Rappresentazione dei dati

Andrea Passerini
passerini@disi.unitn.it

Informatica

Unità di misura per l'informatica

- Un *bit* (b) rappresenta una cifra binaria. E' l'unità minima di informazione.
- Un *Byte* (B) è costituito da 8 bit. Permette di codificare 256 entità di informazione distinte (e.g. caratteri dell'alfabeto, segni di interpunzione)
- Una *parola* (*word*) rappresenta l'insieme di Byte che possono essere trattati da un elaboratore in un'operazione. Il numero di Byte di una parola dipende dall'elaboratore, ad esempio 2, 3, 4, 8 corrispondenti a 16, 24, 32 o 64 bit.

Multipli del Byte

- Kilobyte (KB) = 2^{10} = 1024 Byte (e.g. un file di testo da 300KB)
- Megabyte (MB) = 1024KB $\approx$ 1 milione di Byte (e.g. un'immagine di 30MB)
- Gigabyte (GB) = 1024MB $\approx$ 1 miliardo di Byte (e.g. un hard disk da 80GB)
- Terabyte (TB) = 1024GB $\approx$ 1000 miliardi di Byte (e.g. un archivio da 20TB)

Bit rate

- Esprime la velocità di trasferimento dati in bit per secondo (bps).
- Multipli
  - Kilobit per secondo (Kbps) = 1000bps (e.g. un modem a 56Kbps)
  - Megabit per secondo (Mbps) = 1000Kbps (e.g. una ADSL a 4Mbps)

Nota sul bit rate

- La dimensione dei file viene generalmente espressa in multipli **binari** del **Byte** (e.g. 1MB = 2^{10} KByte = 2^{20} Byte = 1,048,576 Byte)
- Il bit rate viene generalmente espresso in multipli **decimali** del **bit** (e.g. 1Mbps = 1,000,000 bps)
- Quindi per scaricare un file da 4MB con un'ADSL a 4Mbps sono necessari (nella situazione ottimale):

$$\frac{4 \times 2^{20} \times 8}{4 \times 1000000} = 8.388608$$

  ossia circa 8 secondi e mezzo.
- Questo non considerando la compressione e il tempo necessario a trasmettere l'intestazione dei pacchetti

Nota sulla dimensione di HD

- La dimensione dei file viene generalmente espressa in multipli **binari** del **Byte** (e.g. 1MB = 2^{10} KByte = 2^{20} Byte = 1,048,576 Byte)
- La dimensione degli HD viene generalmente espressa in multipli **decimali** del **Byte** (e.g. 1GB = 10^9 Byte)
- Quindi un HD da 250GB contiene file per:

$$\frac{250 \times 10^9}{250 \times 2^{30}} \approx 233GB$$

Hertz (Hz)

- Grandezza per misurare la rapidità (frequenza) dei dispositivi digitali
- Il nome deriva dal fisico Heinrich Rudolf Hertz
- 1 Hz corrisponde ad un ciclo o una oscillazione al secondo
- Multipli
  - Kilohertz (KHz) = 1000 Hz (si usa per misurare la frequenza di refresh dello schermo)
  - Megahertz (MHz) = 1000 KHz
  - Gigahertz (GHz) = 1000 MHz (si usa per misurare la frequenza di clock dei calcolatori)

Hertz e clock

- Il clock è un dispositivo che funziona come un metronomo sincronizzando tutte le operazioni dei dispositivi digitali
- Nota:
  - L'unità di misura Hz si può usare come indicatore relativo della velocità di elaborazione di un computer
  - ovvero si può usare per paragonare due processori con la stessa architettura (e.g. un Pentium 4 a 2.8GHz contro un Pentium 4 a 3.2GHz)
  - NON si possono però fare paragoni fra processori diversi (e.g. RISC Motorola contro Pentium)

Dots Per Inch (DPI) ossia punti per pollice

- Grandezza per misurare la densità di punti o *definizione* o *risoluzione*.
- Un pollice quadrato corrisponde ad un area di 2.54 cm^2
- Stampanti e schermi usano matrici di punti (*pixel*) per rappresentare immagini 2D.
- Maggiore è il numero di punti nell'unità di area maggiore è l'accuratezza con la quale si definisce un'immagine o un testo
- Esempio:
  - 300 dpi per le stampanti laser
  - 1200 dpi per gli scanner
- La risoluzione di uno schermo si indica riportando il numero di pixel visualizzati sul lato orizzontale e su quello verticale (e.g. 1280x800 o 860x640)

Codifica dei dati

Codifica binaria

- Qualsiasi informazione deve essere codificata in binario per poter essere trattata da un calcolatore.
- Vedremo come vengono codificati:
  - Numeri interi
  - Numeri “reali” (*real*)
  - Caratteri

Codifica dei numeri

Numero di bit fissato

- Il calcolatore può fare operazioni su due numeri solo se sono codificati con lo stesso numero di bit.
- Si fissa il numero di bit k con cui si rappresenta un certo insieme di numeri (e.g. interi a 8 bit)
- Ogni numero di tale insieme deve essere rappresentato con k bit, eventualmente aggiungendo zeri a sinistra
- Esempio per interi positivi a $k = 8$ bit

$$(54)_{10} = (00110110)_2$$

Rappresentazione di interi

Codifica valore assoluto con segno

- Si riserva il bit più significativo (quello più a sinistra) al segno (0=positivo, 1= negativo).
- Si codifica con i restanti bit il modulo del numero.
- Esempio con $k = 8$ bit:

$$(+54)_{10} = (00110110)_2$$
$$(-54)_{10} = (10110110)_2$$

Problema

- Problema: il *bit di segno* deve essere trattato in maniera diversa rispetto agli altri bit (complica i circuiti per somma e sottrazione).
- Si utilizza la codifica in *complemento a 2* che permette di realizzare sottrazioni con complementazioni e somme (non la vedremo).

Overflow

- Si verifica quando la somma di due numeri è al di fuori del rango (*range*) di valori permessi nella rappresentazione scelta per i numeri.
- Ad esempio il rango dei numeri interi rappresentati in complemento a 2 con k bit è $[-2^{k-1}, 2^{k-1} - 1]$
- Il risultato dell'operazione non è corretto nel caso si verifichi un overflow.
- Gli elaboratori elettronici verificano la condizione di overflow quando effettuano l'addizione binaria e la segnalano mettendo ad 1 uno speciale *bit di overflow*.

Traslazione logica (*shift logico*)

- Consiste nello spostare a destra (*shift logico a destra*) o a sinistra (*shift logico a sinistra*) i bit di un numero binario.
- Nella traslazione a destra, il bit meno significativo viene perso, quello più significativo viene posto a 0.
- Un numero binario traslato a destra viene diviso per 2.
- Nella traslazione a sinistra, il bit più significativo viene perso, quello meno significativo viene posto a 0.
- Un numero binario traslato a sinistra viene moltiplicato per 2.

Operazioni eseguite in termini di altre operazioni

- Con la codifica in complemento a 2, la sottrazione si realizza con una complementazione (operazione semplicissima) ed un'addizione.
- Si può realizzare un unico circuito che effettui somme e sottrazioni con notevoli risparmi di costi.
- Una moltiplicazione può essere realizzata tramite una sequenza di addizioni e di traslazioni a sinistra.
- Una divisione può essere realizzata tramite una sequenza di sottrazioni e di traslazioni a destra.
- Le operazioni più semplici vengono eseguite da appositi circuiti (a livello *hardware*).
- Operazioni più complesse sono eseguite in termini di esecuzione di altre operazioni più semplici sotto il controllo di programmi (a livello *software*).

Rappresentazione di numeri “reali” (*real*)

Rappresentazione in virgola fissa

- Si stabilisce un numero di bit k_1 da assegnare alla parte intera ed un numero di bit k_2 da assegnare alla parte frazionaria.
- Ad esempio per numeri a 32 bit se ne assegnano 16 alla parte intera e 16 alla parte frazionaria.
- Adatta solo a casi particolari in cui l’intervallo di valori da rappresentare è noto a priori.
- Inadatta nella maggior parte delle applicazioni scientifiche.

Rappresentazione in virgola mobile (*floating point*)

- Si parte dalla rappresentazione scientifica in cui il numero è il prodotto di due parti: una parte frazionaria ed un fattore di scala, che è una potenza del 10.
- Ad esempio il numero 23.5 può essere rappresentato come:

$$23.5 \times 10^0 \qquad 2.35 \times 10^1$$
$$235 \times 10^{-1} \qquad 0.235 \times 10^2$$
$$\ldots \qquad \ldots$$

- Rappresentazione scientifica normalizzata: la parte frazionaria ha la cifra più a sinistra diversa da 0 e subito seguita dal punto decimale (e.g. 2.35×10^1).
- Nella rappresentazione floating point si rappresenta il numero come una coppia: la *mantissa* corrispondente alla parte frazionaria, la *caratteristica* (o *esponente*) che corrisponde all'esponente del fattore di scala.

Rappresentazione floating point binaria

- Nel caso binario mantissa ed esponente sono rappresentati in binario, ed il fattore di scala è una potenza del 2.
- Esempi:

Numero	Rapp. norm.	Mantissa	Esponente
+101010.0	$+1.010100 \times 10^{101}$	+1.010100	+101
+0.000110	$+1.100000 \times 10^{-100}$	+1.100000	-100
-110.1100	-1.101100×10^{10}	-1.101100	+010

Codifica di caratteri

- Repertorio: insieme di caratteri considerati, definito mediante i nomi dei caratteri e magari una loro rappresentazione visiva.
- Numero di codice: tabella che associa ad ogni carattere un numero da un dato insieme di numeri naturali.
- Codifica: un metodo per associare a ciascun numero di codice una sequenza di bit.
- Nel caso più semplice ogni carattere ha un numero tra 0 e 127 e la codifica è semplicemente la codifica binaria del numero in 7 bit.

Codice ASCII

- Acronimo di *American Standard Code for Information Interchange*.
- 7 bit per carattere, si possono rappresentare $2^7 = 128$ caratteri distinti. I codici sono tipicamente scritti in notazione esadecimale.
- I codici da 0 a 1F sono usati per *caratteri di controllo*
- I codici da 20 a 7E sono usati per *caratteri stampabili*
- Ordine alfabetico: cifre 0-9 prima dei caratteri alfabetici, maiuscole prima delle minuscole (si rifletterà nell'ordinamento lessicografico delle stringhe).

Tabella dei codici 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	&#127;	DEL

Source: www.LookupTables.com

Limitazioni del codice ASCII

- I caratteri internazionali di numerose lingue europee (quali ad esempio è,ù,ç,å,æ,ü,ø) non sono rappresentabili
- nessuno dell'elevatissimo numero di simboli delle lingue asiatiche è rappresentabile
- La standardizzazione è importante: nella trasmissione e memorizzazione elettronica i caratteri sono rappresentati da insiemi di Byte ed è importante che il "trasmettitore" ed il "ricevente" adottino le stesse convenzioni!
- In assenza di opportune convenzioni, testi generati su un dato sistema possono risultare corrotti se visualizzati in un sistema diverso (capita facilmente con la posta elettronica).

ISO Latin-1 (ISO 8859-1)

- Il repertorio contiene il repertorio ASCII come sottoinsieme e i codici per questi caratteri sono identici a quelli ASCII
- Il codice usa 8 bit: 256 caratteri distinti
- Contiene vari simboli usati dai linguaggi dell'Europa occidentale (Italiano, Francese, Spagnolo, Tedesco, Danese, etc.):

¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®		°	±	²	³	´	µ	¶	·	¸
¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð
Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â	ã	ä	å	æ	ç	è
é	ê	ë	ì	í	î	ï	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

La famiglia ISO 8859

- La famiglia contiene ben 15 alfabeti standard, tra cui ad esempio:

  8859-2 (Latin-2) lingue dell'Europa centrale e orientale
  8859-5 (Latin/Cyrillic) lingue Slave
  8859-7 (Latin/Greek) Greco moderno
  8859-9 (Latin-5) Turco
  8859-10 (Latin-6) lingue nordiche (Islandese)
  8859-15 (Latin-9) Latin-1 con l'Euro

ISO 10646 e UNICODE

- ISO 10646 è uno standard internazionale che definisce lo UCS (Universal Character Set)
- Il repertorio è molto ampio e contiene decine di migliaia di caratteri già definiti, con spazio per espansioni future
- Estende ISO Latin-1 nello stesso senso in cui ISO Latin-1 estende ASCII
- UNICODE è uno standard definito dal consorzio UNICODE (fondato nel ’92) e definisce un repertorio e una codifica pienamente compatibili con ISO 10646
- UNICODE implementa solo una parte di ISO 10646

UNICODE

- Il vantaggio di UNICODE è di definire un insieme di caratteri adatti a trattare tutti i linguaggi, mentre la famiglia ISO 8859 definisce sottoinsiemi di caratteri adatti a trattare solo alcuni linguaggi alla volta
- Mentre con altri standard lo stesso carattere può avere codifiche diverse (a seconda dell'alfabeto usato), in UNICODE ogni carattere ha una codifica unica
- Ha una grande diffusione industriale (Apple, HP, IBM, Microsoft, Oracle, Sun, etc.)
- E' supportato da vari sistemi operativi ed internet browser più recenti.

UNICODE

- Il codice usa 16 bit (fino a 64K caratteri)
- I primi 256 codici coincidono con quelli di ISO Latin-1
- Esempio di codifica per caratteri armeni: