Sistemas de numeración y codificación de la información

Como dijimos en el documento anterior, los sistemas informáticos utilizan internamente el sistema de numeración *binario*, que está basado únicamente en dos dígitos (0 y 1). También hemos comentado que las personas utilizamos el sistema de numeración *decimal* que, como su nombre indica, se basa en el uso de diez dígitos (del 0 al 9).

En definitiva, un sistema de numeración es un conjunto de reglas y símbolos que nos permiten representar valores numéricos.

Esto significa que un determinado valor se representará de forma diferente según el sistema de numeración que estemos empleando. O al contrario, la misma representación puede equivaler a diferentes valores cuando usamos sistemas de numeración distintos.

Veamos un ejemplo:

Sistema de numeración	
Decimal	Binario
2	10
10	1010
220	11011100

Cuando mezclamos valores representados en diferentes sistemas de numeración, podemos indicar el sistema al que pertenece cada número añadiendo a su derecha un subíndice con el número de símbolos que utiliza dicho sistema. Por ejemplo:

$$220_{(10} = 11011100_{(2}$$

Tanto el sistema de numeración *decimal* como el *binario* (y el resto de los que usaremos en este capítulo) son posicionales. Esto significa que cada dígito representa un determinado valor que depende del propio dígito y de la posición que éste ocupa en el número. Para explicar esta idea, tomemos este número: $327_{(10}$

El dígito 7 representa 7 unidades porque se encuentra en la primera posición de la derecha. Sin embargo, el dígito 2 representa 2 decenas, es decir, 20 unidades, porque lo encontramos en la segunda posición, comenzando por la derecha. Del mismo modo, el dígito 3 representa 3 centenas (300 unidades), porque aparece en la tercera posición, comenzando por la derecha.

También existen sistemas de numeración antiguos que no eran posicionales, como el romano o el egipcio.

Todos los sistemas de numeración posicionales se rigen por el *Teorema fundamental de la numeración* que establece que el valor de un número viene dado por la suma de cada uno de sus dígitos multiplicados por su base elevada a la posición que ocupa dicho dígito, teniendo en cuenta que las unidades ocupan la posición 0.

$$N = \sum_{i=-d}^{i=n} dígito_i \times base^i$$

Siguiendo esta idea, podríamos representar el número anterior, usando su *expresión posicional*, de la siguiente forma:

$$327_{(10} = 3 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

Convertir valores *binarios* a *decimal*

En Informática el *Teorema fundamental de la numeración* tiene aprovechamiento inmediato, ya que nos permite convertir un número representado en *binario* a *decimal* de una forma muy sencilla.

El truco está en escribir en *decimal* la expresión posicional del número en *binario*. De esta forma, al resolverla, obtendremos el resultado en *decimal*.

Veamos un ejemplo:

$$11011100_{(2} = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$= 128 + 64 + 16 + 8 + 4 = 220_{(10}$$

Convertir valores *decimales* a *binarios*

Cuando necesitemos realizar la conversión en sentido contrario, podemos aplicar el método de las divisiones sucesivas.

Básicamente, este método consiste en realizar una división entera (es decir, sin obtener decimales) del número en *decimal* entre dos (la base a la que vamos a convertir). Una vez concluida la división, tomamos el cociente y volvemos a dividirlo entre dos. Así continuamos hasta que obtengamos un cociente igual a cero. En este punto, tomamos los restos de todas las divisiones en orden inverso a como han ido apareciendo (es decir, el último resto será el primero).

Como ejemplo, veamos la misma conversión del punto anterior, pero en el sentido contrario:

Dividendo	Divisor	Cociente	Resto
220	2	110	0
110	2	55	0
55	2	27	1
27	2	13	1
13	2	6	1
6	2	3	0
3	2	1	1
1	2	0	1

11011100

Los sistemas de numeración *octal* y *hexadecimal*

Una de las dificultades que plantea el trabajo con números *binarios* es la cantidad de dígitos que empleamos para representar cantidades relativamente pequeñas (en el ejemplo anterior hemos necesitado 8 dígitos *binarios* para representar el mismo valor que en *decimal* representamos con tres).

Por este motivo, en lugar de trabajar directamente en *binario*, es frecuente utilizar otros sistemas de numeración que utilicen un mayor número de símbolos.

Podríamos pensar que el sistema de numeración perfecto para este trabajo es el *decimal*, porque es al que estamos más acostumbrados. Sin embargo, los sistemas de numeración cuya base es una potencia de 2 tienen una ventaja añadida: la conversión entre ellos es automática.

Este es el motivo por el que suelen utilizarse los sistemas de numeración *octal* y *hexadecimal*.

El sistema de numeración *octal*

Como indica su nombre, el sistema de numeración *octal* utiliza ocho dígitos (0 a 7) para representar valores.

Como ya hemos dicho más arriba, la ventaja que representa ***octal*** sobre ***decimal*** es que utiliza una base que es potencia de 2 ($2^3 = 8$). En resumidas cuentas, esta característica nos va a permitir que cada tres caracteres en *binario* se puedan representar, directamente, con un carácter en *octal*.

Para comprenderlo, observemos la siguiente tabla que nos muestra valores en *octal* y en *binario*:

Binario	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

De esta forma, para convertir un número escrito en *binario* al sistema *octal*, bastaría con tomar los bits agrupados de tres en tres, comenzando por la derecha y completando con los ceros necesarios a la izquierda:

$$011011100_{(2}$$

A continuación, buscamos cada grupo de tres dígitos en la tabla anterior y los sustituimos por su dígito correspondiente en la tabla:

$$\begin{array}{ccc} 011 & 011 & 100_{(2} \\ 3 & 3 & 4_{\ (8} \end{array}$$

Por lo tanto, el número en *binario* $11011100_{(2}$ corresponde con el número $334_{(8}$ en *octal*.

Resulta evidente que el paso de *octal* a *binario* es igual de sencillo.

Convertir valores de *octal* a *decimal*

La forma más sencilla de convertir un valor en *octal* a *decimal* es aplicar el *Teorema fundamental de la numeración* como ya hemos hecho con el *binario*. Como la expresión posicional del número en *octal* la escribimos en *decimal*, al resolverla, obtendremos el resultado en *decimal*.

Veamos un ejemplo:

$$334_{(8} = 3\times8^2+3\times8^1+4\times8^0$$
$$=192 + 24 + 4 = 220_{(10}$$

Convertir valores de *decimal* a *octal*

Para realizar la conversión de un valor *decimal* a *octal*, podemos aplicar el método de las divisiones sucesivas que ya usamos con el *sistema binario*. Es decir, realizaremos la división entera (sin obtener decimales) del número en *decimal* entre ocho (la base a la que vamos a convertir). Una vez concluida la división, tomamos el cociente y volvemos a dividirlo entre ocho. Este proceso continuará hasta que obtengamos un cociente igual a cero.

En este punto, tomamos los restos de todas las divisiones en orden inverso a como han ido apareciendo (es decir, el último resto será el primero).

Como ejemplo, veamos la misma conversión del punto anterior, pero en el sentido contrario:

Dividendo	Divisor	Cociente	Resto
220	8	27	4
27	8	3	3
3	8	0	3

334

El sistema de numeración *hexadecimal*

El sistema de numeración *hexadecimal* utiliza dieciséis dígitos para representar valores. Como nosotros sólo utilizamos los dígitos del 0 al 9, y no son suficientes, en este caso se añaden las letras A a F para representar los dígitos que nos faltan. Por lo tanto, los dígitos que utilizamos para representar valores en *hexadecimal* son:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Ya hemos comentado más arriba que, como pasaba con el sistema *octal*, la ventaja que representa *hexadecimal* sobre *de cimal* es que utiliza una base que es potencia de 2 ($2^4 = 16$). De forma similar a lo que ocurría en *octal*, esta característica nos va a permitir que cada cuatro caracteres en *binario* se puedan representar, directamente, con un carácter en *hexadecimal*.

Para comprobarlo, vamos a partir de la tabla que nos muestra valores en *hexadecimal* y en *binario*:

Binario	Octal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

De esta forma, para convertir un número escrito en *binario* al sistema *hexadecimal*, bastaría con tomar los bits agrupados de cuatro en cuatro, comenzando por la derecha y completando con los ceros que sean necesarios a la izquierda:

$$11011100_{(2}$$

A continuación, buscamos cada grupo de cuatro dígitos en la tabla anterior y los sustituimos por su dígito correspondiente en la tabla:

$$\begin{array}{ll} 1101 & 1100_{(2} \\ \mathrm{D} & \mathrm{C}_{(16} \end{array}$$

Por lo tanto, el número en *binario* $11011100_{(2}$ corresponde con el número $DC_{(16}$ en *hexadecimal*.

Como puedes suponer, el paso de *hexadecimal* a *binario* es igual de sencillo.

Convertir valores de *hexadecimal* a *decimal*

Como en el caso del *binario* y el *octal*, la forma más sencilla de convertir un valor en *hexadecimal* a *decimal* es aplicar el *Teorema fundamental de la numeración*. Como antes, la expresión posicional del número en *hexadecimal* la escribimos en *decimal*. De este modo, al resolverla, obtendremos el resultado en *decimal*.

Para entender el siguiente ejemplo, debemos tener en cuenta el valor, en *decimal*, de los últimos 6 dígitos *hexadecimal*es:

Hexadecimal	Decimal
A	10
B	11
C	12
D	13
E	14
F	15

Veamos el ejemplo:

$$\begin{aligned} DC_{(16} &= 13\times16^{1}+12\times16^{0} \\ &= 208 + 12 = 220_{(10} \end{aligned}$$

Convertir valores de *decimal* a *hexadecimal*

Para realizar la conversión de un valor *decimal* a *hexadecimal*, aplicaremos de nuevo el método de las divisiones sucesivas que hemos utilizado con anterioridad. Es decir, realizaremos la división entera (sin obtener decimales) del número en *decimal* entre dieciséis (la base a la que vamos a convertir). Una vez concluida la división, tomamos el cociente y volvemos a dividirlo entre dieciséis. Este proceso continuará hasta que obtengamos un cociente igual a cero.

Llegado este momento, tomamos los restos de todas las divisiones en orden inverso a como han ido apareciendo (es decir, el último resto será el primero), acordándonos de sustituir aquellos valores que sean mayores que 9 por su dígito correspondiente en *hexadecimal*.

Como ejemplo, veamos la misma conversión del punto anterior, pero en el sentido contrario:

Dividendo	Divisor	Cociente	Resto
220	16	13	12 (C)
13	16	0	13 (D)