

5. Números Enteros

5.1. introducción

- Sistema Posicional:
 - Número en Base Decimal
 - Dígitos, Valor?
 - Centenas, decenas, unidades
 - Posición → índice
 - Posición → pesos → Potencias $base^{posición}$
 - Valor= sumatorio de dígitos ponderados con su peso posicional
 - Ejemplo: 1197

5.2. Base binaria

- Base 2 . Dígitos : 0,1.
- Pesos $2^0, 2^1, 2^2, 2^3, 2^4 \rightarrow 1, 2, 4, 8, 16$
- Conversión Decimal-Binaria:
 - Divisiones sucesivas / 2 \rightarrow Dividendo1 = 2*Cociente1 + Resto1
 - $Cociente1 = 2 * Cociente2 + Resto2 \rightarrow$ Dividendo1 = $2 * (2 * Cociente2 + Resto2) + Resto1 = Resto1 * 2^0 + Resto2 * 2^1 + Cociente2 * 2^2$
 - Resto1 es el dígito binario de la posición 0, Resto2 es el dígito binario de la posición 1, Cociente es el dígito binario de la posición 2.
 - Regla: los dígitos binarios son todos los restos y el último cociente.
 - La división se termina cuando un cociente no es divisible por 2, es decir, el cociente es 1. Este cociente es el MSB.
 - Ejemplo: decimal 1197 \rightarrow binario 10010101101

Cuadro 1: Conversión decimal binario

Número	1ª Div		2ª Div		3ª Div		4ª Div		5ª Div		6ª Div	
	Coc	Resto	Coc	Resto	Coc	Resto	Coc	Resto	Coc	Resto	Coc	Resto
1197	598	1	299	0	149	1	74	1	37	0	18	1

Número	7ª Div		8ª Div		9ª Div		10ª Div	
	Coc	Resto	Coc	Resto	Coc	Resto	Coc	Resto
1197	9	0	4	1	2	0	1	0

5.3. Base Octal

- Base 8
- Dígitos: 0-7
- Pesos: 8 elevado a la posición
- En C se especifica la base con el prefijo 0 \rightarrow int 077;

- Conversión Octal \leftrightarrow Binario y viceversa \rightarrow cada dígito octal se descompone en un binario de 3 bits
- decimal 1197 \rightarrow binario 10010101101 \rightarrow octal 02255

5.3.1. Base Hexadecimal

- Base 16
- Dígitos: 0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F
- Pesos: 16 elevado a la posición
- En C se especifica la base con el prefijo `0x` \rightarrow `int 0xAF;`
- Hexadecimal \leftrightarrow Binario y viceversa \rightarrow cada dígito hexadecimal se descompone en un binario de 4 bits
- decimal 1197 \rightarrow binario 10010101101 \rightarrow `0x4AD`

5.4. Calculadora

- Calculadora en el sistema Linux
 - `candido@lur:~$ echo "obase=2 ; ibase=16; 80AA010F" | bc`
 - 10000000101010100000000100001111
 - `echo "obase=10 ; ibase=16; 80AA010F" | bc` \rightarrow es obligado poner primero la base del formato de salida
 - 2.158.625.039
 - Intérprete `$ bc`

5.5. Python

- <https://docs.python.org/3/tutorial/index.html>
- `help(builtins)`

```
bin(1197) -> '0b10010101101'
oct(1197) -> '02255'
hex(1197) -> '0x4ad'
int(0x4ad) -> 1197
```