



Pascalina (particolare interno)

Informatica

La numerazione

Vibo Valentia, 24 ottobre 2005

Ercole Colonese

e.colonese@virgilio.it

Il sistema binario (B=2)

La base 2 è la più piccola per un sistema di numerazione

Cifre: 0 1 – bit (binary digit)

Esempi:

Forma
polinomia



$$(101101)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = \\ 32 + 0 + 8 + 4 + 0 + 1 = (45)_{10}$$

$$(0,0101)_2 = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = \\ 0 + 0,25 + 0 + 0,0625 = (0,3125)_{10}$$

$$(11,101)_2 = 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = \\ 2 + 1 + 0,5 + 0 + 0,125 = (3,625)_{10}$$

Dal bit al byte

- Un **byte** è un insieme di 8 **bit** (un numero binario a 8 cifre)

$b_7b_6b_5b_4b_3b_2b_1b_0$

- Con un byte si rappresentano 256 numeri interi (da 0 a 255)

00000000

00000001

00000010

00000011

$2^8 = 256$ valori distinti

.....

11111110

11111111

- È l'elemento base con cui si rappresentano i dati nei calcolatori
- Si utilizzano sempre dimensioni multiple (di potenze del 2) del byte: 2 byte (16 bit), 4 byte (32 bit), 8 byte (64 bit)...

Dal byte al kilobyte

- **Potenze del 2**

$$\begin{aligned}2^4 &= 16 \\2^8 &= 256 \\2^{16} &= 65536\end{aligned}$$

$$\begin{aligned}2^{10} &= 1024 && \text{(K=Kilo)} \\2^{20} &= 1048576 && \text{(M=Mega)} \\2^{30} &= 1073741824 && \text{(G=Giga)}\end{aligned}$$

- **Cosa sono KB (Kilobyte), MB (Megabyte), GB (Gigabyte) e TB (Terabyte)?**

$$\begin{aligned}1 \text{ KB} &= 2^{10} \text{ byte} = 1024 \text{ byte} \\1 \text{ MB} &= 2^{20} \text{ byte} = 1048576 \text{ byte} \\1 \text{ GB} &= 2^{30} \text{ byte} = 1073741824 \text{ byte} \\1 \text{ TB} &= 2^{40} \text{ byte} = 1099511627776 \text{ byte}\end{aligned}$$

Dal decimale intero al binario

- Si divide ripetutamente il numero intero decimale per 2 fino ad ottenere un quoziente nullo; le cifre del numero binario sono i resti delle divisioni; la cifra più significativa è l'ultimo resto

Esempio: convertire il numero decimale 43 in binario $(43)_{10}$

$$\begin{array}{r} 43 : 2 = 21 + 1 \\ 21 : 2 = 10 + 1 \\ 10 : 2 = 5 + 0 \\ 5 : 2 = 2 + 1 \\ 2 : 2 = 1 + 0 \\ 1 : 2 = 0 + 1 \end{array}$$

resti

bit più significativo


$$(43)_{10} = (101011)_2$$

Da frazione decimale a binario

- Si moltiplica ripetutamente il numero frazionario decimale per 2, fino ad ottenere una parte decimale nulla o, dato che la condizione potrebbe non verificarsi mai, per un numero prefissato di volte; le cifre del numero binario sono le parti intere dei prodotti successivi; la cifra più significativa è il risultato della prima moltiplicazione


Esempio: convertire in binario $(0,21875)_{10}$ e $(0,45)_{10}$

$$\begin{array}{l} 0,21875 \times 2 = 0,4375 \\ 0,4375 \times 2 = 0,875 \\ 0,875 \times 2 = 1,75 \\ 0,75 \times 2 = 1,5 \\ 0,5 \times 2 = 1,0 \end{array}$$



$$(0,21875)_{10} = (0,00111)_2$$

$$\begin{array}{l} 0,45 \times 2 = 0,9 \\ 0,90 \times 2 = 1,8 \\ 0,80 \times 2 = 1,6 \\ 0,60 \times 2 = 1,2 \\ 0,20 \times 2 = 0,4 \text{ etc.} \end{array}$$



$$(0,45)_{10} \approx (0,01110)_2$$

Da binario a decimale

- L'espansione esplicita in potenze del 2 (forma polinomiale) è la più semplice ...

$$(101011)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (43)_{10}$$
$$= 32 + 0 + 8 + 0 + 2 + 1 = 43$$

- ... ma si può operare anche nel modo seguente: si raddoppia il bit più significativo e si aggiunge al secondo bit; si raddoppia la somma e si aggiunge al terzo bit ... e si continua fino al bit meno significativo

Esempio: convertire in decimale $(101011)_2$

bit più significativo

1	x 2 = 2	+	0	
2	x 2 = 4	+	1	
5	x 2 = 10	+	0	
10	x 2 = 20	+	1	
21	x 2 = 42	+	1	= 43

$$(101011)_2 = (43)_{10}$$

La numerazione

Sistema esadecimale

- La base 16 è molto usata in campo informatico

Cifre: 0 1 2 3 4 5 6 7 8 9 A B C D E F

- La corrispondenza in decimale delle cifre oltre il 9 è

$$\begin{array}{ll} A = (10)_{10} & D = (13)_{10} \\ B = (11)_{10} & E = (14)_{10} \\ C = (12)_{10} & F = (15)_{10} \end{array}$$

Esempio:

$$\begin{aligned} (3A2F)_{16} &= 3 \times 16^3 + 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = \\ &= 3 \times 4096 + 10 \times 256 + 2 \times 16 + 15 = (14895)_{10} \end{aligned}$$

Da binario a esadecimale

- Una cifra esadecimale corrisponde a 4 bit

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

- Si possono rappresentare numeri binari lunghi con poche cifre (1/4)
- La conversione da binario ad esadecimale è immediata, raggruppando le cifre binarie in gruppi di 4 (da destra) e sostituendole con le cifre esadecimali secondo la tabella precedente

Codifica dei caratteri alfabetici

- Oltre ai numeri, le applicazioni informatiche elaborano anche caratteri (**simboli**)
- Gli elaboratori elettronici invece trattano numeri
- Si codificano i caratteri e i simboli per mezzo di numeri
- Per poter scambiare dati (**testi**) in modo corretto, occorre definire uno standard di codifica

A	→	01000001
3	→	00110011
\$	→	00100100

Codifica dei caratteri alfabetici ...

- Quando si scambiano dati, deve essere noto il tipo di codifica utilizzato
- In genere un sistema informatico deve supportare più standard di codifica
- La codifica deve prevedere le lettere dell'alfabeto, le cifre numeriche, i simboli, la punteggiatura, i caratteri speciali per certe lingue (æ, ã, ë, è, ...)
- Il più diffuso è il codice **ASCII (American Standard Code for Information Interchange)**

... Codifica ASCII

- Definisce una tabella di corrispondenza fra ciascun carattere e un codice a 7 bit (128 caratteri)
- I caratteri, in genere, sono rappresentati con 1 byte (8 bit); i caratteri con il bit più significativo a 1 (quelli con codice dal 128 al 255) rappresentano un'estensione della codifica
- La tabella comprende sia caratteri di controllo (codici da 0 a 31) che caratteri stampabili
- I caratteri alfabetici/numerici hanno codici ordinati secondo l'ordine alfabetico/numerico

0 48	A 65	a 97
1 49	B 66	b 98
.....
8 56	Y 89	y 121
9 57	Z 90	Z 122
cifre	maiuscole	minuscole

Caratteri di controllo ASCII

- I caratteri di controllo (codice da 0 a 31) hanno funzioni speciali
- Si ottengono o con tasti specifici o con una sequenza Ctrl+carattere

Ctrl	Dec	Hex	Code	Nota
^@	0	0	NULL	carattere nullo
^A	1	1	SOH	partenza blocco
.....
^G	7	7	BEL	beep
^H	8	8	BS	backspace
^I	9	9	HT	tabulazione orizzontale
^J	10	A	LF	line feed (cambio linea)
^K	11	B	VT	tabulazione verticale
^L	12	C	FF	form feed (alim. carta)
^M	13	D	CR	carriage return (a capo)
.....
^Z	26	1A	EOF	fine file
^[27	1 B	ESC	escape
.....
^_	31	1F	US	separatore di unità

Caratteri ASCII stampabili

Dec Hx Chr Dec Hx Chr Dec Hx Chr Dec Hx Chr Dec Hx Chr Dec Hx Chr

32	20	SPACE	48	30	0	64	40	@	80	50	P	96	60	`	112	70	p
33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
36	24	\$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
40	28	(56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
41	29)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
43	2B	+	59	3B	;	75	4B	K	91	5B	[107	6B	k	123	7B	{
44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	
45	2D	-	61	3D	=	77	4D	M	93	5D]	109	6D	m	125	7D	}
46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

Nota: il valore numerico di una cifra può essere calcolato come differenza del suo codice ASCII rispetto al codice ASCII della cifra 0 (es. '5'-'0' = 53-48 = 5)

Tabella ASCII completa

- I codici oltre il 127 non sono compresi nello standard originario

128	Ç	144	É	160	á	176	☒	193	⊥	209	ƒ	225	ß	241	±
129	ü	145	æ	161	í	177	☑	194	⊤	210	π	226	Γ	242	≥
130	é	146	Æ	162	ó	178	☐	195	⊢	211	ℓ	227	π	243	≤
131	â	147	ô	163	ú	179		196	—	212	ℓ	228	Σ	244	ƒ
132	ä	148	ö	164	ñ	180	†	197	+	213	ƒ	229	σ	245	∫
133	à	149	ò	165	Ñ	181	‡	198	‡	214	π	230	μ	246	+
134	â	150	û	166	ª	182	‡	199	‡	215	‡	231	τ	247	∞
135	ç	151	ù	167	º	183	π	200	ℓ	216	‡	232	φ	248	°
136	ê	152	_	168	¿	184	¶	201	¶	217	∫	233	⊙	249	.
137	ë	153	Ö	169	_	185	‡	202	≠	218	∫	234	Ω	250	.
138	è	154	Û	170	¬	186		203	¶	219	■	235	δ	251	√
139	ï	156	£	171	½	187	¶	204	‡	220	■	236	∞	252	_
140	î	157	¥	172	¾	188	∫	205	=	221	■	237	φ	253	²
141	ï	158	_	173	¡	189	∫	206	‡	222	■	238	ε	254	■
142	À	159	ƒ	174	«	190	∫	207	≠	223	■	239	∧	255	
143	Á	192	∟	175	»	191	∫	208	≠	224	α	240	≡		

Codifica UNICODE ...

- È lo standard emergente per la codifica dei caratteri nei testi; è basato sulle caratteristiche del codice ASCII, ma supera la limitazione di poter rappresentare in modo coerente solo l'alfabeto latino
- Fornisce un **unico codice per ogni carattere di ogni lingua scritta**, indipendentemente dalla piattaforma, dal linguaggio o dal programma
- Lo standard iniziale prevedeva di codificare i caratteri con 16 bit, per un totale di oltre 65.000 caratteri rappresentabili

... Codifica UNICODE

- La versione 3.0 dello standard fornisce i codici per **49.194** caratteri, derivati dagli alfabeti usati nel mondo, dagli insiemi di ideogrammi, dalle collezioni di simboli
- L'ultima versione dello standard definisce tre tipi diversi di codifica che permettono agli stessi dati di essere trasmessi in
 - byte (8 bit – UTF-8),
 - word (16 bit – UTF-16) o
 - double word (32 bit – UTF-32)
- Tutte le codifiche presuppongono, al più, l'utilizzo di 32 bit per carattere

Codifica delle immagini ...

- Le immagini sono anch'esse codificate come una sequenza di bit: il processo di “traduzione” da un'immagine ad una sequenza binaria prende il nome di **digitalizzazione**
 - ✓ L'immagine è suddivisa in punti o **pixel** (picture element), e ciascun punto viene codificato con un numero, che corrisponde ad un colore o ad un particolare tono di grigio
 - ✓ Si utilizza un numero di colori o di sfumature che sia una potenza del 2, in modo da codificare l'informazione legata a ciascun pixel con un opportuno numero di bit

... Codifica delle immagini ...

- Le immagini sono memorizzate come lunghe sequenze di bit: per interpretarle è necessario conoscere ...
 - ✓ ... le dimensioni dell'immagine (base ed altezza in numero di pixel), detta anche **risoluzione**
 - ✓ ... il numero di colori (o toni di grigio) disponibili per ogni pixel
- Se un'immagine è codificata ad una data risoluzione, potrà comunque essere presentata su un dispositivo a più bassa risoluzione, a patto di “ignorare” alcuni dei bit che descrivono i pixel

... Codifica delle immagini ...

- Come è avvenuto per i caratteri, anche per le immagini sono stati definiti standard di codifica, che assicurano la compatibilità fra sistemi diversi, per quanto concerne la trasmissione e la visualizzazione delle immagini
 - ✓ TIFF
 - ✓ JPEG
 - ✓ PNG
- Per ridurre lo spazio necessario per memorizzare le immagini si utilizzano tecniche di **compressione** (utili anche per la trasmissione su rete Internet)

... Codifica delle immagini

- Le tecniche di compressione si dividono in...
 - ✓ **Tecniche lossless**: non provocano perdita di informazione; sono adatte a codificare immagini in cui sono presenti ampie aree monocromatiche \Rightarrow si codificano in maniera compatta insieme di pixel aventi le stesse caratteristiche
 - ✓ **Tecniche lossly**: provocano perdita di informazione, facendo decadere la qualità dell'immagine
- Normalmente ai formati **JPEG** e **PNG**, molto diffusi per lo scambio di immagini su Internet, si applicano metodi di compressione **lossly**