

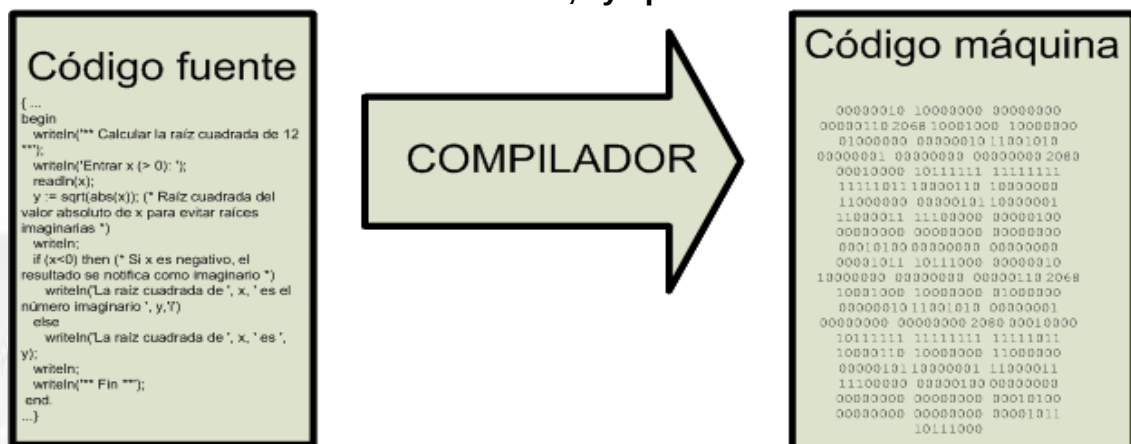
Definición de Código Máquina

En el mundo de la Programación, existen dos tipos de lenguajes que se emplean durante todo el proceso, y que significan quizá el punto de partida y el desenlace de un camino obligatorio que debe recorrerse para poder crear una Aplicación, un Programa o simplemente modificar o reparar los errores que hayan sido originados accidentalmente por otro programador.



En primer lugar, tenemos un Lenguaje de Programación en la que un usuario se encarga de redactar los procedimientos y las reglas específicas bajo unas normas predeterminadas, siendo considerado como Lenguaje de Alto Nivel, en el cual se elabora el Código Fuente, con un conjunto de ordenes y palabras que son entendidas por los usuarios, permitiendo su lectura, análisis y redacción (además de la corrección de errores)

Pero el ordenador no es posible por sí solo de leer esto, y es por ello que debemos recurrir a una herramienta llamada Compilador, que se encarga de realizar revisiones del mismo, y poder realizar una traducción



al Lenguaje de Bajo Nivel, que es equivalente al conocido como Código Objeto, o bien también llamado como Código Máquina.

En Programación, se define como el Código Máquina a aquel que es el proveniente de la tarea de compilación efectuada directamente sobre el Código Fuente, con el que se obtiene posteriormente el Código de Bytes (en inglés, Bytecode) que es la conglomeración de distintos archivos que forman parte de ejecutables para que el ordenador pueda hacer uso del código anteriormente programado.

De este modo, tenemos que tener en cuenta que con un programa Editor se realiza el primer paso, la elaboración del Código Fuente, que continúa con la Compilación del mismo mediante un Programa Objeto que se encarga de traducir el mismo hacia el paso previo al Código Binario (es decir, la transmisión o no-transmisión de

energía eléctrica) que aprovechan los dispositivos pertenecientes al ordenador.

Para que el Código Objeto pueda ser ejecutado finalmente, es necesario utilizar una aplicación llamada Enlazador (también conocido por su equivalente en inglés, Linker) que se encarga de crear todos los archivos necesarios para su ejecución, incluyendo además las Bibliotecas que permiten asignar las funciones a los distintos dispositivos del sistema, solicitar distintos Servicios al sistema operativo o básicamente hacer uso de todo el Hardware que sea necesario para la ejecución de dicho Programa o Aplicación que hemos creado.

Una instrucción en lenguaje máquina puede, por ejemplo, representarse de la siguiente forma:

011011001010010011110110

Como se puede observar, estas instrucciones serán fáciles de leer por la computadora y difíciles por un programador, y viceversa, esta razón hace difícil escribir programas en código o lenguaje a máquina y requiere buscar otro lenguaje para comunicarse con la computadora, pero que sea más fácil de escribir y leer por el programador. Para evitar la tediosa tarea de escribir programas en lenguaje máquina, se han diseñado otros lenguajes de programación que facilitan la escritura y posterior ejecución de los programas.

De binario a decimal

En sistema decimal, las cifras que componen un número son las cantidades que están multiplicando a las distintas potencias de diez (10, 100, 1000, 10000, etc.)

Por ejemplo, $745 = 7 \cdot 100 + 4 \cdot 10 + 5 \cdot 1$
O lo que es lo mismo: $745 = 7 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$

En el sistema binario, las cifras que componen el número multiplican a las potencias de dos (1, 2, 4, 8, 16,)

$2^0=1, 2^1=2, 2^2=4, 2^3=8, 2^4=16, 2^5=32, 2^6=64, \dots$

Por ejemplo, para pasar a binario un número decimal, empezamos por la derecha y vamos multiplicando cada cifra por las sucesivas potencias de 2, avanzando hacia la izquierda:

$$10110_2 = 0 \cdot 1 + 1 \cdot 2 + 1 \cdot 4 + 0 \cdot 8 + 1 \cdot 16 = 2 + 4 + 16 = 22_{10}$$
$$110_2 = 0 \cdot 1 + 1 \cdot 2 + 1 \cdot 4 = 2 + 4 = 6_{10}$$

Método Simplificado

64	32	16	8	4	2	1
↑	↑	↑	↑	↑	↑	↑
1	0	1	0	0	1	0
=						
64	+	16	+		2	=
82						

De decimal a binario

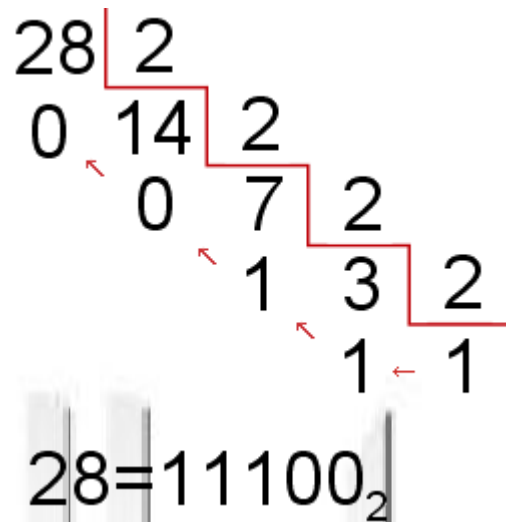
Para hacer la conversión de decimal a binario, hay que ir dividiendo el número decimal entre dos y anotar en una columna a la derecha el resto (un 0 si el resultado de la división es par y un 1 si es impar).

La lista de ceros y unos leídos de abajo a arriba es el resultado.

Ejemplo: vamos a pasar a binario 79_{10}

79 1 (impar). Dividimos entre dos:
39 1 (impar). Dividimos entre dos:
19 1 (impar). Dividimos entre dos:
9 1 (impar). Dividimos entre dos:
4 0 (par). Dividimos entre dos:
2 0 (par). Dividimos entre dos:
1 1 (impar).

Por tanto, $79_{10} = 10011112$



Código ASCII

Código ASCII (parcialmente representado) que se emplea para representar los caracteres alfanuméricos, es decir, letras, números y signos. Este código comprende los números decimales del 0 al 255. Del 0 al 31 corresponde a instrucciones. El número 32 corresponde a la orden de ejecutar espacios entre palabras cuando oprimimos la barra espaciadora en el teclado. Del 33 al 127 corresponde a los caracteres alfanuméricos más utilizados. A partir del número 128 aparecen otras letras y algunos signos que generalmente no aparecen en el teclado del ordenador.

Si quieres escribir cualesquiera de los caracteres alfanuméricos incluidos entre el número 33 y el 255, sólo tienes que abrir el procesador de textos y activar el teclado numérico. Si ese teclado no se encuentra activado,

sólo tienes que oprimir la tecla "Bloq Num" en el propio teclado (cuando está<activado se reconoce porque se enciende el primer LED, situado encima de esa tecla, que aparece con<el nombre "N/Lock"). Seguidamente se oprime la tecla "Alt" y se teclea, simultáneamente, sin soltarla, el<número decimal correspondiente a la letra, número o signo del Código ASCII que queremos obtener. A<continuación soltamos la tecla "Alt" y el carácter aparecerá escrito en el procesador.

En el código binario, el número "0" corresponde igualmente al "0" y el "255" al "1111 1111". Cada uno de<los caracteres alfanuméricos del Código ASCII equivale a un Byte de información, aunque el número<binario correspondiente al decimal no ocupe ocho cifras.

El código ASCII comprende sólo hasta el número decimal 255, porque a partir de ahí, el número 256 en<binario pasa a ser 1 0000 0000, sobrepasando los ocho dígitos requeridos para completar un byte de información.

Decimal	Signif.	Código Binario		Decimal	Signif.	Código Binario
32	Espacio	10 0000		95	_	101 1111
33	!	10 0001		96	`	110 0000
34	"	10 0010		97	a	110 0001
35	#	10 0011		98	b	110 0010
36	\$	10 0100		99	c	110 0011
37	%	10 0101		100	d	110 0100
38	&	10 0110		101	e	110 0101
39	'	10 0111		102	f	110 0110
40	(	10 1000		103	g	110 0111
41	)	10 1001		104	h	110 1000
42	*	10 1010		105	i	110 1001
43	+	10 1011		106	j	110 1010
44	,	10 1100		107	k	110 1011
45	-	10 1101		108	l	110 1100
46	.	10 1110		109	m	110 1101
47	/	10 1111		110	n	110 1110
48	0	11 0000		111	o	110 1111
49	1	11 0001		112	p	111 0000
50	2	11 0010		113	q	111 0001
51	3	11 0011		114	r	111 0010
52	4	11 0100		115	s	111 0011
53	5	11 0101		116	t	111 0100
54	6	11 0110		117	u	111 0101

55	7	11 0111		118	v	111 0110
56	8	11 1000		119	w	111 0111
57	9	11 1001		120	x	111 1000
58	:	11 1010		121	y	111 1001
59	;	11 1011		122	z	111 1010
60	<	11 1100		123	{	111 1011
61	=	11 1101		124		111 1100
62	>	11 1110		125	!	111 1101
63	?	11 1111		126	~	111 1101
64	@	100 0000		127	¡	111 1110
65	A	100 0001		128	Ç	1000 0000
66	B	100 0010		130	é	1000 0010
67	C	100 0011		144	É	1001 0000
68	D	100 0100		157	Ø	1001 1101
69	E	100 0101		160	á	1010 0000
70	F	100 0110		161	í	1010 0001
71	G	100 0111		162	ó	1010 0010
72	H	100 1000		163	ú	1010 0011
73	I	100 1001		164	ñ	1010 0100
74	J	100 1010		165	Ñ	1010 0101
75	K	100 1011		166	ª	1010 0110
76	L	100 1100		167	º	1010 0111
77	M	100 1101		168	¿	1010 1000
78	N	100 1110		169	®	1010 1001
79	O	100 1111		171	½	1010 1010
80	P	101 0000		172	¼	1010 1100
81	Q	101 0001		173	ì	1010 1101
82	R	101 0010		181	Á	1011 0101
83	S	101 0011		184	©	1011 1000
84	T	101 0100		214	Í	1101 0110
85	U	101 0101		224	Ó	1110 0000
86	V	101 0110		225	ß	1110 0001
87	W	101 0111		230	μ	1110 0110
88	X	101 1000		233	Ú	1110 1001
89	Y	101 1001		241	±	1111 0001

90	Z	101 1010		243	$\frac{3}{4}$	1111 0011
91	[	101 1011		246	$\div$	1111 0110
92	\	101 1100		248	$^{\circ}$	1111 1000
93	]	101 1101		252	3	1111 1100
94	^	101 1110		253	2	1111 1101