Lógica Computacional

Sistemas de numeración

Sistemas de numeración

- Sistema de numeración: Conjunto de reglas que permiten la representación de cantidades utilizando símbolos.
- Estos símbolos representan valores...
- ...que combinados determinan el número a representar.

Diego Codevilla - CC-BY-SA 4.0

Sistema posicional

- El sistema decimal, es posicional, o sea:

- Cada símbolo tiene un valor **absoluto**. Ej. 0 1 2 3 4 5 6 7 8 9
- Y un valor **relativo** que depende de la **posición que ocupa en la cadena de símbolos**:
- Ej. Si un 3 está en la segunda columna (30), vale 3×10^1 y en la cuarta columna? $3000 = 3 \times 10^3$
- Esta tendencia se repite infinitamente, siempre multiplicando por 10 más cada posición que se agregue.

Diego Codevilla - CC-BY-SA 4.0

Sistema posicional

- Ejemplos:

- $10 = 1 \times 10 + 0 \times 1$

- $11 = 1 \times 10 + 1 \times 1$

- $54 = 5 \times 10 + 4 \times 1$

- $96 = 9 \times 10 + 6 \times 1$

Diego Codevilla - CC-BY-SA 4.0

Sistema posicional

- $129 = 1 \times 10^2 + 2 \times 10^1 + 9 \times 10^0 = 1 \times 100 + 2 \times 10 + 9$
- $554 = 5 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 = 5 \times 100 + 5 \times 10 + 4$
- $986 = 9 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 = 9 \times 100 + 8 \times 10 + 6 \times 1$
- $25432 = 2 \times 10^4 + 5 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$

Diego Codevilla - CC-BY-SA 4.0

Sistema posicional

- Un número con 1 dígitos representará 10 cantidades diferentes.
- Un número con 2 dígitos representará 100 cantidades diferentes (10^2)
- Un número con 3 dígitos representará 1000 = 1000 (o sea 10^3) cantidades diferentes.

En general un número de **n** dígitos representará b^n números diferentes en base b.

Ej. Con 3 dígitos en base 16 se puede representar 16^3 = 4096 valores.

Diego Codevilla - CC-BY-SA 4.0

Binario

- Las circuitos de las computadoras funcionan con dos niveles: son circuitos digitales.
- Por eso solo pueden usar dos símbolos.
- Usan sistema binario natural (u otros códigos binarios...)
- En sistema binario se usan dos símbolos: 0 y 1
- Con estos dos símbolos se representan todas las cantidades...

Y qué cantidad representa 1_2 ? ¿Y 101_2?

¿Y este? 10_2

Diego Codevilla - CC-BY-SA 4.0

Binario

En el sistema binario, se cumplen las mismas reglas que en el sistema decimal, pero teniendo en cuenta que la base es 2, por lo que se modifican los pesos de cada columna:

Ej.: pesos de 3 dígitos en decimal 10^2 10^1 10^0 => 100 10 1

y en binario: $2^2 + 2^1 + 2^0$ => 4 2 1

Diego Codevilla - CC-BY-SA 4.0

Binario

Contando en binario:

Si empezamos a contar en binario arrancamos con el 0, luego con el 1 y nos quedamos sin valores. Entonces, al igual que en cualquier otro sistema, debemos agregar un nuevo dígito, formando así el 10. Este valor es el 2 en decimal, por lo tanto, el 1 en la posición uno (recordemos contar la posición cero) duplica su valor relativo, tomando valor 2.

Si seguimos contando tenemos el 11, 3 en decimal, y ya nos quedamos sin combinaciones posibles con dos dígitos, pasando a tener que formar un número de tres cifras, el 100. Este número representa al 4 en decimal, por lo que nuevamente podemos decir que el valor relativo del 1 en la segunda posición duplica su valor respecto al de la primera.
Esto se repetirá hasta el infinito, si seguimos agregando dígitos, los pesos se irán duplicando, en la tercera posición valdrá 8, en la cuarta 16, en la quinta 32, etc.

Diego Codevilla - CC-BY-SA 4.0

Binario

- Cada una de las posiciones de los dígitos duplica el peso siguiendo la siguiente progresión; 1, 2, 4, 8, 16, 32, 64, 128, etc…

- Podemos decir entonces que el peso está dado por la fórmula 2^n donde n es la posición del bit y el 2 viene dado por la base del sistema binario.

- Es análogo al sistema decimal visto, donde la base es 10.

Diego Codevilla - CC-BY-SA 4.0

Binario - Bit, Byte, Nibble

Bits. Bytes. Nibble:

- El **bit** es la mínima unidad de información en el sistema binario. Es un dígito binario, y puede tomar dos valores, **0** y **1**

- **Un Byte** es una unidad de información digital. Tiene una longitud de 8 bits que puede tomar valores entre **00000000** y **11111111**, (que corresponden al 0 y al 255 en decimal)

- Un nibble es cada una de las mitades de un byte. Ej. el número binario 10100011 está compuesto por dos nibbles: 1010 y 0011

Diego Codevilla - CC-BY-SA 4.0

Binario - Pesos, pasaje a decimal

Tomemos como ejemplo un Byte (8 bits) y analicemos los pesos de cada una de las posiciones:

peso	128	64	32	16	8	4	2	1
bit	7	6	5	4	3	2	1	0

¿Cómo pasar de binario a decimal?

La base es 2...

$$101011_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$= 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1$$

$$= 32 + 0 + 8 + 0 + 2 + 1 = 43$$

Diego Codevilla - CC-BY-SA 4.0

Binario - Pasaje a decimal

1	1	0	0	1	1	1	0

$1 \times 2^7 = 128$ $1 \times 2^6 = 64$ $0 \times 2^5 = 0$ $0 \times 2^4 = 0$ $1 \times 2^3 = 8$ $1 \times 2^2 = 4$ $1 \times 2^1 = 2$ $0 \times 2^0 = 0$

$128 + 64 + 0 + 0 + 8 + 4 + 2 + 0 =$

206

Se puede decir que, como sólo vamos a multiplicar por **0** ó **1**, simplemente sumaremos los pesos en cuyas posiciones haya un 1.

Diego Codevilla - CC-BY-SA 4.0

Binario - Pasaje a decimal

1	1	0	1	0	1	0	0

128 + 64 + 0 + 16 + 0 + 4 + 0 + 0 =

212

0	1	0	0	0	0	1	1

0 + 64 + 0 + 0 + 0 + 0 + 2 + 1 =

67

Diego Codevilla - CC-BY-SA 4.0

Binario - Pasaje a decimal

1	0	1	0	1	1	0	0

128 + 0 + 32 + 0 + 8 + 4 + 0 + 0 =

172

1	1	1	1	0	0	0	0

128 + 64 + 32 + 16 + 0 + 0 + 0 + 0 =

240

Diego Codevilla - CC-BY-SA 4.0

Binario - De decimal a binario

Pasemos de decimal a binario:

Existen varias maneras de pasaje de números entre sistemas pero la idea de este resumen es buscar la más rápida.

Nos basaremos en la premisa de que el pasaje de binario a decimal lo realizamos sumando los pesos, por lo que este pasaje lo haremos en forma inversa, restando. Si restamos los pesos y ubicamos un 1 en la posición del peso restado estaremos creando el número en binario.

Veamos cómo sería:

Partamos de un número de longitud igual o menor a un Byte, ej: 233. En este caso, sabemos que por ser menor a 255, no supera un Byte. Lo que tenemos que hacer es ir restando los pesos, del mayor al menor y ubicando un 1 en cada peso que hemos podido restar, hasta llegar a cero.
Empezamos con 128. 233 es mayor que 128 entonces lo podemos restar:

233 - 128 = 105
Restamos y agregamos un 1 en la posición restada.

Diego Codevilla - CC-BY-SA 4.0

Binario - De decimal a binario

Pasaje de decimal a binario - Ejemplo: 204_{10}

El mayor peso que le podemos restar a 204 es 128 => 204 - 128 = 76

El mayor peso que se le puede restar a 76 es 64 => 76 - 64 = 12

A 12 le podemos restar 8 => 12 - 8 = 4

Y a 4 le podemos restar 4 => 4 - 4 = 0 -> fin de la operación.



El número 204_{10} expresado en binario queda: $1\ 1\ 0\ 0\ 1\ 1\ 0\ 0_2$

Diego Codevilla - CC-BY-SA 4.0

Binario - De decimal a binario

Pasaje de decimal a binario - Ejemplo: 985_{10}

El mayor peso que le podemos restar a 985 es 512 => 985 - 512 = 473

473 - 256 = 217

217 - 128 = 89

89 - 64 = 25

25 - 16 = 9

9 - 8 = 1

1 - 1 = 0 -> fin

Queda:

512	256	128	64	32	16	8	4	2	1
1	1	1	1	0	1	1	0	0	1_2

Diego Codevilla - CC-BY-SA 4.0

Binario - De decimal a binario

240_{10}:

240 - 128 = 112
112 - 64 = 48
48 - 32 = 16
16 - 16 = 0

11110000_2

110_{10}:

10 - 64 = 46
46 - 32 = 14
14 - 8 = 6
6 - 4 = 2
2 - 2 = 0

01101110_2

19_{10}:

19 - 16 = 3
3 - 2 = 1
1 - 1 = 0

00010011_2

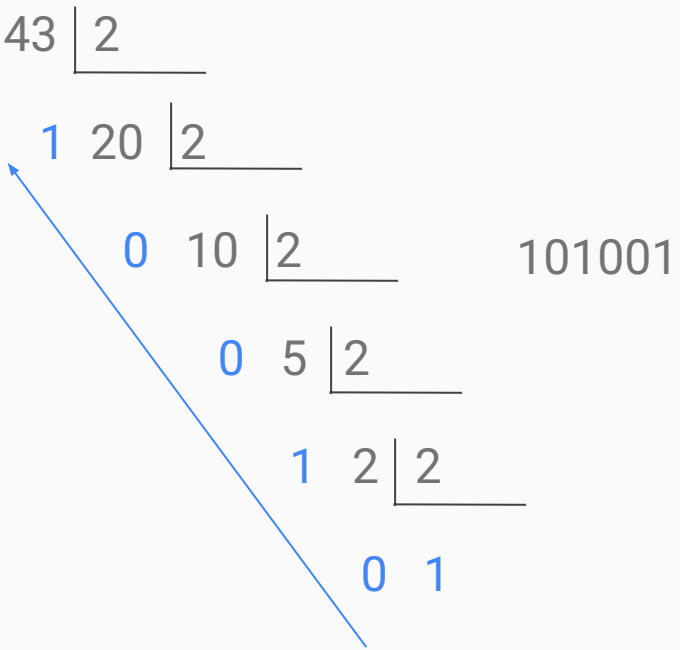
Diego Codevilla - CC-BY-SA 4.0

Binario - De decimal a binario

Otro método:

Dividimos por la base: 2

(Sirve para pasar de decimal a cualquier base. Ej. para pasar a hexadecimal se divide por 16)



Diego Codevilla - CC-BY-SA 4.0

Hexadecimal

- Este sistema numérico tiene base 16. Se usan 16 símbolos para representar las cantidades.

- Los símbolos son: del 0 al 9 y de la A hasta la F.

  Ej. $A45F_{16}$ 67_{16} EFB_{16}

Diego Codevilla - CC-BY-SA 4.0

Tabla Decimal - Hexa - Binario

Decimal	Hexa	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Decimal	Hexa	Binario
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Diego Codevilla - CC-BY-SA 4.0

Hexadecimal - Pasaje a decimal

- Para pasar a decimal podemos usar el método ya visto:



Ej. $5E3_{16} = 5 \cdot 16^2 + E \cdot 16^1 + 3 \cdot 16^0 = 5 \cdot 256 + 14 \cdot 16 + 3 \cdot 1 = 1507$



Ej. $A45F_{16} = 10 \cdot 16^3 + 4 \cdot 16^2 + 5 \cdot 16^1 + 15 \cdot 16^0 =$

$10 \cdot 4096 + 4 \cdot 256 + 5 \cdot 16 + 15 \cdot 1 = 42079$

Ej.: $A3C = A \cdot 16^2 + 3 \cdot 16^1 + C \cdot 16^0 = 10 \cdot 256 + 3 \cdot 16 + 12 \cdot 1 = 2620$

Diego Codevilla - CC-BY-SA 4.0

Hexadecimal - Pasaje a decimal

Ej. $B5_{16}$ en decimal será $B \cdot 16^1 + 5 \cdot 16^0 = 11 \cdot 16 + 5 \cdot 1 = 181$

y 181 en binario será: 1 0 1 1 0 1 0 1

Ej. $3C_{16}$ en decimal = 60, que en binario es 0 0 1 1 1 1 0 0

En estos dos ejemplos se observa que cada dígito hexa coincide con cuatro bits de la representación binaria...

Diego Codevilla - CC-BY-SA 4.0

Hexadecimal

- Se puede observar que cada dígito en hexadecimal se puede representar con 4 bits

- Esto no es casual y el uso de este sistema es por ser una “compresión” del sistema binario.

- Cada nibble puede representarse con un dígito hexadecimal. Y un byte puede ser representado por dos dígitos en hexadecimal.

Diego Codevilla - CC-BY-SA 4.0

Hexadecimal - a binario

De **hexadecimal a binario**, se hace “mentalmente” escribiendo cada dígito hexa como su representación en binario:

AB = 10101011, ya que A = 1010 y B=1011

7F = 01111111, ya que 7 = 0111 y F=1111

9D = 10011101

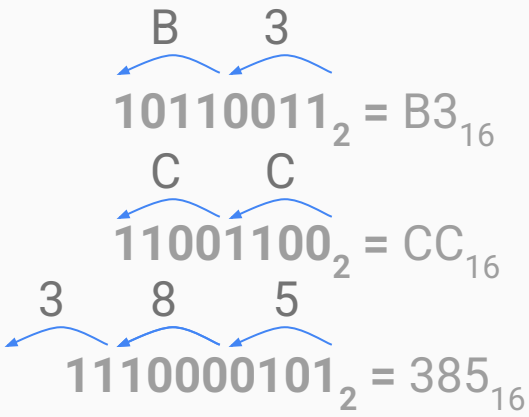
99 = 10011001

E2 = 11100010

Diego Codevilla - CC-BY-SA 4.0

Hexadecimal - de binario a Hexa

Para pasar de **Binario a Hexadecimal** también lo hacemos “mentalmente”, dividiendo el número binario en grupos de cuatro (desde la derecha) y escribiendo el dígito hexa correspondiente a cada grupo:



$00110110_2 = ????_{16}$

$101010_2 = ????_{16}$

$001111001010_2 = ????_{16}$

Diego Codevilla - CC-BY-SA 4.0

Ejercitación

Pasar a binario y hexadecimal los siguientes números decimales:

a. 15
b. 78
c. 67
d. 10
e. 16
f. 193
g. 222
h. 101
i. 1973
j. 2023

Diego Codevilla - CC-BY-SA 4.0

Ejercitación

Pasar a decimal y binario los siguientes números hexadecimales:

a. 4
b. C
c. 21
d. 3A
e. 1011
f. F5
g. 6B
h. 19D
i. 55
j. 6A5

Diego Codevilla - CC-BY-SA 4.0

Ejercitación

Pasar a decimal y hexadecimal los siguientes números binarios:

a. 110
b. 111
c. 101
d. 1010
e. 10110
f. 10011
g. 10110110
h. 11110000
i. 11000011
j. 1010101101

Diego Codevilla - CC-BY-SA 4.0