

Contenido

Sistemas de Numeración en Informática	3
Introducción	3
Bits, bytes y términos de medida	3
Binario	4
Hexadecimal	5
Bibliografía	6

Copyright©2016.

Autor: Yunges, Bárbara Carina

¡Copia este texto!

Los textos que componen este trabajo se publican bajo formas de licenciamiento que permiten la copia, la redistribución y la realización de obras derivadas, siempre y cuando éstas se distribuyan bajo las mismas licencias libres y se cite la fuente. El copyright de los textos individuales corresponde a los respectivos autores.

Este trabajo está licenciado bajo un esquema Creative Commons Atribución CompartirIgual (CC-BY-SA) 4.0 Internacional. <<http://creativecommons.org/licenses/by-sa/4.0/deed.es>>`_



Sistemas de Numeración en Informática

Introducción

Las computadoras son dispositivos electromecánicos contruidos por conmutadores electrónicos. En los niveles más bajos de cálculo éstas reaccionan sólo a impulsos eléctricos. Estos impulsos son interpretados por la computadora como estados ON u OFF.

El 1 representa el estado ON y el 0, el estado OFF. Esos unos y ceros se conocen como dígitos binarios o *bits*.

El código ASCII (código normalizado americano para el intercambio de información), es el más utilizado para la representación de datos alfanuméricos en una computadora. Utiliza dígitos binarios para representar los símbolos pulsados en el teclado.

Bits, bytes y términos de medida

Los bits son dígitos binarios, unos o ceros. En una computadora, están representados por conmutadores on/off o por la ausencia o presencia de cargas eléctricas, pulsos de luz u ondas de radio.

Por ejemplo: Un 0 binario puede estar representado por 0 voltios de electricidad; un 1 binario puede estar representado por +5 voltios de electricidad.

Las computadoras están diseñadas para utilizar colecciones de 8 bits. Ésta se denomina: *byte*. En una computadora, 1 byte respresenta una sola ubicación de almacenamiento direccionable. Estas ubicaciones de almacenamiento representan un valor o un solo caracter de datos, como por ejemplo podemos citar al código ASCII.

¿Por qué ASCII utiliza números binarios de 8 bits o 1 byte?.

Debido a que debemos dar cabida a todos los caracteres de nuestro alfabeto. Si utilizamos un sólo bit solo tenemos posibilidad de representar dos valores: 1 y 0. Si utilizamos dos bits las combinaciones aumentan, como se muestra en la siguiente tabla:

Num. de bits	Combinaciones posibles
1	2: 0,1
2	4: 00,01,10,11
3	8: 000,001,010,011,100 101,110,111

La cantidad necesaria para poder abarcar todos nuestros caracteres sería de 8 bits, 1 byte, que nos da 256 combinaciones posibles.

En la siguiente imagen se muestra como se forman algunos de los caracteres del código ASCII:

Binário	Caracter	Binário	Caracter	Binário	Caracter
0010 0001	!	0100 0001	A	0110 0001	a
0010 0010	"	0100 0010	B	0110 0010	b
0010 0011	#	0100 0011	C	0110 0011	c
0010 0100	\$	0100 0100	D	0110 0100	d
0010 0101	%	0100 0101	E	0110 0101	e
0010 0110	&	0100 0110	F	0110 0110	f
0010 0111	'	0100 0111	G	0110 0111	g
0010 1000	(0100 1000	H	0110 1000	h
0010 1001)	0100 1001	I	0110 1001	i

A partir del byte, las unidades de capacidad de información, en el sistema internacional, se miden en múltiplos, siendo un kilobyte (Kb) 1024 bytes, tal y como se refleja en la tabla siguiente:

Medida	Equivalencia
Byte	8 bits
Kilobyte	1024 bytes
Megabyte	1024 kilobytes
Gigabyte	1024 megabytes
Terabyte	1024 gigabytes
Petabyte	1024 terabytes
Exabyte	1024 petabytes
Zettabyte	1024 exabytes
Yottabyte	1024 zettabytes

Binario

El sistema binario o en base 2, sólo utiliza dos símbolos (0 y 1), en lugar de los diez símbolos utilizados en el decimal, o en Base 10. La posición, o lugar de cada dígito respresenta el número 2 (o número base) elevado a una potencia (exponente).

- Número de símbolos: Dos
- Símbolos: 0,1
- Exponente Base: $2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$

Binario a Decimal

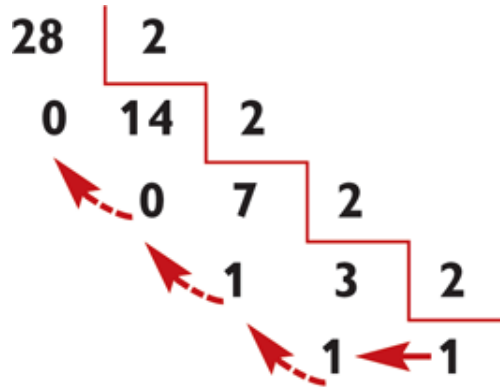
Ejemplo: tenemos el número: 00010110 y lo queremos pasar a decimal.

Para hacer esto multiplicamos el número por el exponente base y sumamos de esta forma:

$$00010110 = (0 * 2^7) + (0 * 2^6) + (0 * 2^5) + (1 * 2^4) + (0 * 2^3) + (1 * 2^2) + (1 * 2^1) + (0 * 2^0) = 0 + 0 + 0 + 16 + 0 + 4 + 2 + 0 = 22$$

Decimal a Binario

Para hacer la conversión de decimal a binario, hay que ir dividiendo el número decimal por dos y anotar el resto (un 0 si el resultado de la división es par y un 1 si es impar). La lista de ceros y unos leídos de abajo a arriba es el resultado. Ejemplo:



$$28 = 11100_2$$

Hexadecimal

El código hexadecimal sólo sirve para representar información a nivel usuario. La computadora no utiliza el mismo, pero facilita el trabajo del programador.

El código hexadecimal convierte cada grupo de 4 bits en un solo carácter. De esta forma, un byte (ocho 1 y 0) se representará por 2 caracteres. A cada grupo de 4 bits se le denomina nibble, de forma que 2 nibbles forman un byte. Si contamos todas las combinaciones con 4 bits veremos que hay 16 posibilidades. El código hexadecimal establece un solo carácter a cada uno de ellos, empezando por la combinación mas baja posible (0000) a la cual le otorga el 0, siguiendo con los números en orden hasta el 9 y finalizando con letras en orden alfabético, como muestra la siguiente tabla:

Código Binario	Hexadecimal
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

Es importante entender que al convertir cada cuatro 1 y 0 en un solo carácter la cantidad de texto se divide entre cuatro y al programador le es mas sencillo escribir y leer la información, aunque el ordenador sigue trabajando con la codificación binaria.

Bibliografía

- Libro Guia del primera año CCNA 1 y 2 - Academia de Networking de Cisco Systems
- <http://www.educoteca.com/uploads/4/6/2/3/46232277/binario.pdf>
- <https://informaticaencaude.wordpress.com/2011/10/04/conversion-de-decimal-a-binario/>
- http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_contenidos_2c.htm