

Clase 2: Explicación de conceptos y sistemas numéricos

Tabla de equivalencia de datos.

Teniendo en cuenta que los datos viajan en impulsos eléctricos llamados bits (0 y 1) decimos que:

8 Bits	1 Byte (Mínima unidad de almacenamiento)
1024 Byte's	1 KByte
1024 KByte's	1 MByte
1024 MByte's	1 GByte
1024 GByte's	1 TByte

Conceptos de digital y analógico:

Digital:

Se dice que una señal es digital cuando las magnitudes de la misma se representan mediante valores discretos en lugar de variables continuas. Por ejemplo, el interruptor de la luz sólo puede tomar dos valores o estados: abierto o cerrado, o la misma lámpara: encendida o apagada.

Analógico:

Se dice que una señal es analógica cuando las magnitudes de la misma se representan mediante variables continuas, esto es análogas a las magnitudes que dan lugar a la generación de esta señal.

Tabla ASCII:

American Standard Code for Information Interchange. (Estándar Americano para Intercambio de Información).

Con la finalidad de estandarizar todos los símbolos informáticos y que estos tengan una referencia de forma de poder acceder desde cualquier sistema operativo fue creada la tabla ASCII. La tabla básica de caracteres ASCII está compuesta por 128 caracteres incluyendo símbolos y caracteres de control. Existe una versión extendida de 256, la forma de presentar un carácter en pantalla es la de Presionar la tecla "ATL" y mantenerla presionada mientras desde el teclado numero se digita el numero del carácter que queremos que se muestre en pantalla.

Tabla ASCII:

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Tabla ASCII extendida:

128	Ç	144	É	161	í	177	⌘	193	⌞	209	⌠	225	β	241	±
129	ù	145	æ	162	ó	178	⌘	194	⌟	210	⌡	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⌠	211	⌢	227	π	243	≤
131	â	147	ô	164	ñ	180	†	196	—	212	⌣	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	‡	197	‡	213	⌤	229	σ	245	∫
133	à	149	ò	166	°	182	‡	198	‡	214	⌥	230	μ	246	+
134	â	150	û	167	°	183	π	199	‡	215	⌦	231	τ	247	±
135	ç	151	ù	168	¿	184	¶	200	⌘	216	⌧	232	Φ	248	°
136	ê	152	—	169	—	185	‡	201	⌘	217	⌨	233	Θ	249	.
137	ë	153	Ö	170	¬	186	‡	202	⌘	218	〈	234	Ω	250	.
138	è	154	Û	171	½	187	¶	203	⌠	219	■	235	δ	251	√
139	ï	156	£	172	¾	188	‡	204	‡	220	■	236	∞	252	—
140	î	157	¥	173	ı	189	‡	205	=	221	■	237	φ	253	z
141	ı	158	—	174	«	190	‡	206	‡	222	■	238	ε	254	■
142	Ä	159	f	175	»	191	¶	207	⌘	223	■	239	∩	255	
143	Å	160	á	176	⌘	192	⌘	208	⌘	224	α	240	≡		

Distintos SO:

Será bueno de mencionar que el SO es el Software básico necesario para el funcionamiento del PC, para eso destacamos las generaciones de los SO de Microsoft y un listado de SO Linux.

Microsoft	Microsoft (Servidores)	Linux
Dos 1.0 – Dos 6.22		Suse
Windows 95	Windows NT 3.0	Red Hat
Windows 98	Windows NT 4.0	Mandraque
Windows ME	Windows 2000	Ubuntu
Windows XP	Windows 2003	
Windows Vista		

Otros SO: Unix, Novell, OS/2, etc.

Sistemas de Numeración

El sistema Decimal:

El sistema de numeración que utilizamos se denomina decimal ya que emplea diez dígitos para indicar una cantidad, y es además un sistema posicional ya que cada dígito debe su valor a la posición que ocupa en la cantidad a la que pertenece.

Valor Posicional	10^3	10^2	10^1	10^0		
4 en 10^3				4	4×10^3	4
3 en 10^1			3		3×10^1	30
5 en 10^2		5			5×10^2	500
1 en 10^0	1				1×10^0	1000

El sistema binario:

El sistema Binario, como el decimal, es un sistema posición; pero el valor de la posición viene dado por potencia de 2 ($2^0, 2^1, 2^2, \dots$) ya que solo se utilizan dos dígitos, el cero y el uno. Por tanto, si queremos convertir un número en base 2 (binario) al sistema decimal (base 10), no tenemos más que multiplicar el dígito (0 o 1) por la potencia de 2 correspondiente a su posición, véase tabla adjunta.

Valor Posicional	2^3	2^2	2^1	2^0		Valor decimal
1 en 2^3				1	1×2^3	1
1 en 2^1			1		1×2^1	2
0 en 2^2		0			0×2^2	0
1 en 2^0	1				1×2^0	8
Total						11

Como $1 + 2 + 0 + 8 = 11$ tenemos que $1011(2) = 11(10)$

Si lo que queremos es convertir un número binario a decimal, dividiremos sucesivamente el valor decimal por 2 hasta llegar a 1. Los restos de de las divisiones nos indicarán el valor binario.

División	Cociente	Resto
11 / 2	5	1
5 / 2	2	1
2 / 2	1	0
1 / 2		1

División	Cociente	Resto
52 / 2	26	0
26 / 2	13	0
13 / 2	6	1
6 / 2	3	0
3 / 2	1	1
1 / 2		1

La lectura se realiza del último resto al primero. 110100

Por lo tanto $52(10) = 110100(2)$

Los ordenadores “utilizan” este sistema de numeración, en cada posición de memoria solo pueden almacenar 1 BIT (o un cero o un uno).

El sistema Hexadecimal:

Como los anteriores, también es posicional. En este caso el valor de la posición viene dado por potencias de 16 (16⁰, 16¹, 16², . . .).

Como solo poseemos 10 caracteres para representar los posibles dígitos, se añaden las letras A, B, C, D, E y F

Por tanto en base 16 disponemos de los siguientes caracteres 0 ,1, 2, 3, 4, 5, 6, 7, 8, 9, A = 10, B = 11, C = 12, D = 13, E = 14 y F = 15.

Para realizar la conversión al sistema decimal seguiremos un método similar al anterior.

Valor Posicional	16 ³	16 ²	16 ¹	16 ⁰		Valor decimal
F en 16 ⁰				F	F x 16 ⁰	15
2 en 16 ¹			2		2 x 16 ¹	32
5 en 16 ²		5			5 x 16 ²	1280
A en 16 ³	A				A x 16 ³	40960
					Total	42287

Por tanto, como $15 + 32 + 1280 + 40960 = 42287$ tenemos que $A52F(16) = 42287(10)$, también se suele representar como A52Fh, indicando la h que se trata de un valor hexadecimal.

Si lo que queremos es convertir una cantidad hexadecimal a decimal, seguiremos un método similar al utilizado con los valores binarios, teniendo en cuenta que si obtenemos como restos 10, 11,12,13, 14,15 debemos sustituirlos por A, B, C, D, E o F

División	Cociente	Resto
332 / 16	20	12 = C
20 / 16	1	4
1 / 16		1

Por tanto $332_{(10)} = 14C_{(16)}$ ó $14Ch$

El sistema hexadecimal se suele utilizar ampliamente en informática, por ejemplo para indicar direcciones de memoria.