Ico_hoja.gif[E1.7.1](http://www.zator.com/Cpp/E1_7_1.htm#Unidades de medida)).

[9]  La zona exterior, que tiene más sectores.  La información del fabricante parece confirmar que el dispositivo utiliza la técnica de grabación "zoned-bit recording" antes mencionada [http://www.zator.com/Hardware/images/Ico_hojaFup.gif](http://www.zator.com/Hardware/H8_1_2.htm#Nota-2).

[10]  Los Sistemas MS-DOS tienen una utilidad de verificación de la integridad de los discos duros **chkdsk.exe**, que incluye información sobre el tamaño de la unidad, espacio libre, etc.  Las versiones MS-DOS contenidas en Windows 9x también disponen de ella, aunque recomiendan usar **scandisc.exe** para este menester.  Sin embargo, **chkdsk** puede seguir siendo útil para obtener información sobre la unidad.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8_1_2.htm#TOP)  Inicio](http://www.zator.com/Hardware/H8_1_2.htm#TOP)

#### 8.2  Discos externos

##### §1  Antecedentes

Seguramente no estamos inventando la pólvora si decimos que el mundo en que vivimos, se digitaliza a pasos agigantados. Gran parte de la humanidad dispone cada día de más cacharros ("gadgets") digitales. Desde teléfonos móviles a cámaras fotográficas, pasando por reproductores de música y un largo etcétera (ayer vi por primera vez un portarretratos digital [[1](http://www.zator.com/Hardware/H8_2.htm#[1])]). Esto sin contar con que el PC se vuelve cada día más "personal"; no solo existe un PC por cada empleado en cualquier oficina, empresa, o negocio. Incluso en el entorno doméstico son ya muchos los hogares que disponen de más de un ordenador, y la tendencia es que, a partir de cierta edad, cada miembro de la familia disponga del suyo propio (seguramente portátil). En realidad es una repetición de lo ocurrido con los teléfonos móviles.

La consecuencia es que la cantidad de información que generamos y almacenamos crece de forma paralela a la digitalización de nuestras vidas. No solo en las empresas, en las que desde siempre ha habido que archivar cartas, facturas, albaranes, expedientes etc. Almacenamiento que ahora se realiza en formato digital. También en el ámbito privado y doméstico, la cantidad de información que manejamos crece de forma imparable. Los estudiantes trabajan sobre libros y apuntes electrónicos, y casi todo el mundo archiva ya su correspondencia en formato digital (casi seguro en MS Word). También queremos conservar nuestros email, que a fin de cuentas representan un registro de nuestra vida; y por supuesto, queremos conservar nuestras fotos; nuestra música (el MP3 ha desbancado casi por completo al CD), e incluso las películas que bajamos de la Red, o los momentos singulares capturados con el móvil, que ya es capaz de grabar pequeñas secuencias de audio y video.

La consecuencia práctica de lo anterior es que periódicamente, los almacenamientos en línea (discos) de los equipos se quedan pequeños, en un proceso que podríamos calificar como "natural". Tradicionalmente los grandes sistemas empresariales no tenían gran problema en este sentido; los discos eran fácilmente ampliables, ya que se trataba de unidades externas que se alojaban en la misma sala que la unidad central. Por lo general los cables de conexión y alimentación se tendían por debajo de un falso suelo desmontable, que era y sigue siendo casi obligado en este tipo de instalaciones.

Sin embargo, en los ordenadores personales (PCs) el asunto no es tan fácil. El espacio interno es limitado, y aunque las unidades de disco pueden ser sustituidas (cambiadas por otras de más capacidad), generalmente es complicado instalar más de dos discos duros en una caja estándar. Sobre todo considerando que además debe montarse una unidad CD/DVD y quizás una disquetera.  Para complicarlo aún más, las carcasas modernas (barebones) tienden a ser cada vez más pequeñas, y los denominados "Home entertainment" o "Home media center" [[2](http://www.zator.com/Hardware/H8_2.htm#[2])], que comienzan a proliferar como un electrodoméstico más, son del mismo tamaño que un reproductor de video tradicional.  Si nos referimos a los portátiles, que son cada vez más frecuentes como ordenador personal, e incluso en las empresas, como opción para el personal móvil, el asunto de la ampliación del almacenamiento se presenta aún más problemático.

##### §2 La solución externa

Desde 2005 a esta parte, ha empezado a emerger con fuerza un abanico de soluciones, destinadas tanto al ámbito empresarial, como a la informática personal; del hogar, o de las PYMES [[3](http://www.zator.com/Hardware/H8_2.htm#[3])]. Consiste en la utilización de discos externos montados en carcasas ("housing") específicamente diseñadas para este propósito.  Puede afirmarse que sin excepción, todos los fabricantes, incluyendo los de discos duros, han presentado soluciones en este sentido.

Existen distintos tipos, soluciones y capacidades. Para empezar comenzaremos clasificándolos según el uso al que están destinados, aunque debemos advertir que dichos usos no son necesariamente excluyentes.  Por ejemplo, una unidad pensada primordialmente para transportar datos, puede ser eventualmente utilizada como unidad de back-up y viceversa.  La clasificación sería como sigue:

* **Unidades para transporte de datos**.  Son unidades de pequeño tamaño en cuyo diseño priva la movilidad y robustez.  Representan la evolución natural del primitivo disquete y han sustituido a este.
* **Unidades de back-up**.  Son dispositivos cuya misión es almacenar copias de seguridad. Las unidades suelen ser extraíbles e intercambiables.
* **Servidores de ficheros**.  En esta categoría se incluyen desde unidades externas para informática personal, a unidades SAN / NAS para redes (intranets) de pequeñas empresas, o departamentales.

Desde fechas relativamente recientes, las siglas SAN / NAS han irrumpido con fuerza en la tecnología actual.

**NAS** ("Network Attached Storage").  Sistema de almacenamiento enlazado (ligado) a redes. El concepto NAS fué creado en 1996 y desde su nacimiento ha sido una opción asequible y fácil de instalar. Las unidades NAS están unidas a la red a través de una LAN convencional, los dispositivos NAS se gestionan de forma independiente, por lo que resultan muy adecuados para su utilización a nivel departamental o de grupo de trabajo.

El sistema se basa en dispositivos de almacenamiento (discos) instalados en servidores (generalmente mediante conexiones SCSI o SATA). A su vez, los servidores están conectados a la red local mediante protocolos TCP/IP.  Este es precisamente uno de sus inconvenientes, el acceso a los datos solo puede hacerse a través del servidor NAS correspondiente; cuando cualquier nodo necesita acceder al servidor los datos deben viajar por la LAN, con la consiguiente penalización en el tráfico.

**SAN** ("Storage Area Network") es un nuevo concepto (1998) que trata de dar respuesta al volumen creciente de datos que han de ser almacenados en los actuales entornos de red. Creando una SAN los usuarios pueden separar el tráfico de almacenamiento del resto de las operaciones de red, y obtener conexiones directas entre los servidores y los dispositivos de almacenamiento.

Básicamente una **SAN** es una red especializada que permite accesos rápidos y fiables entre servidores y los recursos de almacenamiento independientes o externos. En una SAN (al contrario que en el paradigma NAS), los dispositivos de almacenamiento no son propiedad exclusiva de ningún servidor, sino que, por el contrario, se comparten entre todos los servidores de la red como si fueran recursos "peer". De la misma forma que una LAN permite conectar clientes y servidores, una SAN puede establecer comunicación entre servidores y dispositivos de almacenamiento, o entre servidores y entre dispositivos de almacenamiento. Contra lo que pudiera parecer, una SAN no precisa de una red específica. Puede utilizarse para este fin una subred virtual dentro de la infraestructura de una intranet compartida.

En el presente capítulo ofreceremos un breve repaso a las principales características de las distintas soluciones que ofrece la tecnología actual.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8_2.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H8_2.htm#TOP)

[1]  12-05-2006. Aunque conocía de su existencia por las revistas especializadas, me refiero a la exposición en un comercio de un dispositivo real.

[2]  Equipos que integran un PC de altas prestaciones, junto con todos los dispositivos de E/S necesarios para la grabación y reproducción de música y video. Representan una evolución del reproductor DVD tradicional, que a pasado a convertirse en un dispositivo polifacético orientado al entretenimiento. Aparte de los puertos E/S propios de cualquier PC estándar, incluyen un grabador/reproductor CD/DVD con posibilidad de grabación de video. Sintonizador de FM y capturadora de TDT (Televisón Digital Terrestre); conexión a Internet; tarjeta de sonido multicanal, y quizás ranuras para lectura de distintos tipos de tarjetas de memoria (para cámaras de fotos). Generalmente dotado de un mando a distancia y utilizando la pantalla del televisor como salida. A la fecha (2006) suelen montar MS Windows XP Media Center Edition 2005 como Sistema Operativo.

[3]  Acrónimo de "Pequeña y Mediana Empresa". Equivalente al SOHO ("Small Office/Home Office") en la literatura inglesa.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8_2.htm#TOP)  Inicio](http://www.zator.com/Hardware/H8_2.htm#TOP)

#### 8.1.2c  El Sector de Arranque

##### §1  Sector de arranque

De todos los sectores de una unidad de disco, el primero de la primera cabeza del primer cilincro (**CHS**  0,0,1), tiene una importancia y significado especial.  Es el sitio al que se dirige la BIOS cuando busca si existe en el sistema un dispositivo cargable (http://www.zator.com/Hardware/images/Ico_hoja.gif [4.3](http://www.zator.com/Hardware/H4_3.htm#Carga del Sistema)).  Por esta razón se denomina **sector de arranque** **MBR** ("Master boot record") o bloque maestro de carga ("Master boot block").  Sus 512 bytes contienen tres bloques con información sobre la arquitectura física y lógica del disco: el  **Código maestro de carga** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H8_1_2c.htm#MBC); la  **Tabla de particiones** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H8_1_2c.htm#Tabla maestra de particiones), **y la Firma** [http://www.zator.com/Hardware/images/Ico_hojaFdwn.gif](http://www.zator.com/Hardware/H8_1_2c.htm#Firma del sector de carga).

**Nota**:  Debido a que los disquetes no pueden contener particiones, son una excepción a esta regla y su primer sector, denominado sector de carga del volumen, tiene una estructura ligeramente distinta (ver sector de carga http://www.zator.com/Hardware/images/Ico_hoja.gif[8.1.2c2](http://www.zator.com/Hardware/H8_1_2c2.htm)).

##### §2  Código maestro de carga MBC

Si el disco es "bootable", los primeros 446 bytes del **MBR** (sector de arranque), están ocupados por un pequeño trozo de código denominado código maestro de carga **MBC**  ("Master Boot Code") o cargador inicial (**bootstrap loader**), que es cargado por la BIOS para comenzar el proceso de carga. El bootstrap loader repasa la tabla maestra de particiones (ver a continuación) buscando una partición activa. En caso de encontrarla, busca su sector inicial (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2c2](http://www.zator.com/Hardware/H8_1_2c2.htm#VBS)), carga su código en memoria, y le transfiere el control.  Dicho código es ya capaz de cargar y ejecutar cualquier otro programa situado en cualquier partición del disco.  Que a su vez inicializará directamente el SO, o tal vez una utilidad conocida como gestor de arranque, que permite elegir entre distintas alternativas.

Como vemos, es un proceso en cadena:  el bootstrap loader es cargado en memoria por un programa situado en la BIOS, y a su vez es capaz de continuar la carga del Sistema Operativo.

##### §2  Tabla maestra de particiones MPT

A continuación del **MBC**, se sitúa la tabla maestra de particiones **MPT** ("Master Partition Table").  Está constituida por cuatro trozos de 16 bytes (4 entradas) que contienen información sobre las particiones definidas en la unidad.

**Nota**:  las particiones a que nos referimos son las denominadas **particiones primarias** o **volúmenes** [[2](http://www.zator.com/Hardware/H8_1_2c.htm#[2])].  Hay que recordar que solo hay sitio para cuatro, por lo que un disco duro solo puede contener cuatro particiones primarias.  Cualquier otra que pueda establecerse, se denomina **partición secundaria** o volumen lógico y debe estar contenida en alguna de las primarias.

Los desplazamientos de cada una de las 4 entradas son respectivamente 1BEh (446); 1CEh (462);  1DEh (478) y 1EEh (494). Cada entrada contiene la siguiente información [[3](http://www.zator.com/Hardware/H8_1_2c.htm#[3])] (**T** = tamaño del campo en bytes):

|  |  |
| --- | --- |
| **T** | **Descripción** |
| 1 | Estado de la partición (00h = Inactiva;  80h = Activa) |
| 1 | Principio de la partición (cabeza). |
| 2 | Principio de la partición (Cilindro/Sector en forma codificada).  La codificación, bastante enrevesada. Se justifica porque en el momento de su diseño, la conservación de espacio era una preocupación fundamental en el mundo informático (la misma que condujo al famoso episodio del Y2K) y porque existen dos datos empaquetados en una palabra de 16 bits, que a su vez están en forma invertida; "Back-words" o "Little endian" (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.6a](http://www.zator.com/Cpp/E2_2_6a.htm)).  De los 16 bits disponibles, 6 se reservan para el sector y 10 para el cilindro.  Lo que conduce a un máximo de 64 = 26 sectores, y 1024 = 210 cilindros.   El esquema de distribución de bits es el siguiente:   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | bits 7-0 del cilindro | | | | | | | | bits 9-8  cilindro | | bits 5-0 del sector | | | | | |   Estos valores, junto a las 256 = 28 posibilidades par las cabezas señaladas en el byte anterior, originarían posteriormente numerosos inconvenientes a la hora de manejar discos muy grandes  (http://www.zator.com/Hardware/images/Ico_hoja.gif[6.2](http://www.zator.com/Hardware/H6_2.htm)). |
| 1 | Tipo de partición **ID**.  Contiene detalles sobre el formato de la partición.  Por ejemplo, si es FAT-12, FAT-16 o FAT-32; si será accedida utilizando direccionamiento CHS tradicional; ECHS, o LBA (http://www.zator.com/Hardware/images/Ico_hoja.gif [6.2](http://www.zator.com/Hardware/H6_2.htm)).   |  |  | | --- | --- | | **ID** | **Descripción** | | **01h** | Partición primaria FAT12 o volumen lógico (volumen de menos de 32,680 sectores) | | **04h** | Partición primaria FAT16 o volumen lógico (32,680–65,535 sectores o 16 MB–33 MB) | | **05h** | Partición DOS Extendida | | **06h** | Partición FAT16 BIGDOS o volumen lógico (33 MB–4 GB) | | 07h | Partición NTFS o volumen lógico ("Installable File System") | | 0Bh | Partición FAT32 o volumen lógico (hasta 2048 GB) | | 0Ch | Partición FAT32 o volumen lógico usando extensiones de la INT 13h BIOS | | 0Eh | Partición FAT16 BIGDOS o volumen lógico usando extensiones de la INT 13h BIOS | | 0Fh | Partición DOS extendida usando extensiones de la INT 13h BIOS | | 12h | Partición EISA | | 42h | Dynamic disk volume | | 86h | Legacy FT FAT16 disk | | 87h | Legacy FT NTFS disk | | 8Bh | Legacy FT volume formateado con FAT32 | | 8Ch | Legacy FT formateado con FAT32 usando extensiones de la INT 13h BIOS |   **Nota**:  MS-DOS solo puede acceder a las particiones o volúmenes señalados en negrita. |
| 1 | Final de la partición (cabeza). |
| 2 | Final de la partición (Cilindro/Sector en forma codificada). |
| 4 | Número de sectores entre el MBR y el primer sector de la partición. |
| 4 | Número de sectores en la partición. |

Para que un disco "maestro" (cuyo MBR pueda ser accedido por el proceso de carga de la BIOS) sea "bootable", se precisa que alguna de las entradas de la tabla contenga un 80h en el bit de estado, señalando cual es **partición activa**.  http://www.zator.com/Hardware/images/Bombilla_.gifRecuerde que esta indicación es necesaria incluso si solo existe una partición y que en caso de haber varias, solo una puede estar activa!.

##### §3  Firma del sector de carga

Los dos últimos bytes del sector de arranque (**MBR**) contienen dos caracteres (55h, AAh). que son denominados firma del sector de carga ("Boot record signature")

##### §4  Ejemplo

A continuación se muestra el volcado ASCII del sector de arranque **MBR** de un disco, utilizando la utilidad **debug** del DOS (http://www.zator.com/Hardware/images/Ico_hoja.gif [DEBUG](http://www.zator.com/Hardware/H1_7_1.htm)).

Se trata de una unidad C...

|  |
| --- |
|  |

##### §5  Resumen

Una vez realizado el formateo físico de la unidad (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2](http://www.zator.com/Hardware/H8_1_2.htm#Formateo de bajo nivel)), la **MPT** es creada mediante un programa especial; **fdisk.exe** en los Sistemas DOS/Windows y **/sbin/fdisk** en Linux (http://www.zator.com/Hardware/images/Ico_hoja.gif [8.1.2c3](http://www.zator.com/Hardware/H8_1_2c3.htm)), que se encarga de realizar las "particiones" del disco. En cambio, el código del bootstrap loader es instalado en su caso por el programa de instalación del Sistema Operativo.

Un disco pude no ser "bootable" por carecer de cargador inicial en su **MBR**, o la señal "activa" en alguna de sus particiones. Pero para ser utilizable debe contener al menos una partición, y por consiguiente una entrada en su **MPT**.  En el primer caso, la BIOS es incapaz de continuar el proceso de carga, resultando un mensaje de error del tipo: "No Boot device available o quizás en las BIOS antiguas: "NO ROM BASIC - System Halted.  Sin embargo, una vez cargado el SO, si su MBR contiene una entrada en su **MPT**, el volumen correspondiente podrá ser utilizado sin ningún problema como área de almacenamiento.

**Nota**:  La razón de este último mensaje es que los primitivos PCs IBM tenían un BASIC pregrabado en la ROM de la BIOS, de forma que si no podía encontrarse un dispositivo de donde cargar el sistema, la BIOS exhibía este mensaje de error del BASIC.

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8_1_2c.htm#TOP)  Inicio.](http://www.zator.com/Hardware/H8_1_2c.htm#TOP)

[2]  En algunos Sistemas las particiones son denominadas "Disk slices" y las entradas en la tabla, etiquetas de disco ("disklabels").

[3]  Los parámetros de más de un byte están almacenados en forma de palabras invertidas ("Back-words") o "Little endian" (http://www.zator.com/Hardware/images/Ico_hoja.gif [E2.2.6a](http://www.zator.com/Cpp/E2_2_6a.htm)), y como es usual, los desplazamientos empiezan por cero desde el comienzo del sector.  El primer byte tiene desplazamiento cero.  El último 511 (1FFh).

[[http://www.zator.com/Hardware/images/arrow_btt.gif](http://www.zator.com/Hardware/H8_1_2c.htm#TOP)  Inicio](http://www.zator.com/Hardware/H8_1_2c.htm#TOP)