



## Programación de Sistemas

Mtro. en IA José Rafael Rojano Cáceres

[rrojano@gmail.com](mailto:rrojano@gmail.com)

<http://www.uv.mx/rrojano>

## Intérpretes

- Los intérpretes son programas que parecen ejecutar código fuente como si éste fuera un ejecutable.
- A diferencia de un compilador que genera todo el código de un "solo paso" el intérprete ejecuta línea a línea al programa.
- Una consideración importante en el diseño de los intérpretes está relacionado a sus reglas sintácticas (**gramáticas**), ya que al ser ejecutados línea a línea cada regla de producción debe ser independiente del resto. Otra característica en los intérpretes es que son lenguajes **no tipificados** ello quiere decir que el valor de sus variables se determina al momento de la ejecución. De esta forma se simplifica el proceso de traducción.

## Representación, Codificación, estructuras de control, acceso a dispositivos, etc.

## Representación (1/n)

- La representación numérica permite codificar la información en alguna forma significativa.
- Para ello existen diferentes sistemas que codifican la información, algunos ejemplos son:
  - Decimal: alfabeto = {0,1,2,3,4,5,6,7,8,9}
    - base = 10.
  - Octal: alfabeto = {0,1,2,3,4,5,6,7}
    - base = 8.
  - Hexadecimal: alfabeto = {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
    - base = 16.
  - Binario: alfabeto = {0,1}
    - base = 2.

## Representación (2/n)

- Ahora si nos referimos a la representación de valores numéricos debemos recordar que la computadora asigna  $n$  cantidad de bits para ello.
- Por ejemplo para los enteros *sin signo* se pueden tener  $2^n - 1$  posiciones.
- Si por el contrario queremos representar números negativos *con signo* deberemos optar por alguna codificación, en general se emplea el complemento a 2. En tal caso se tienen  $-2^{n-1}$  hasta  $2^{n-1} - 1$ .

## Ejemplo

Binary	Hex	Unsigned Decimal	Signed Decimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	-8
1001	9	9	-7
1010	A	10	-6
1011	B	11	-5
1100	C	12	-4
1101	D	13	-3
1110	E	14	-2
1111	F	15	-1

## Ejercicios

- Obtener las tablas de codificación asigno magnitud, complemento a uno y dos, para los siguientes rangos:
  - N=4, N=6, N=8

## Solución a la codificación de 4 bits

### Codificación con 4 bits

binario	Enteros sin signo		Enteros con signo		
	Decimal	Signo/Mag	Comp 1	Comp 2	
0 0 0 0	0	0	0	0	
0 0 0 1	1	1	1	1	
0 0 1 0	2	2	2	2	
0 0 1 1	3	3	3	3	
0 1 0 0	4	4	4	4	
0 1 0 1	5	5	5	5	
0 1 1 0	6	6	6	6	
0 1 1 1	7	7	7	7	
1 0 0 0	8	-0	-7	-8	
1 0 0 1	9	-1	-6	-7	
1 0 1 0	10	-2	-5	-6	
1 0 1 1	11	-3	-4	-5	
1 1 0 0	12	-4	-3	-4	
1 1 0 1	13	-5	-2	-3	
1 1 1 0	14	-6	-1	-2	
1 1 1 1	15	-7	-0	-1	

## Representación de Flotantes

### Punto flotante

- La representación de los números de punto flotante o números reales tiene que ver precisamente con como almacenar la parte *racional* y la parte *fraccional*.
- Para ello generalmente se usa la representación de punto flotante bajo la norma IEEE 754

### Repaso

- 12345 se puede representar con exponentes como:
  - $12345 \times 10^0$
  - $0.12345 \times 10^5$
  - $123450000 \times 10^{-4}$  incluso
  - $0.0012345 \times 10^7$
- El último ejemplo no es bueno ya que si tuviéramos tan solo 5 dígitos de magnitud perderíamos precisión al codificar como:
  - $0.00123 \times 10^7$

## Emplear notación exponencial

- Para el empleo de la notación exponencial se requieren 4 componentes: [Englander 02]
  - El signo del número  $\rightarrow +$
  - La magnitud, llamada mantisa  $\rightarrow 12345$
  - El signo del exponente  $\rightarrow +$
  - La magnitud del exponente  $\rightarrow 3$

Ejemplo:

$$-0.35790 \times 10^{-6}$$

## representación del punto flotante en la computadora



- Aunado al esquema anterior se debe especificar la ubicación implícita del punto mediante la especificación del exceso N.

## IEEE 754

- Esta norma emplea con precisión de 32 o 64 bits.
- Se formatea el exponente con exceso a 127
- Su normalización tiene la forma:
  - 1.MMMMMMM...
- El número 0.0 se define mediante una mantisa de 0 junto con el valor especial de exponente 0.

## Resumen IEEE 754

Exponente	Mantisa	Valor
0	+/- 0	0
0	no 0	$\pm 2^{-126} \times 0.M$
$1-2^{54}$	cualquiera	$\pm 2^{-127} \times 1.M$
255	+/- 0	$\pm \infty$
255	no 0	condición especial

## Ejemplo

- Conversión del número 253.75 a binario de PF
  - Multiplique el valor por 100 = 25375
  - Representélo en binario = 110001100011111 o también  $1.110001100011111 \times 2^{14}$
  - El valor quedaría pues:
    - 0 10001101 110001100011111
      - signo
      - Exceso a 127 (127+ 14)
      - Mantisa (se suprime El 1 inicial)

## Contra ejemplo

- Proceso inverso
  - Dividir entre el equivalente binario del PF para 100 decimal
  - En el formato IEEE 754 convertir 100 a binario  $\rightarrow 1100100$  que es:
    - 010000101100100
  - Dividir el resultado entre este valor usando la división de PF

## representación de caracteres

<http://czyborra.com/utf/index.htm>

- De forma general se puede decir que existen 3 código alfanuméricos principales para representar caracteres:
  - ASCII (American Standard Code for Information Interchange)
  - Unicode
  - EBCDIC (Extended Binary Coded Decimal Interchange Code)

## Juego de caracteres

- La codificación de caracteres se hace a través de lo que se denomina **juego de caracteres**.
- El juego de caracteres tiene dos elementos:
  - El repertorio, que son los caracteres en sí, por ejemplo
    - U+0041 LATIN CAPITAL LETTER A ('A').
  - El código de mapeo, que permite ubicar al carácter dentro del juego de caracteres
    - 65 es el código en ASCII para la 'A'
- Cuando un código  $x$  mapea a un carácter  $c$  se dice que  $x$  es el *code point* de  $c$

## Unicode

- El conjunto de caracteres Unicode emplea 16 bits.
- Unicode permite representar 65536 caracteres cubriendo prácticamente todos los alfabetos del mundo.
- Actualmente tiene definidos 49,000 símbolos, restando 6400 al uso privado y 10000 para usos futuros.
- Tradicionalmente se escribe como U+4 a 6 dígitos.
  - U+00C9, U+1D122

## ISO8859-n

- ISO8859-1 es un juego de caracteres equivalente a los primeros 256 caracteres del Unicode.
- ISO8859-2 a ISO8859-16, son 15 juegos de caracteres que tienen 256 repertorios. Todos ellos comparten los 128 primeros, pero difieren en los restantes 128.

## Windows-1252

- Consiste de un repertorio de 256 caracteres, también conocido como CP1252. Su definición es muy parecida a la del ISO8859-1

## ASCII

- Este juego de caracteres tiene los primeros 128 caracteres idénticos a Unicode.
- Es un estándar bastante utilizado en la transmisión de datos en Internet.
- ASCII es un código de 7 bits (128 caracteres) posteriormente el ANSI definió un código de 8 bits conocido como Latin-1

## Character Encoding

## UTF-8

- La codificación de caracteres o character encoding determina como un carácter se codifica en una cadena de bits.
- Existe una gran cantidad de formas de codificar, pero las más importantes son:
  - UTF-8
  - UTF-16
  - UTF-32

## UTF-32

- Este tipo de codificación es la más simple. Solo codifica cada carácter en 32 bits.
- La codificación de un carácter es simplemente su *code point*.

## UTF-16

- En esta codificación algunos caracteres son codificados en 16 o 32 bits.
- Queda pendiente como investigación de la clase

## UTF-8

- Queda pendiente para investigación de la clase



## Bibliografía

1. [Donovan 72] John Donovan, Systems Programming, McGraw Hill, 1972
2. [Englander 02] Irv Englander, Arquitectura computacional 2da Edición, CECSA, 2002
3. [Jurgens 91] David Jurgens, Help PC 2.10 software de referencia, 1991.
4. [Oney 96] Michael Oney, Systems Programming for Windows 95, Microsoft Press, 1996
5. [Powell 01] Robert Powell, C# and the .NET Framework The C++ perspective, Sams, 2001
6. [Tanenbaum 00] Andrew Tanenbaum, Organización de computadoras un enfoque estructurado, Pearson Education, 2000
7. [Tischer 96] Michael Tischer, PC Interno 5, Marcombo, 1996.
8. [Duran 07] Luis Duran Rodriguez, El gran libro del PC Interno, AlfaOmega, 2007
9. [Conger 92] Conger, James L. , Windows API bible : the definitive programmer's reference, Waite Group Pr