UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL MAR DEL PLATA
SISTEMAS DE PROCESAMIENTO DE DATOS
1er Año – 1er Cuatrimestre
PROFESORES: L. Chiessa - R. Soto - E. Monaco - G. Gimenez - V. Tomich

Sistemas Numéricos

Un sistema numérico son un conjunto de símbolos y reglas que se utilizan para representar datos numéricos o cantidades. Se caracterizan por su base que indican el número de símbolos distinto que utiliza y además es el coeficiente que determina cual es el valor de cada símbolo dependiendo de la posición que ocupe. Estas cantidades se caracterizan por tener dígitos enteros y fraccionarios.
Si a_j indica cualquier dígito de la cifra, b la base del sistema de numeración y además de esto la cantidad de dígitos enteros y fraccionarios son n y k respectivamente, entonces el número representado en cualquier base se puede expresar de la siguiente forma:

$$N_b = [a_{n-1}.a_{n-2}.a_{n-3}.........a_3.a_2.a_1.a_0,a_{-1}.a_{-2}.a_{-3}.......a_{-k}]_b$$

Donde: **j** = {n-1, n-2,.........2, 1, 0,-1, -2, ..., -k} y **n + k** indica la cantidad de dígitos de la cifra.

Por ejemplo, el número 31221, 32_4 en base cuatro tiene: **n=5** y **k=2** con la parte entera: $\mathbf{a_{n-1}}=\mathbf{a_4}=3$; $\mathbf{a_3}=1$; $\mathbf{a_2}=2$; $\mathbf{a_1}=2$; $\mathbf{a_0}=1$ y parte fraccionaria $\mathbf{a_{-1}}=3$; $\mathbf{a_{-2}}=2$

SISTEMA DECIMAL: Este es el sistema que manejamos cotidianamente, está formado por diez símbolos {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} por lo tanto la base del sistema es diez (10).

SISTEMA BINARIO: Es el sistema que utiliza internamente el hardware de las computadoras actuales, se basa en la representación de cantidades utilizando los dígitos 1 y 0. Por tanto su base es 2 (número de dígitos del sistema). Cada dígito de un número en este sistema se denomina bit (contracción de **bi**nary digi**t**). Se puede utilizar con nombre propio determinados conjuntos de dígitos en binario. Cuatro bits se denominan **cuaterno** (ejemplo: 1001), ocho bits **octeto** o **byte** (ejemplo: 10010110), al conjunto de 1024 bytes se le llama **Kilobyte** o simplemente **K,** 1024 Kilobytes forman un **megabyte** y 1024 megabytes se denominan **Gigabytes.**

SISTEMA OCTAL: El sistema numérico octal utiliza ocho símbolos o dígitos para representar cantidades y cifras numéricas. Los dígitos son: {0, 1, 2, 3, 4, 5, 6, 7}; la base de éste es ocho (8) y es un sistema que se puede convertir directamente en binario como se verá más adelante.

SISTEMA HEXADECIMAL: El sistema numérico hexadecimal utiliza dieciséis dígitos y letras para representar cantidades y cifras numéricas. Los símbolos son: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}; la base del sistema es dieciséis (16). También se puede convertir directamente en binario como se verá más adelante. En la tabla 1.1 se muestran los primeros veintiuno números decimales con su respectiva equivalencia binaria, octal y hexadecimal.

DECIMAL	BINARIO	OCTAL	HEXADECIMAL
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Tabla 1.1. Equivalencia entre sistemas de los primeros veintiuno números decimales.

CONVERSIÓN ENTRE LOS SISTEMAS NUMÉRICOS

CONVERSIÓN DECIMAL-BINARIO: Los métodos más conocidos son:

Divisiones sucesivas entre 2: Consiste en dividir sucesivamente el número decimal y los cocientes que se van obteniendo entre 2, hasta que una de las divisiones se haga 0. La unión de todos los restos obtenidos escritos en orden inverso, nos proporcionan el número inicial expresado en el sistema binario.

Ejemplo 1:

```
10 | 2
0    5 | 2
     1   2 | 2
         0   1 | 2
             1   0
```

$10_{(10)}=1010_{(2)}$

Ejemplo 2:

```
50 | 2
0    25 | 2
     1    12 | 2
          0    6 | 2
               1   3 |
                         10(10)=1010(2)
                       | 2
                   1     1 | 2
                         1   0
```

Multiplicación sucesiva por 2: Se utiliza para convertir una fracción decimal a binario, consiste en multiplicar dicha fracción por 2, obteniendo en la parte entera del resultado el primero de los dígitos binarios de la fracción binaria que buscamos. A continuación, repetimos el mismo proceso con la parte fraccionaria del resultado anterior, obteniendo en la parte entera del nuevo resultado el segundo de los dígitos buscados. Iteramos sucesivamente de esta forma, hasta que desaparezca la parte fraccionaria o hasta que tengamos los suficientes dígitos binarios que nos permitan no sobrepasar un determinado error.

Ejemplo: Convertir la fracción decimal 0.0828125 en fracciones binarias

0.828125	x	2	=	**1**.656250
0.656250	x	2	=	**1**.31250
0.31250	x	2	=	**0**.6250
0.6250	x	2	=	**1**.250
0.250	x	2	=	**0**.50
0.50	x	2	=	**1**.0

$0.828125_{10}= 0.110101_2$

Métodos de las restas sucesivas de las potencias de 2: Consiste en tomar el número a convertir y buscar la potencia de 2 más grande que se pueda restar de dicho número, tomando como nuevo número para seguir el proceso el resultado de la resta. Se repiten las mismas operaciones hasta que el número resultante en una de las restas es 0 o inferior al error que deseamos cometer en la conversión. El número binario resultante será un uno (1) en las posiciones correspondientes a las potencias restadas y un cero (0) en las que no se han podido restar.

Ejemplo:
Convertir el número decimal 1994 a binario.

Posición	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valor	1024	512	256	128	64	32	16	8	4	2	1
Digito	1	1	1	1	1	0	0	1	0	1	0

1994	-	1024	=	970
970	-	512	=	458
458	-	256	=	202
202	-	128	=	74
74	-	64	=	10
10	-	8	=	2

$1994_{10}=11111001010_2$

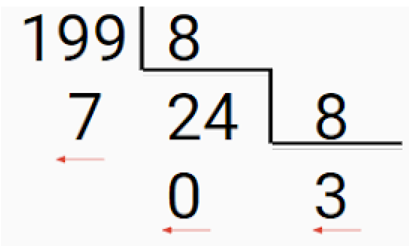
CONVERSIÓN DE BINARIO A DECIMAL: El método consiste en reescribir él número binario en posición vertical de tal forma que la parte de la derecha quede en la zona superior y la parte izquierda quede en la zona inferior. Se repetirá el siguiente proceso para cada uno de los dígitos comenzados por el inferior: Se coloca en orden descendente la potencia de 2 desde el cero hasta n, donde el mismo el tamaño del número binario, el siguiente ejemplo ilustra de la siguiente manera. Utilizando el teorema fundamental de la numeración tenemos que 1001.1es igual a:

$$1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} = 9.5_{(10)}$$

CONVERSIÓN DECIMAL – OCTAL: Consiste en dividir un número y sus sucesivos cocientes obtenidos por ocho hasta llegar a una división cuyo cociente sea 0. El número Octal buscado es el compuesto por todos los restos obtenidos escritos en orden inverso a su obtención.

Ejemplo:

$199_{(10)}=3710_{(8)}$



CONVERSIÓN DE UNA FRACCIÓN DECIMAL A UNA OCTAL: Se toma la fracción decimal y se multiplica por 8, obteniendo en la parte entera del resultado el primer dígito de la fracción octal resultante y se repite el proceso con

la parte decimal del resultado para obtener el segundo dígito y sucesivos. El proceso termina cuando desaparece la parte fraccionaria del resultado o dicha parte fraccionaria es inferior al error máximo que deseamos obtener.
Ejemplo:

0.140625*8 = 1.125
0.125*8 = 1.0

$0.140625_{(10)}=0.11_{(8)}$

CONVERSIÓN OCTAL A DECIMAL: Existen varios métodos siendo el más generalizado el indicado por el TFN (Teorema fundamental de la numeración) que hace la conversión de forma directa por medio de la formula. **Ej. :** utilizando el teorema fundamental de la numeración tenemos que 4701 es igual a:

$$4*8^3+7*8^2+0*8^1+1*8^0=2497_{(10)}$$

CONVERSIÓN DECIMAL – HEXADECIMAL: Se divide el numero decimal y los cocientes sucesivos por 16 hasta obtener un cociente igual a 0. El número hexadecimal buscado será compuesto por todos los restos obtenidos en orden inverso a su obtención.
Ejemplo:

1000	**16**	
40	**62**	**16**
8	**14**	**3**

$1000_{(10)}=3E8_{(16)}$

CONVERSIÓN DE UNA FRACCIÓN DECIMAL A HEXADECIMAL: a la fracción decimal se multiplica por 16, obteniendo en la parte entera del resultado el primer dígito de la fracción hexadecimal buscada, y se repite el proceso con la parte fraccionaria de este resultado. El proceso se acaba cuando la parte fraccionaria desaparece o hemos obtenido un número de dígitos que nos permita no sobrepasar el máximo error que deseemos obtener.

Ejemplo: Pasar a hexadecimal la fracción decimal 0.06640625

0.06640625*16 = 1.0625
0.0625*16 = 1.0

$0.06640625_{(10)}=0.11_{(16}$

CONVERSIÓN HEXADECIMAL- DECIMAL: el método más utilizado es el TFN que nos da el resultado por la aplicación directa de la formula. **Ej. :** utilizando el teorema fundamental de la numeración tenemos que 2CA es igual a: $2*16^2+C*16^1+A*16^0=714_{(10)}$

CONVERSIÓN DE HEXADECIMAL-BINARIO: para convertir un número hexadecimal a binario, se sustituye cada dígito hexadecimal por su representación binaria según la siguiente tabla.

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

Ej.: pasar el número 2BC a binario

2	B	C
0010	**1011**	**1100**

Finalmente él número hexadecimal en binario es igual a: **001010111100**

CONVERSIÓN DE OCTAL A BINARIO: para convertir un numero octal a binario se sustituye cada dígito octal en por sus correspondientes tres dígitos binarios según la siguiente tabla.

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

Ej.: Convertir el número octal 1274 en binario.

1	**2**	**7**	**4**
001	**010**	**111**	**100**

Por lo tanto, el número octal en binario es igual a: **001010111100**