CALCOLATORI

Codifica del testo

Giovanni Iacca
giovanni.iacca@unitn.it

Lezione basata su materiale preparato dal Prof. Luca Abeni

UNIVERSITÀ DEGLI STUDI DI TRENTO
Dipartimento di Ingegneria
e Scienza dell'Informazione

Codifica del testo

- Come rappresentare il testo tramite sequenze di 0 e 1?
  - Testo: sequenza di caratteri
  - Quindi, il problema è rappresentare caratteri come sequenze di 0 e 1...
- Ricorda: n bit possono codificare 2^n simboli diversi
- Quanti possibili caratteri si devono rappresentare?
- Alfabeto “anglosassone”: 7 bit ($2^7 = 128$ diversi caratteri)
  - Lettere maiuscole e minuscole, numeri, punteggiatura, ecc.
- ASCII (American Standard Code for Information Interchange)

Lo standard ASCII

- Specifica come codificare lettere, numeri e punteggiatura su 7 bit
  - Ma un byte è composto da 8 bit...
  - Bit più significativo sempre a 0
- Cosa fare per caratteri accentati o “strani”?
  - Ci sono altre 128 combinazioni di bit disponibili...
- Extended ASCII: usa 8 bit per codificare caratteri addizionali
  - Non esiste un unico standard “esteso”...
  - Varie estensioni per supportare vari alfabeti (europa dell’est, ovest, ecc.)

Lo standard ASCII

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[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	[NEGATIVE 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	[ENG OF 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	[SUBSTITUTE]	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	[DEL]

Esempio

- Codifichiamo la parola “Ciao”
  - C è codificata come $67 = 0x43 = 01000011$
  - i è codificata come $105 = 0x69 = 01101001$
  - a è codificata come $97 = 0x61 = 01100001$
  - o è codificata come $111 = 0x6F = 01101111$

01000011	01101001	01100001	01101111
C	i	a	o

ASCII Esteso

- Un byte con valore < 128 (bit più significativo a 0) si interpreta in modo univoco come carattere
  - Esempio: 01000001 è sempre "A"
- L'interpretazione di byte col bit più significativo ad 1 non è univoca
  - ISO 8859-1 (Latin1): caratteri dell'Europa Occidentale (lettere accentate, ecc.)
  - ISO 8859-2: caratteri dell'Europa Orientale
  - ISO 8859-5: per i caratteri cirillici
  - ...
- Esempio: il valore 224 è "à" per ISO 8859-1, "ŕ" per ISO 8859-2, ecc..

Problemi con ASCII Esteso

- ASCII esteso: codifica non univoca di caratteri “non standard”
- Problemi nella condivisione di documenti
  - Se utilizzo una “è” in un documento testo e lo trasmetto ad altre persone...
  - ... devo assicurarmi che i computer delle altre persone utilizzino ISO 8859-1 come il mio computer...
  - ... altrimenti strani simboli possono essere visualizzati al posto della mia “è”
- E cosa dire degli alfabeti che non usano l’alfabeto anglosassone (Cirillico, Cinese, Giapponese, Arabo, ecc.)?

Altri standard di codifica dei caratteri

- Codifica univoca di tutti i possibili caratteri: 8 bit non bastano!!!
- Unicode: fino a 2^{32} simboli!!!
  - Possono servire 32 bit (4 byte) per carattere...
- I simboli unicode possono essere codificati in vari modi
  - UTF-32: ogni simbolo è composto da 32 bit
  - UTF-16: ogni simbolo è composto da 16 o più bit (simboli a lunghezza variabile)
  - UTF-8: ogni simbolo è composto da 8 o più bit
    - Compatibile con ASCII