BIt WOrld

rappresentazione dei dati in un computer

(2011.02)

Indice

Introduzione

Il computer è una macchina a stati binari: nella sua memoria sono conservate configurazioni di simboli che possono assumere solo due valori (On/Off o, da convenzione comune, 1/0). Questi appunti trattano di *cosa* e *in che modo* si possono codificare, utilizzando due soli stati, informazioni di varia natura. Le informazioni che possono trovare posto, nella memoria di un computer, vanno dai dati numerici (i computer nascono per velocizzare calcoli numerici) ai dati alfanumerici. Con la sempre maggiore diffusione dei computer, la disponibilità di memorie sempre più capienti a prezzi sempre più bassi, la diffusione di Internet e le sue possibilità di permettere condivisione veloce, le necessità di conservazione si sono estese anche ai cosiddetti formati multimediali: immagini, suoni, filmati.

Il linguaggio binario assume così, e lo sarà sempre di più, la valenza di un sistema di codifica universale di dati di qualsiasi formato anche se, come si esaminerà, ciò non è privo di problematiche dovute all'adattamento del sistema alla rappresentazione di dati con caratteristiche, da questo, molto diverse.

La sequenza degli argomenti trattati percorre la strada della codifica dei dati, partendo da quelli numerici per passare poi a quelli alfanumerici e multimediali mettendo in evidenza, di volta in volta, le diversità rispetto al modo comune di pensare a quel tipo di dato e i limiti di affidabilità delle varie codifiche in ragione delle peculiarità di un computer.

Questi appunti non intendono essere esaustivi e quindi non vengono esaminati, soprattutto nella parte riguardante i formati multimediali, tutti i formati esistenti, ma soltanto quelli più comunemente usati al fine della comprensione dei meccanismi di base della conservazione dei dati.

I dati numerici

Intanto occorre chiarire che si definisce un tipo di dato numerico o alfanumerico, non tanto in base ai simboli utilizzati (lettere o cifre numeriche), quanto piuttosto in base alle operazioni che si intendono effettuare su di esso. Se le operazioni previste sono quelle dell'aritmetica elementare (somme, sottrazioni, moltiplicazioni, divisioni) si parlerà di formati numerici, negli altri casi si parlerà di formati alfanumerici. Per esempio il Codice di Avviamento Postale (CAP) di una città è un dato alfanumerico e così lo sarà anche, per esempio, un numero di telefono o il numero di partita IVA di una azienda.

A parte il fatto che, come ricordato prima, nella rappresentazione dei numeri all'interno di un computer viene utilizzato un codice che prevede solo due simboli (codice binario), c'è una ulteriore differenza con i numeri che si utilizzano nelle comuni elaborazioni umane. La dimensione della memoria fisica di un computer porta come conseguenza l'uso, per i numeri, di aritmetica a precisione finita e fissa. Per esempio: se si potessero conservare numeri in formato decimale, se si stabilisse di utilizzare 2 cifre per la rappresentazione dei numeri decimali interi e senza segno, si potrebbero rappresentare solo i numeri che rientrano fra: 00, 01, 02, ... 99.

Le operazioni all'interno di questo insieme di numeri evidenziano alcuni limiti:

- 05 + 98 = 103

  05 - 08 = -3

  La prima operazione produce un risultato più grande del più grande numero rappresentabile con la convenzione utilizzata e, quindi, non è rappresentabile. Lo stesso vale per la seconda

operazione solo che, stavolta, il risultato non è rappresentabile perché più piccolo del più piccolo disponibile. Nel primo caso si parla di un errore di *overflow*, nel secondo di un errore di *underflow*.

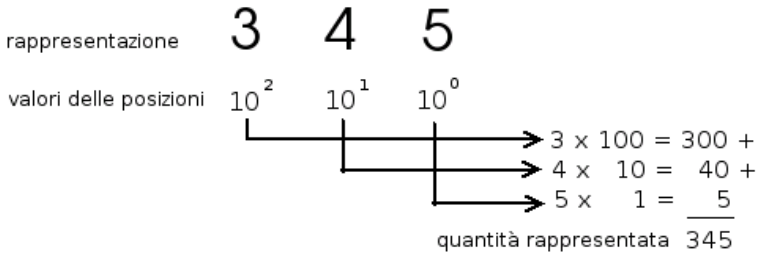
- 05 / 02 = 2.5

  In questo caso il risultato non è rappresentabile perché non fa parte dell'insieme dei numeri rappresentabili: è di altro tipo.

Anche se si modificano le convenzioni per la rappresentazione dei numeri, i problemi evidenziati si pongono sempre. In definitiva, rispetto alla Matematica, i risultati di certe operazioni, nel mondo dell'aritmetica a precisione finita, sono sbagliati. Tutto ciò non vuol dire che i computer non sono macchine affidabili, altrimenti non sarebbero universalmente presenti, ma che bisogna conoscerne il funzionamento in modo da avere presente limiti e campo di applicabilità.

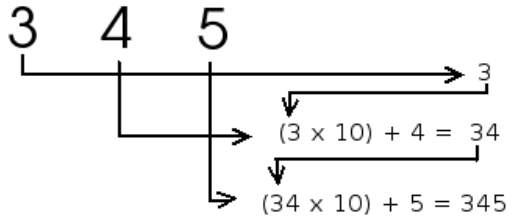
Sistemi di numerazione

I numeri che, generalmente, si incontrano nelle comuni elaborazioni sono espressi come sequenze di simboli ognuno dei quali facente parte dell'insieme 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Si potrebbe dire che si tratta di *parole* formate utilizzando i caratteri dell'*alfabeto decimale* (perché formato da 10 simboli) se non fosse per il fatto che, queste stringhe, rappresentano delle quantità. Per individuare la quantità rappresentata bisogna tener conto del fatto che i numeri ordinari sono espressi in **notazione posizionale**:



ogni cifra (il singolo simbolo) rappresenta una quantità dipendente anche dal posto che occupa all'interno della rappresentazione, per cui, ordinariamente, si dice che nel numero 345 ci sono 3 centinaia, 4 decine e 5 unità e la quantità associata alla rappresentazione del numero viene calcolata sommando le quantità rappresentate dalle singole cifre.

Come conseguenza della notazione posizionale si può notare che il passaggio da una posizione all'altra, all'interno della rappresentazione del numero, comporta la moltiplicazione per il valore 10. Si può calcolare il valore associato al numero anche cominciando dalla cifra più a sinistra (la *cifra più significativa*, quella con valore di posizione maggiore), riportandola alla posizione successiva e sommando il valore alla cifra esistente in quella posizione e così via fino a raggiungere la cifra più a destra (la *cifra meno significativa*, con valore di posizione minore):

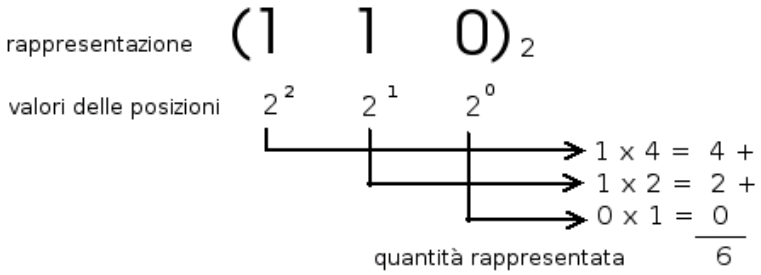


I valori delle posizioni, nell'esempio, dipendono dal fatto che si è utilizzato il sistema decimale

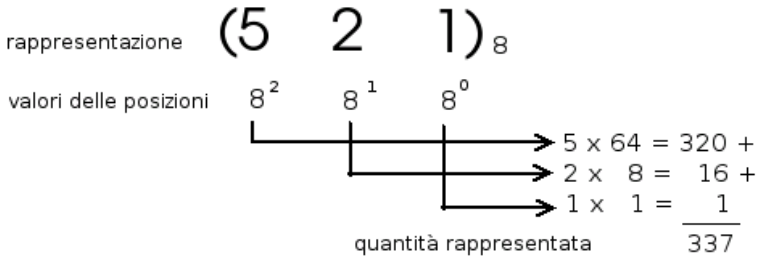
ovvero si è utilizzata la **base 10** per la rappresentazione del numero. La notazione posizionale permette la scrittura di un numero qualsiasi utilizzando un ristretto numero di simboli; la quantità di simboli può essere variata a piacimento. Ovviamente cambiando la base del sistema di numerazione, nel caso di calcolo della quantità rappresentata, variano, oltre alla quantità di simboli utilizzabili, anche i valori legati alle posizioni delle cifre.

Nel mondo del computer interessano rappresentazioni che utilizzano le basi 2, 8, 16:

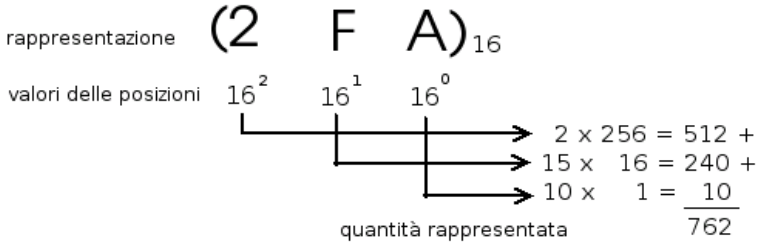
- **Sistema Binario**: i simboli utilizzati sono 0, 1. Le cifre di un numero binario si chiamano bit (BInary digiT, cifra binaria).



- **Sistema Ottale**: i simboli utilizzati sono 0, 1, 2, 3, 4, 5, 6, 7



- **Sistema Esadecimale**: i simboli utilizzati sono 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. In questo caso poiché i simboli della rappresentazione superano quelli del sistema decimale, vengono utilizzate le prime lettere maiuscole dell'alfabeto, con i valori: 10 (A), 11 (B), 12 (C), 13 (D), 14 (E), 15 (F). In questo sistema di numerazione, un numero, fino al valore 15 viene scritto utilizzando una sola cifra.



Nelle varie rappresentazioni, quando necessario, per evitare ambiguità nel calcolo del valore rappresentato, viene specificata la base utilizzata per la scrittura del numero.

Il calcolo del valore rappresentato in una qualsiasi base può essere, anche, effettuato con il metodo visto nell'esempio del numero decimale, naturalmente sostituendo, al 10, il valore della base in cui è scritto il numero di cui si vuole calcolare la quantità rappresentata.

Il sistema binario utilizzando solo due simboli è quello più adatto per rappresentare i numeri in un computer. Spesso però, per motivi pratici di leggibilità, si utilizzano anche i sistemi ottale ed

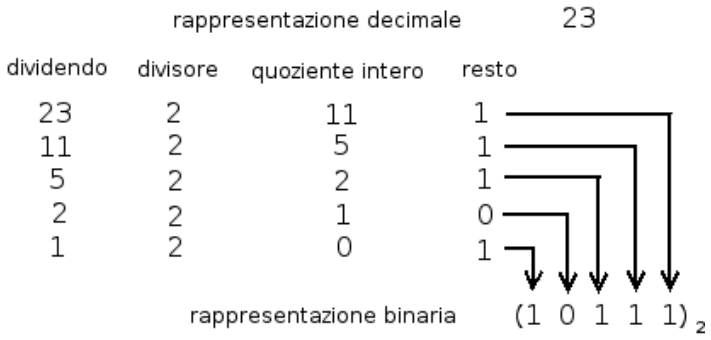
esadecimale.

Conversioni di base

La notazione posizionale dei sistemi di numerazione consente facilmente di convertire un numero da una base ad una qualsiasi altra. Si è visto in precedenza come ricavare il valore decimale da un numero scritto in binario, ottale o esadecimale. Questi sono i sistemi di interesse e sono gli unici presi in esame anche se le regole applicate, con i dovuti adattamenti, valgono per un numero scritto in qualsiasi base.

Per convertire un numero decimale nella sua rappresentazione in una base diversa si può applicare il seguente algoritmo: *si calcola il quoziente intero fra il numero decimale e la base di destinazione e successivamente si ripete la divisione considerando il quoziente ottenuto dall'operazione precedente e la base di destinazione. Il processo termina quando il quoziente risulta nullo. Le cifre, della rappresentazione del numero nella base richiesta, sono i resti delle varie divisioni scritte a partire dalla cifra meno significativa e procedendo fino a quella più significativa (ultimo resto della divisione con quoziente zero).*

Per ulteriore chiarezza, a titolo di esempio, si riporta la conversione in binario di un numero decimale:



Se serve convertire un numero decimale in esadecimale, basta eseguire le successive divisioni intere per 16.

I numeri binari, in ragione del basso valore delle posizioni, hanno rappresentazioni molto estese. Il numero 23, dell'esempio, ha 5 cifre nella rappresentazione binaria. Anche un numero con valore piccolo ha una rappresentazione, in binario, formata da una lunga stringa di 0 e 1, con difficoltà di lettura e facilità di inversioni fra bit qualora si debba trascrivere. Per questi motivi, anche se i dati numerici nel computer sono conservati come stringhe binarie, è comune utilizzare rappresentazioni che facilmente possono trasformare un numero binario in qualcosa di più comprensibile e più corto.

A questo scopo si utilizzano i sistemi ottale ed esadecimale per i rapporti che queste basi hanno con la binaria: 8 e 16 sono infatti, rispettivamente, la terza e la quarta potenza del 2. Questo vuol dire che una cifra ottale vale 3 cifre binarie e una cifra esadecimale ne vale 4:

Tabella di conversione

binario	ottale
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

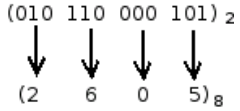
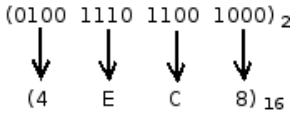


Tabella di conversione

binario	esadecimale	binario	esadecimale
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
0111	7	1111	F



Il numero binario si suddivide a gruppi di 3 o 4 cifre (a seconda se si converte in ottale o esadecimale) cominciando dalla cifra meno significativa e aggiungendo, se serve e come riportato negli esempi, zeri nella parte significativa, in modo da completare i gruppi. Ad ogni gruppo si fa, poi, corrispondere la conversione nella base di destinazione del gruppo, come anche riportato nelle due tabelle di conversione.

Operazioni con i numeri binari

Ammesso che si sia scelto di rappresentare numeri binari utilizzando 5 bit, la somma fra due numeri binari si effettua con le regole solite dell'aritmetica:

```
riporti   1 1
          10011 +
          01011
        -------
          11110
```

Nell'operazione mostrata sono evidenziati i riporti che qui, a differenza dei numeri decimali, si presentano quando la somma supera 1: il numero 2 prende il posto del 10 nel sistema decimale. Se c'è un riporto oltre la quinta cifra, si presenta un errore di overflow: il numero è troppo grande per poter essere rappresentato con 5 bit.

Nella rappresentazione a 5 bit dell'esempio devono essere codificati anche numeri negativi. Il sistema comunemente utilizzato nei computer per rappresentare numeri negativi è quello del **complemento a due**: il bit più significativo indica il segno (0 per positivo, 1 per negativo), la codifica dei numeri positivi avviene nel modo solito. Per codificare un numero negativo si segue la regola: prima si codifica il numero positivo, si commutano i bit (0 diventa 1 e 1 diventa 0) e, infine si aggiunge 1.

codifica in complemento a due con 5 bit

00000 = 0	10000 = -16
00001 = 1	10001 = -15
00010 = 2	10010 = -14
00011 = 3	10011 = -13
00100 = 4	10100 = -12
00101 = 5	10101 = -11
00110 = 6	10110 = -10
00111 = 7	10111 = -9
01000 = 8	11000 = -8
01001 = 9	11001 = -7
01010 = 10	11010 = -6
01011 = 11	11011 = -5
01100 = 12	11100 = -4
01101 = 13	11101 = -3
01110 = 14	11110 = -2
01111 = 15	11111 = -1

Di tutte le possibili combinazioni che si possono fare con la quantità di bit assegnata, metà rappresentano numeri positivi e metà rappresentano numeri negativi. In coerenza con questa rappresentazione, lo zero è un numero positivo.

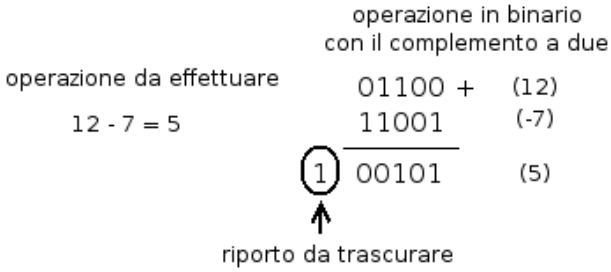
Per calcolare il margine di rappresentatività (l'intervallo dei valori che possono essere rappresentati) basta calcolare la quinta potenza di 2 (32) e dividere a metà il numero e considerare che il numero zero rientra tra i numeri positivi.

In definitiva utilizzando una codifica binaria in complemento a due con 5 bit, si può rappresentare qualsiasi numero intero che soddisfi le diseguaglianze:

$$-16 \leq x \leq +15$$

La differenza fra due numeri si ottiene aggiungendo al primo numero, il secondo negativo e

trascurando il riporto oltre la quinta cifra:



In questa rappresentazione l'operazione 14 + 2 produce un overflow e -14 -5 un underflow.

Per quanto riguarda le rimanenti due operazioni dell'aritmetica elementare (moltiplicazione e divisione), valgono le solite regole con l'unica avvertenza che, per esempio, il quoziente della divisione fra 5 e 2, essendo un numero non intero, non fa parte dell'insieme dei numeri rappresentabili e ci si limiterebbe a considerare il quoziente intero, cioè, nell'esempio, 2.

Il formato binario con complemento a due viene utilizzato in C/C++ per le variabili di tipo `int`, la quantità di bit utilizzati, da cui è possibile dedurre facilmente l'insieme degli interi rappresentabili, si può conoscere facendo visualizzare il risultato della funzione `sizeof(int)`.

Floating point

La rappresentazione in binario puro, così come si è trattata, va bene per numeri interi e non molto grandi: si pensi a quanti bit occorrerebbero per conservare in memoria la misura della distanza fra la terra e il sole o la massa di un elettrone. C'è da aggiungere che, nei casi elencati, si tratta di numeri molto grandi o molto piccoli ma che non hanno necessità di avere molte cifre di precisione. Per esempio il numero, scritto nella notazione scientifica, $12x10^{128}$ ha una grandezza di 128 cifre ma una precisione di 2 cifre.

In Informatica questo tipo di numeri vengono chiamati floating point (virgola mobile). Qualsiasi numero si può sempre esprimere come formato da una parte intera nulla e la prima cifra decimale diversa da zero: basta moltiplicare per una opportuna potenza del 10. Per esempio il numero 12,34 può essere espresso come $0,1234x10^2$. Questa ultima viene chiamata **forma normalizzata**. L'esponente del 10 (2) viene chiamato **caratteristica**, la parte decimale viene chiamata **mantissa**.

Naturalmente in un computer si possono conservare solo stringhe binarie, ma qui per chiarire le proprietà e i limiti del formato floating point, si assume di conservare il numero in formato decimale. Questo non fa perdere di generalità a quanto si farà notare perché: rispetto alla rappresentazione, reale nel computer, del formato binario, cambiano soltanto i valori, ma le proprietà, essendo conseguenza del metodo, rimangono uguali.

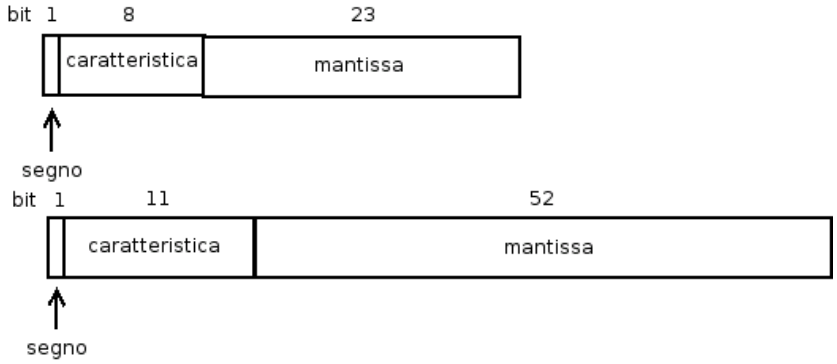
Se si sceglie di rappresentare un numero floating point utilizzando 2 cifre sia per la caratteristica che per l'esponente, si potranno rappresentare i numeri da $-0,99x10^{99}$ a $+0,99x10^{99}$.

- La caratteristica fornisce l'ordine di grandezza: in questo caso utilizzando soltanto quattro cifre (2 per la caratteristica e 2 per la mantissa) si riescono a conservare numeri con 99 cifre.
- Il numero di cifre della mantissa fornisce le cifre di precisione dei numeri rappresentati.

Il formato floating point è quanto di più vicino ai numeri reali matematici. Ci sono tuttavia delle grosse differenze:

- Se il risultato di una operazione produce un valore fuori dal range dei numeri rappresentati, c'è un errore di overflow e il risultato è sbagliato. In questi casi c'è poco da fare: l'errore è dovuto alla finitezza della rappresentazione.
- C'è un intorno del numero zero in cui i numeri non sono rappresentabili. Per esempio, utilizzando due cifre sia per la caratteristica che per la mantissa, non sono rappresentabili numeri fra zero e $+0{,}01 \times 10^{-99}$, così per numeri piccoli vicini allo zero dalla parte negativa (errori di underflow). In questi casi il problema è risolvibile: zero è una approssimazione soddisfacente.
- Fra due reali qualsiasi c'è sempre un ulteriore reale (insieme continuo). Non è così per i floating point: l'insieme è discreto. Anche in questi casi, come il precedente, se il risultato non è rappresentabile si può adottare l'arrotondamento al valore rappresentabile più vicino.

Verso la fine degli anni '70 la IEEE (Institue of Electrical Electronics Engineers) istituì un comitato per la standardizzazione del formato floating point. I risultati furono le definizioni di due formati: singola precisione a 32 bit e doppia precisione a 64 bit con, rispettivamente, 23 e 52 cifre binarie di precisione.



La caratteristica è conservata con il sistema **eccesso 2^7** ed **eccesso 2^{10}**, rispettivamente (l'esponente della potenza è espresso dalla quantità di bit meno una unità). Seguendo questo metodo, se si segue l'esempio, riportato prima, di utilizzare 5 bit:

- la caratteristica +5 verrebbe conservata come $2^4 + 5 = 21 = (10101)_2$
- la caratteristica -5 verrebbe conservata come $2^4 - 5 = 11 = (01011)_2$

In tutte e due i casi si trova lo stesso numero del sistema complemento a due, ma con il bit del segno invertito.

Lo standard IEEE definisce pure, per la risoluzione degli errori di overflow, una particolare registrazione (caratteristica con tutte cifre 1 e mantissa con tutte cifre 0) chiamata **infinito macchina** che segue le stesse regole della matematica: per esempio aggiunto ad un numero qualsiasi ritorna come risultato sé stesso.

Il formato floating point è utilizzato da C/C++ per le variabili di tipo `float` (singola precisione) e `double` (doppia precisione).

L'elaboratore è provvisto di hardware per l'aritmetica binaria e l'aritmetica floating point.

I dati alfanumerici

La codifica dei dati alfanumerici è meno problematica di quella dei dati numerici: i caratteri utilizzati in questi dati appartengono ad insiemi finiti. Le problematiche da considerare prevedono la scelta di una quantità di bit tali da poter permettere la rappresentazione dell'insieme dei caratteri e il rispetto dei criteri di ordinamento dei caratteri: qualsiasi sia la codifica adottata, la stringa 'a' deve precedere 'b' o 'ab'.

L'alfabeto inglese è stato il primo codificato sia per questione geografiche (i computer sono nati in nazioni che parlano lingue anglosassoni) che per la sua semplicità (non ci sono, per esempio, le varianti accentate dell'alfabeto italiano).

Considerando la quantità dei caratteri da codificare si arrivò alla conclusione che 7 bit bastavano per codificare tutti i caratteri dell'alfabeto. Con 7 bit si possono produrre $2^7 = 128$ combinazioni diverse: bastano per i caratteri da rappresentare e, in più, si possono utilizzare alcune combinazioni per codificare i codici di controllo che viaggiano dentro un computer.

Inizialmente ogni computer utilizzava la propria codifica che, naturalmente, era diversa da quella di un altro. In questo modo, però, non c'era alcuna possibilità di comunicazione fra computer diversi: era necessario uniformare le codifiche. Al giorno d'oggi la codifica universalmente utilizzata è il **codice ASCII** (American Standard Code for Information Interchange). Esiste, ancora, un altra codifica utilizzata, però, soltanto nei grossi computer IBM: è la codifica EBCDIC (Extended Binary Coded Decimal Interchange Code).

Codifica ASCII

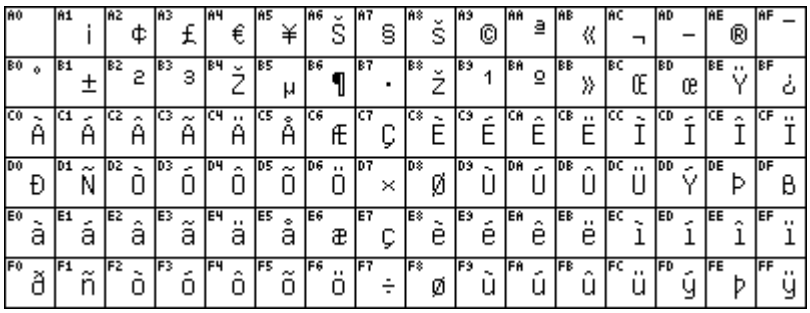
Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Coerentemente con l'ordinamento alfabetico, per esempio, la lettera 'A' ha un codice immediatamente precedente quello di 'B' (rispettivamente, in decimale, 65 e 66), così che il computer può effettuare un ordinamento che segue le stesse regole utilizzate nell'alfabeto rappresentato.

Per conservare in memoria un singolo carattere in ASCII, teoricamente, basterebbero 7 bit; in realtà ne furono utilizzati 8. L'ottavo bit (quello più significativo) veniva utilizzato come codice di controllo durante la trasmissione del carattere.

Con la diffusione del computer, e l'aumento dell'affidabilità dell'hardware, per implementare i caratteri particolari di alcune lingue, per esempio le lettere accentate in italiano, si pensò di utilizzare l'ottavo bit per espandere il set di caratteri. In questo modo il set di caratteri rappresentabili raddoppia: i codici da 128 a 255, in decimale, furono utilizzati per le localizzazioni. Per poter permettere le comunicazioni fra computer fu stabilito uno standard: **ISO 8859** (International Organization for Standardization).

ISO 8859-15



L'ISO 8859 comprende diverse parti che contengono i caratteri specifici di una famiglia di linguaggi: ISO 8859-1, chiamato anche *latin-1*, è la parte che copre la quasi totalità dei linguaggi dell'Europa dell'Ovest. Successivamente all'introduzione del simbolo dell'euro, fu creato anche ISO 8859-15, ovvero *latin-9*. La differenza fra ISO 8859-1 e ISO 8859-15 consiste, appunto, nella presenza, nel secondo, del simbolo €.

Il sistema ad 8 bit e l'uso del semplice (senza lettere accentate) alfabeto inglese ha permesso la diffusione di semplici tecnologie di comunicazione. Tuttavia anche con le estensioni ISO, non si è in grado di includere tutte le lingue del mondo: obiettivo perseguito per una comunicazione globale. Inoltre parecchie lingue utilizzano caratteri composti: per esempio, in italiano, una lettera accentata si ottiene come sovrapposizione fra la lettera e l'apostrofo.

Unicode o ISO 10646, standard in costruzione, si propone di stabilire un insieme di regole per rappresentare tutti i possibili caratteri. La rappresentazione nella memoria di un computer prevede una lunghezza variabile in ragione delle esigenze. La realizzazione pratica di questo tipo di codifica comporta l'utilizzo di uno o più byte a seconda del carattere da rappresentare.

La transizione verso l'uso della codifica dell'insieme dei caratteri universali è la codifica **UTF-8** (Unicode Transformation Format). Questa codifica permette la rappresentazione, per mezzo dell'uso di più unità di 8 bit, di lettere non presenti nell'alfabeto inglese. I codici corrispondenti al codice ASCII (bit più significativo 0) vengono rappresentati con un solo byte e seguono la solita codifica. Gli altri caratteri, per esempio quelli contenuti nell'ISO 8859-15, sono codificati in più byte: il primo byte (che ha il bit più significativo 1, per distinguerlo dalla codifica a un solo byte), conserva il numero di byte della rappresentazione; nei successivi byte è codificato il carattere.

Codifica e conservazione di dati digitali

Inizialmente il computer veniva utilizzato per l'elaborazione di dati numerici o alfanumerici ma con il passare del tempo, la diffusione dei personal computer e la presa di coscienza, da parte degli utenti, delle potenzialità di queste macchine, sorge l'esigenza e si studiano i modi come utilizzare il computer per l'elaborazione di informazioni più generali. I problemi che si pongono a questo punto sono di due ordini:

- Come trasformare in dati digitali (che utilizzano soltanto i due stati 1 e 0) informazioni riguardanti cose che, almeno all'apparenza, sembrano non avere niente a che fare con il mondo digitale: una immagine, un suono, un'animazione. In questi casi si tratta di studiare i modi migliori per conservare e riprodurre una immagine o un suono nel modo più vicino possibile alla realtà. Per fare un esempio relativo ad altri ambiti si può pensare alle regole della prospettiva che permettono di rappresentare un oggetto tridimensionale in un foglio di carta che, per sua natura, si presterebbe soltanto a rappresentare oggetti bidimensionali. Nei prossimi paragrafi si esporranno le principali tecnologie che stanno alla base della digitalizzazione di dati multimediali.
- Come conservare, per esempio, in una memoria di massa i dati digitali. Seguendo l'esempio precedente si potrebbe dire che bisogna stabilire in che modo conservare in un file, assieme ai dati sull'immagine, le regole della prospettiva in modo che un software possa, seguendo le regole descritte, interpretare in modo corretto i dati conservati.

Le convenzioni utilizzate per scrivere, interpretare e leggere i contenuti di un file vengono chiamate **formato di file**. Un file è semplicemente una stringa di byte e per rappresentare cose diverse è indispensabile stabilire il significato dei singoli byte: per esempio in un file che conserva una immagine, i primi byte potrebbero essere interpretabili come le dimensioni dell'immagine stessa. Il formato di un file può essere identificato:

- per mezzo dell'**estensione**: si tratta di una serie di lettere (storicamente 3) precedute da un punto e accodate al nome del file. L'estensione indica le modalità di interpretazione del contenuto del file, servono all'utente per riconoscere immediatamente il genere del documento esaminato e può essere utilizzata per associare al file un applicativo così da permettere, in ambienti grafici, di aprire l'applicativo con il file già caricato semplicemente con un doppio click del mouse sul documento.

  Alcune estensioni comuni:

Estensione	***Tipo di file***
`.txt`	File di testo *puro* senza formattazioni
`.odt`	File generati da Writer di OpenOffice
`.doc .docx`	File generati da Word di Office
`.htm .html`	Pagine web statiche
`.php .asp`	Pagine web dinamiche
`.zip .rar`	File compressi
`.png .gif .jpg`	File contenenti immagini
`.vaw .mp3 .ogg`	File sonori

Estensione	*Tipo di file*
`.avi` `.mpeg.` `.flv`	File video

- Utilizzando i **magic number**: in questo caso sono utilizzati i primi byte per l'identificazione. Per esempio negli script (file contenenti istruzioni per una shell) dei sistemi Linux vengono utilizzati `#!` (shabang).
- Utilizzando descrizioni esplicite (**metadati** = informazioni che descrivono i dati): un esempio di questa modalità sono i cosiddetti tipi **MIME** (Multipurpose Internet Mail Extension). Le informazioni trasmesse in una rete sono precedute da una stringa di testo contenente il MIME content-type. Per esempio una stringa del tipo `Content-type: text/plain` identifica il file come testo ASCII puro.

In molti casi le specifiche di formato, il modo con cui sono stati codificati i dati, sono pubblicate ma esistono casi in cui il formato del file è considerato segreto industriale. In conseguenza si parla di **formato aperto** quando sono note le specifiche e di **formato chiuso** quando invece non sono note. Nel caso si tratti di un formato aperto non è indispensabile avere l'applicativo che ha generato il file per poterlo rileggere: è sempre possibile sviluppare un nuovo software che sia in grado di leggere il file.

Se si tratta di formato chiuso il file può essere aperto solo dall'applicazione che lo ha generato limitando la libertà dell'utente. Se si tratta di formati molto utilizzati (è il caso, per esempio, del formato .doc di Word) e si vuole leggere il file anche con software diversi rispetto a quelli con cui sono stati generati, si possono usare (sono le tecniche adottate dagli sviluppatori di OpenOffice) tecniche di *reverse engineering*. Si effettuano tante prove esaminando ogni volta il file ottenuto e si cerca di capire la codifica dei dati. Il reverse engineering fornisce risultati sempre più affidabili quante più sono le prove e le verifiche effettuate.

Multimedia: grafica e formati grafici

Una immagine è visualizzata in un computer utilizzando gli elementi luminosi del video ma il problema principale riguarda come rappresentare, per mezzo di stringhe binarie, le proprietà dell'immagine in modo da consentire ad un software di illuminare in maniera opportuna gli elementi luminosi.

La rappresentazione di una immagine in un computer può avvenire in due modi, ovvero: ci sono due tipi di computer grafica.

- **Grafica bitmap** (grafica raster). Una immagine visualizzata sul video di un computer è formata da un rettangolo di pixel (le unità che formano l'immagine sul video), ognuno dei quali di un determinato colore. Per ogni pixel devono essere conservate informazioni, su colore e luminosità, di ognuno dei tre colori principali che compongono un video RGB; informazioni che occupano un byte ciascuno. In definitiva, per esempio, una immagine di 800 x 600 pixel, avrà bisogno di uno spazio di 800 x 600 x 3 = 1.440.000 byte, ovvero circa 1,37 Mb. Questa sarà l'occupazione di memoria richiesta per la visualizzazione dell'immagine. L'immagine ha la stessa struttura del video quindi la visualizzazione è diretta. Ingrandire l'immagine, in questo tipo di grafica, vuol dire aumentare la dimensione del singolo pixel quindi, per esempio, quello che prima era un pixel dello schermo diventa un quadratino di 2 x 2. Se l'ingrandimento è alto, l'effetto prodotto diventa poco piacevole.

- **Grafica vettoriale** (grafica ad oggetti). Il disegno non è memorizzato come insieme di punti ma con formule matematiche che descrivono l'oggetto. Gli oggetti grafici rappresentabili sono punti, linee, curve. Ogni oggetto del disegno viene memorizzato, dal programma che permette di generare il grafico o lo visualizza, assieme agli altri oggetti dell'immagine, descritto da opportune formule matematiche. Una immagine vettoriale si può ingrandire, ruotare, senza perdere in qualità e, in genere, richiede meno spazio in memoria rispetto all'equivalente bitmap.

Una immagine per la quantità di informazioni che deve conservare, occupa molto spazio e questo comporta problemi di occupazione nelle memorie di massa, ma anche, per esempio, ritardi nella trasmissione dell'immagine in rete. Per ottimizzare lo spazio occupato e la velocità di trasmissione sono stati studiati degli algoritmi di compressione che consentono di ridurre le dimensioni del file grafico.

In un grafico ci sono delle zone contigue dello stesso colore e, in questo caso, si può conservare l'informazione di un pixel e della quantità dei pixel con lo stesso colore. In questo modo lo spazio necessario per conservare le informazioni sull'immagine diminuisce; tanto più quanto nell'immagine sono presenti aree di colore uniforme. Ulteriore risparmio di spazio si può ottenere considerando le caratteristiche della visione: l'occhio umano non percepisce alcune differenze di colore e il cervello percepisce meglio i contorni di un oggetto che sono quelli occorrenti, più dell'interno della figura, per completarla. In questo modo un singolo pixel può rappresentare quelli che gli sono più vicini, anche se il colore è leggermente diverso: in riproduzione, l'occhio umano difficilmente percepirà differenze rispetto all'immagine originale. Utilizzando le caratteristiche della visione e del cervello umano, può quindi essere evitata la conservazione di alcuni dati sulla figura e se ne possono ridurre le dimensioni, quando si conservano i dati dell'immagine in un file.

Il modo con cui le informazioni delle immagini vengono scritte in un file, viene chiamato *formato grafico*.

Nella grafica bitmap i formati grafici si dividono in due categorie: formati **lossy** (con perdita di qualità) e **lossless** (senza perdita di qualità). Nei formati con perdita di qualità la compressione è ottenuta a scapito della non memorizzazione di alcune caratteristiche dell'immagine. I formati senza perdita di qualità a costo di maggiore occupazione di spazio conservano tutte le caratteristiche dell'immagine. Quando una immagine deve essere ancora modificata è conveniente scegliere, per conservarla, un formato non distruttivo. Se, invece, l'immagine deve essere visualizzata conviene adottare un formato distruttivo in modo da salvaguardare lo spazio occupato e caricarla in memoria più rapidamente.

- Formato **GIF** (Graphics Interchange Format): lossless. Nato per la trasmissione di immagini sulla rete Internet. Supporta immagini con al massimo 256 colori, uno dei colori può essere trasparente in modo da visualizzare cosa c'è *sotto* l'immagine, nella versione GIF89a può includere nello stesso file più immagini da mostrare in sequenza (GIF animate), supporta l'interlacciamento cioè la possibilità di visualizzare l'immagine sempre più dettagliata mano a mano che viene caricata.
- Formato **PNG** (Portable Network Graphics): lossless. Con le stesse caratteristiche del formato GIF, fu sviluppato quando l'algoritmo di compressione di quel formato fu brevettato e si richiedeva il pagamento di una royalty. Supporta più livelli di trasparenza.
- Formato **JPEG** (Joint Photographer's Experts Group): lossy. Utilizza degli algoritmi che forniscono ottimi risultati per immagini contenenti un numero elevato di colori. Si presta, quindi,

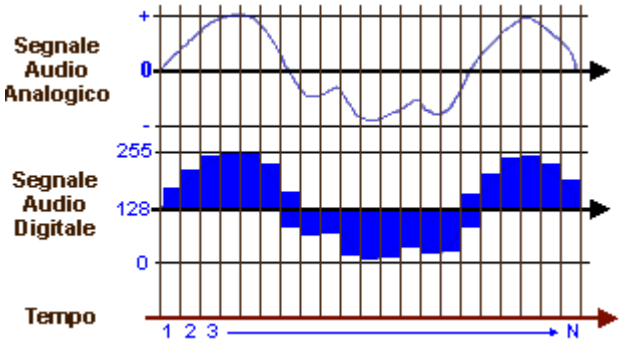
per la memorizzazione di foto e disegni con sfumature. Per grafica con pochi colori fornisce prestazioni inferiori rispetto agli altri formati.

Per quanto riguarda la grafica vettoriale, si può fare cenno a due formati molto comuni:

- Formato **PS** (PostScript). È un file di testo che contiene un programma, che viene interpretato dal processore di una stampante, e che descrive il formato grafico della pagina da stampare. In questo modo, per esempio, anche un file che contiene solo testo, può essere visualizzato o stampato esattamente come generato, in maniera indipendente dalla presenza di caratteri particolari presenti nel computer in cui è stato generato il file, ma non presenti nel computer in cui viene effettuata la visualizzazione o la stampa.
- Formato **PDF** (Portable Document Format). Derivato dal PostScript, a differenza di quest'ultimo non è un programma, ma una sequenza di oggetti grafici ottenuti interpretando un file PostScript. È il risultato dell'interpretazione.

Multimedia: suono e formati sonori

Qualsiasi suono percepito dall'orecchio umano è definibile per mezzo di un onda che colpisce il timpano e viene percepito dal cervello, appunto, come suono.



Per registrare nella memoria di un computer un suono (digitalizzare un suono ovvero trovare un modo come scrivere, utilizzando sequenze binarie, le caratteristiche di un suono), è necessario effettuare un **campionamento**: processo attraverso il quale l'onda viene scomposta in tanti pezzi ad ognuno dei quali viene dato un valore. Maggiore sarà il numero degli intervalli in cui si divide l'onda sonora (i campioni), maggiore sarà la fedeltà ma anche maggiore sarà la dimensione del file che dovrà conservare le informazioni.

Se, per esempio, si utilizza 1 byte per registrare un campione e si divide un secondo in 22.050 campioni (qualità media. La qualità di un CD è 44.100), per registrare 1 minuto di sonoro occorrono 22.050 x 60 = 1.323.000 byte cioè circa 1,26 Mb di spazio.

Per aumentare la qualità del suono registrato si può intervenire su 3 fattori: il maggior numero di campioni, la quantità di bit per rappresentare il campione e, si pensi al suono stereo, i canali. Nel formato **WAV** viene registrato un suono esattamente come descritto. Le dimensioni dei file wave non sono tali però da essere facilmente elaborati e, soprattutto, trasmessi in una rete.

Per ridurre le dimensioni di un file sonoro si possono utilizzare, in ragione del risultato che si vuole raggiungere, strategie diverse che danno origine a formati di file, di cui i più comuni sono:

- Formato **MIDI**. Un file MID contiene delle istruzioni rivolte alla scheda sonora di un computer per produrre un determinato suono, magari ad imitazione di uno strumento musicale. I file MID sono enormemente più piccoli dei WAV, ma la qualità della loro riproduzione cambia in funzione delle potenzialità della scheda sonora. In generale un file MID può essere inviato ad un qualsiasi dispositivo che riproduce suoni: una scheda sonora ma, anche, una tastiera di uno

strumento elettronico.

- Formato **MP3**. Prodotto dal Motion Picture Experts Group, un gruppo di persone, riunito con lo scopo di studiare metodi di compressione dei formati multimediali, che ha proposto una serie di formati che prendono il nome dal gruppo: MPEG 1 (filmati e audio), MPEG 2 (TV digitale), MPEG 4 (applicazioni multimediali). MPEG 1 layer III o, più semplicemente MP3, è il formato con l'algoritmo di compressione più efficiente della famiglia MPEG 1. Nasce con l'obiettivo di ridurre le dimensioni di un file sonoro al fine di poterlo manipolare più agevolmente; per esempio per trasmetterlo in rete. La minore dimensione del file sonoro, rispetto a quella di un file WAV, è ottenuta con un processo di eliminazione della gamma sonora non percepita dall'orecchio umano: i 44.100 campioni di un file sonoro in qualità CD, vista la conformazione dell'orecchio umano, non vengono tutti percepiti è quindi inutile conservarli tutti. L'elevata resa sonora (praticamente l'esecuzione di un MP3 è indistinguibile da un WAV) e le ridotte dimensioni (rapporto 10:1) hanno decretato il successo di questo formato.

Multimedia: grafica in movimento e suono

Anche per poter conservare in un computer informazioni per riprodurre un video è necessaria la presenza di un **codec** (codificatore/decodificatore). Un codec è un programma o un dispositivo in grado di digitalizzare un segnale video/audio per poterlo conservare nella memoria di un computer e, al contrario, di riprodurlo. Il codec si occupa anche della compressione/decompressione del file e si può trovare anche integrato in alcuni componenti hardware: per esempio nei lettori DVD.

Come noto un filmato è costituito da molte immagini (fotogrammi o *frame*) visualizzate in sequenza. Per ridurre la dimensione del file che contiene il filmato, è necessario ridurre le informazioni conservate. In particolare in un fotogramma pixel vicini con intensità di colore simile vengono eliminati. Allo stesso modo come i dati ripetuti in fotogrammi successivi: in generale infatti, tranne per scene diverse e molto movimentate, alcuni dati sarebbero ripetuti e, quindi, si possono conservare una volta sola. Inoltre anche in questo caso ci sono dati del filmato non percepibili dall'occhio umano. Per quanto riguarda il sonoro, inoltre, si può adottare (come nel caso del DivX o XviD) il formato MP3.

Il formato multimediale con più diffusione oggi, il formato **DivX**, nasce come progetto della Microsoft con il nome Div (Digital Internet Video). Utilizzava il codec per il formato MPEG 4, ma fu abbandonato per i risultati scadenti. Il progetto fu ripreso da un programmatore indipendente francese che estrasse il sorgente dal codec e creò DivX;-) includendo l'emoticon sorridente come riferimento sarcastico al fallimento del progetto precedente. Il formato ebbe un enorme successo e il suo creatore decise di svilupparne una copia commerciale, e legale, riscrivendo gli algoritmi di compressione e il codec che non fa più riferimento ad MPEG 4.

Il percorso di DivX da versione fuorilegge (codice crackato da Div) a versione open source a versione commerciale, ha generato una serie di altri progetti ad opera di programmatori delusi. Il progetto considerato più direttamente concorrente è **XviD** (si fa notare che il nome, con evidente riferimento, è DivX scritto al contrario).

Anche DivX o XviD sono codifiche di tipo lossy: un filmato compresso e, poi, decompresso non risulta identico all'originale anche se le differenze possono essere rese quasi impercettibili.

Riferimenti

Per la scrittura di questi appunti sono state consultate le seguenti fonti:

- *Architettura dei computer* di Andrew S. Tanenbaum. Testo utilizzando universalmente e punto di riferimento assoluto sul mondo del computer.
- *http://www.wikipedia.org* anche nella edizione italiana *it.wikipedia.org*. L'enciclopedia libera in rete. Risorsa globale diventata ormai un punto di riferimento.