

Soluzioni più moderne

- Anche i codici a 8 bit (255 caratteri) erano insufficienti a rappresentare tutti i caratteri necessari per uso generale
 - anche solo per le lingue “occidentali”!
- Soluzione: definire **una codifica diversa per ogni lingua...**
 - ISO-8859-xx
 - Windows CodePage (CP-xxx)
- ... **mantenendo la compatibilità ASCII**

Codifica ISO-8859-xx

- Sviluppata da **ISO** (**I**nternational **O**rganization for **S**tandardization)
- Consta di 16 diverse codifiche a 8 bit, ciascuna delle quali include ASCII nei codici “bassi” e altri caratteri per un totale di 191 caratteri e 32 codici di controllo (rimangono alcuni codici liberi)
- Nota anche come ISO-Latin-xx perché codifica tutti i caratteri “latini”

ISO-8859-1

- ISO-8859-1 è stato ampiamente adottato, e può ormai essere visto come il sostituto *de facto* dello standard ASCII.
- ISO 8859-1 è adatto per i linguaggi seguenti:
 - afrikaans, basco, catalano, danese, inglese, faeroese, finlandese, francese, galiziano, irlandese, islandese, **italiano**, norvegese, olandese, portoghese, scozzese, spagnolo, svedese e tedesco.

Elenco ISO-8859-xx

Codifica	Lingue
ISO 8859-1	lingue europee occidentali (Latin-1)
ISO 8859-2	lingue europee orientali (Latin-2)
ISO 8859-3	lingue europee sudoccidentali e varie (Latin-3)
ISO 8859-4	lingue scandinave/baltiche (Latin-4)
ISO 8859-5	alfabeto romano/cirillico
ISO 8859-6	alfabeto romano/arabo
ISO 8859-7	alfabeto romano/greco
ISO 8859-8	alfabeto romano/ebraico
ISO 8859-9	latin-1 modificato per il turco (Latin-5)
ISO 8859-10	lingue lappone/nordiche/eschimesi (Latin-6)
ISO 8859-11	tailandese
ISO 8859-13	lingue baltiche (Latin-7)
ISO 8859-14	celtico (Latin-8)
ISO 8859-15	lingue europee occidentali (Latin-9)
ISO 8859-16	rumeno (Latin-10)

ISO-8859-15

- ISO-8859-15 (ISO-Latin-9) è un aggiornamento di ISO-8859-1 (Latin-1) che aggiunge:
 - il carattere € (per ovvie ragioni assente nello standard originale!)
 - alcuni caratteri rari del Francese e del Finlandese (usati anche in altre lingue)
- Rimuove inoltre alcuni codici non usati

ISO-8859-15

- Con le aggiunte, ISO-8859-15 supporta le lingue seguenti:
 - albanese, basco, bretone, catalano, danese, olandese, inglese, estone, faroese, finlandese, francese, frisone, galiziano, tedesco, islandese, groenlandese, irlandese, gaelico, italiano, latino, lussemburghese, norvegese, portoghese, reto-romanico, gaelico scozzese, spagnolo, svedese (più alcune altre che hanno una traslitterazione nell'alfabeto latino...)
- Il **“nuovo ASCII”**

ISO-8859-15

Decimale	Hex	Descrizione	Carattere	Decimale	Hex	Descrizione	Carattere
160	A0	NO-BREAK SPACE		208	D0	CAPITAL ICELANDIC LETTER ETH	Ð
161	A1	INVERTED EXCLAMATION MARK	¡	209	D1	CAPITAL LETTER N WITH TILDE	Ñ
162	A2	CENT SIGN	¢	210	D2	CAPITAL LETTER O WITH GRAVE ACCENT	Ò
163	A3	POUND SIGN	£	211	D3	CAPITAL LETTER O WITH ACUTE ACCENT	Ó
164	A4	EURO SIGN	€	212	D4	CAPITAL LETTER O WITH CIRCUMFLEX ACCENT	Ô
165	A5	YEN SIGN	¥	213	D5	CAPITAL LETTER O WITH TILDE	Õ
166	A6	CAPITAL LETTER S WITH CARON	Š	214	D6	CAPITAL LETTER O WITH DIAERESIS	Ö
167	A7	PARAGRAPH SIGN	§	215	D7	MULTIPLICATION SIGN	×
168	A8	SMALL LETTER S WITH CARON	š	216	D8	CAPITAL LETTER O WITH OBLIQUE STROKE	Ø
169	A9	COPYRIGHT SIGN	©	217	D9	CAPITAL LETTER U WITH GRAVE ACCENT	Ù
170	AA	FEMININE ORDINAL INDICATOR	ª	218	DA	CAPITAL LETTER U WITH ACUTE ACCENT	Ú
171	AB	LEFT ANGLE QUOTATION MARK	«	219	DB	CAPITAL LETTER U WITH CIRCUMFLEX ACCENT	Û
172	AC	NOT SIGN	¬	220	DC	CAPITAL LETTER U WITH DIAERESIS	Ü
173	AD	SOFT HYPHEN	-	221	DD	CAPITAL LETTER Y WITH ACUTE ACCENT	Ý
174	AE	REGISTERED TRADE MARK SIGN	®	222	DE	CAPITAL ICELANDIC LETTER THORN	þ
175	AF	MACRON	-	223	DF	SMALL GERMAN LETTER SHARP s	ß
176	B0	DEGREE SIGN, RING ABOVE	°	224	E0	SMALL LETTER a WITH GRAVE ACCENT	à
177	B1	PLUS-MINUS SIGN	±	225	E1	SMALL LETTER a WITH ACUTE ACCENT	á
178	B2	SUPERSCRIPIT TWO	²	226	E2	SMALL LETTER a WITH CIRCUMFLEX ACCENT	â
179	B3	SUPERSCRIPIT THREE	³	227	E3	SMALL LETTER a WITH TILDE	ã
180	B4	CAPITAL LETTER Z WITH CARON	Ž	228	E4	SMALL LETTER a WITH DIAERESIS	ä
181	B5	MICRO SIGN	µ	229	E5	SMALL LETTER a WITH RING ABOVE	å
182	B6	PILCROW SIGN	¶	230	E6	SMALL DIPHTHONG a WITH e	æ
183	B7	MIDDLE DOT	·	231	E7	SMALL LETTER c WITH CEDILLA	ç
184	B8	SMALL LETTER Z WITH CARON	ž	232	E8	SMALL LETTER e WITH GRAVE ACCENT	è
185	B9	SUPERSCRIPIT ONE	¹	233	E9	SMALL LETTER e WITH ACUTE ACCENT	é
186	BA	MASCULINE ORDINAL INDICATOR	º	234	EA	SMALL LETTER e WITH CIRCUMFLEX ACCENT	ê
187	BB	RIGHT ANGLE QUOTATION MARK	»	235	EB	SMALL LETTER e WITH DIAERESIS	ë
188	BC	CAPITAL OE DIGRAPH	Œ	236	EC	SMALL LETTER i WITH GRAVE ACCENT	ì
189	BD	SMALL OE DIGRAPH	œ	237	ED	SMALL LETTER i WITH ACUTE ACCENT	í
190	BE	CAPITAL LETTER Y WITH DIAERESIS	ÿ	238	EE	SMALL LETTER i WITH CIRCUMFLEX ACCENT	î
191	BF	INVERTED QUESTION MARK	¿	239	EF	SMALL LETTER i WITH DIAERESIS	ï
192	C0	CAPITAL LETTER A WITH GRAVE ACCENT	À	240	F0	SMALL ICELANDIC LETTER eth	ð
193	C1	CAPITAL LETTER A WITH ACUTE ACCENT	Á	241	F1	SMALL LETTER n WITH TILDE	ñ
194	C2	CAPITAL LETTER A WITH CIRCUMFLEX ACCENT	Â	242	F2	SMALL LETTER o WITH GRAVE ACCENT	ò
195	C3	CAPITAL LETTER A WITH TILDE	Ã	243	F3	SMALL LETTER o WITH ACUTE ACCENT	ó
196	C4	CAPITAL LETTER A WITH DIAERESIS	Ä	244	F4	SMALL LETTER o WITH CIRCUMFLEX ACCENT	ô
197	C5	CAPITAL LETTER A WITH RING ABOVE	Å	245	F5	SMALL LETTER o WITH TILDE	õ
198	C6	CAPITAL DIPHTHONG A WITH E	Æ	246	F6	SMALL LETTER o WITH DIAERESIS	ö
199	C7	CAPITAL LETTER C WITH CEDILLA	Ç	247	F7	DIVISION SIGN	÷
200	C8	CAPITAL LETTER E WITH GRAVE ACCENT	È	248	F8	SMALL LETTER o WITH OBLIQUE STROKE	ø
201	C9	CAPITAL LETTER E WITH ACUTE ACCENT	É	249	F9	SMALL LETTER u WITH GRAVE ACCENT	ù
202	CA	CAPITAL LETTER E WITH CIRCUMFLEX ACCENT	Ê	250	FA	SMALL LETTER u WITH ACUTE ACCENT	ú
203	CB	CAPITAL LETTER E WITH DIAERESIS	Ë	251	FB	SMALL LETTER u WITH CIRCUMFLEX ACCENT	û
204	CC	CAPITAL LETTER I WITH GRAVE ACCENT	Ì	252	FC	SMALL LETTER u WITH DIAERESIS	ü
205	CD	CAPITAL LETTER I WITH ACUTE ACCENT	Í	253	FD	SMALL LETTER y WITH ACUTE ACCENT	ý
206	CE	CAPITAL LETTER I WITH CIRCUMFLEX ACCENT	Î	254	FE	SMALL ICELANDIC LETTER THORN	þ

CodePage-xx

- Idea simile a quella di ISO-8859-xx, ma
 - sottilmente incompatibile
 - non standardizzata
- Codifica introdotta da IBM e Microsoft
 - ciascuno dei due ha poi continuato a “inventare” nuove codepage secondo necessità
 - Usata nei mainframe IBM, in PC-DOS, MS-DOS, Windows

Alcuni esempi

- Codepage definite da IBM per il PC-DOS (e MS-DOS):

Codepage **Uso**

CP437	Codifica usata nel PC IBM originale (1981)
CP737	Codifica MS-DOS per l'alfabeto greco
CP850	Codifica dell'alfabeto latino e greco + simboli grafici
CP852	Codifica per le lingue del centro europa (polacco, romeno, ceco, slovacco)
CP855	Codifica per l'alfabeto cirillico
CP857	Codifica per l'alfabeto turco
CP858	CP850 + simbolo euro (al posto della i senza puntino)
CP860	Codifica per il Portoghese
CP861	Codifica per l'Islandese e altri linguaggi nordici
CP863	Codifica estesa per il Francese (usato in Canada)
CP865	Codifica per il Danese e il Norvegese (e altri linguaggi nordici escluso l'Islandese)
CP866	Altra codifica per il cirillico
CP869	Altra codifica per il greco

Alcuni esempi

- Codepage definite da Microsoft per Windows (anche note come Windows-xx)

Codepage **Uso**

CP1250	Codifica per le lingue del centro Europa (polacco, romeno, slovacco, ungherese, ...)
CP1251	Codifica per l'alfabeto cirillico
CP1252	Codifica per le lingue dell'Europa occidentale (incluso l'Italiano)
CP1253	Codifica per l'alfabeto greco
CP1254	Codifica per l'alfabeto turco
CP1255	Codifica per l'alfabeto ebraico
CP1256	Codifica per l'alfabeto arabo
CP1257	Codifica per le lingue baltiche
CP1258	Codifica per il Vietnamita

...

Alcuni esempi

- Sulla scia di ISO e CP, altri costruttori e paesi hanno introdotto proprie codifiche “ad hoc”:
 - GB2312 per il cinese
 - KOI-8 per il russo e il bulgaro, KOI-8-U per l'Ucraino
 - Apple con il Macintosh introdusse alcune codifiche proprie
 - ancora una volta, leggermente incompatibili con tutte le altre...

CP437

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NULL 0 0	☉ 263A 1	☽ 263B 2	♥ 2665 3	♦ 2666 4	♣ 2663 5	♠ 2660 6	• 2022 7	♠ 25D8 8	○ 25CB 9	⦿ 25D9 10	♂ 2642 11	♀ 2640 12	♪ 266A 13	♪ 266B 14	* 263C 15
1.	▶ 25BA 16	◀ 25C4 17	↕ 2195 18	!! 203C 19	¶ B6 20	§ A7 21	■ 25AC 22	‡ 21A8 23	↑ 2191 24	↓ 2193 25	→ 2192 26	← 2190 27	↔ 221F 28	↔ 2194 29	▲ 25B2 30	▼ 25BC 31
2.	20 32 0	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3.	30 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	:	;	<	=	>	?
4.	@ 40 64	A 41 65	B 42 66	C 43 67	D 44 68	E 45 69	F 46 70	G 47 71	H 48 72	I 49 73	J 4A 74	K 4B 75	L 4C 76	M 4D 77	N 4E 78	O 4F 79
5.	P 50 80	Q 51 81	R 52 82	S 53 83	T 54 84	U 55 85	V 56 86	W 57 87	X 58 88	Y 59 89	Z 5A 90	[5B 91	\ 5C 92] 5D 93	^ 5E 94	_ 5F 95
6.	` 60 96	a 61 97	b 62 98	c 63 99	d 64 100	e 65 101	f 66 102	g 67 103	h 68 104	i 69 105	j 6A 106	k 6B 107	l 6C 108	m 6D 109	n 6E 110	o 6F 111
7.	p 70 112	q 71 113	r 72 114	s 73 115	t 74 116	u 75 117	v 76 118	w 77 119	x 78 120	y 79 121	z 7A 122	{ 7B 123	 7C 124	}	~ 7E 126	△ 2302 127
8.	Ç C7 128	ü FC 129	é E9 130	â E2 131	ä E4 132	à E0 133	å E5 134	ç E7 135	ê EA 136	ë EB 137	è E8 138	ï EF 139	î EE 140	ì EC 141	À C4 142	Å C5 143
9.	É C9 144	æ E6 145	Æ C6 146	ô F4 147	ö F6 148	ò F2 149	û FB 150	ù F9 151	ÿ FF 152	Ö D6 153	Ü DC 154	¢ A2 155	£ A3 156	¥ A5 157	₤ 20A7 158	f 192 159
A.	á E1 160	í ED 161	ó F3 162	ú FA 163	ñ F1 164	Ñ D1 165	ª AA 166	º BA 167	¿ BF 168	¬ 2310 169	¬ AC 170	½ BD 171	¼ BC 172	ı A1 173	« AB 174	» BB 175
B.	⦿ 2591 176	⦿ 2592 177	⦿ 2593 178	 2502 179	 2524 180	 2561 181	 2562 182	 2556 183	 2555 184	 2563 185	 2551 186	 2557 187	 255D 188	 255C 189	 255B 190	 2510 191
C.	L 2514 192	⊥ 2534 193	⊥ 252C 194	⊥ 251C 195	— 2500 196	⊥ 253C 197	⊥ 255E 198	⊥ 255F 199	⊥ 255A 200	⊥ 2554 201	⊥ 2569 202	⊥ 2566 203	⊥ 2560 204	= 2550 205	⊥ 256C 206	⊥ 2567 207
D.	⊥ 2568 208	⊥ 2564 209	⊥ 2565 210	⊥ 2559 211	⊥ 2558 212	⊥ 2552 213	⊥ 2553 214	⊥ 256B 215	⊥ 256A 216	⊥ 2518 217	⊥ 250C 218	■ 2588 219	■ 2584 220	■ 258C 221	■ 2590 222	■ 2580 223
E.	α 3B1 224	β DF 225	Γ 393 226	π 3C0 227	Σ 3A3 228	σ 3C3 229	μ B5 230	τ 3C4 231	φ 3A6 232	θ 398 233	Ω 3A9 234	δ 3B4 235	∞ 221E 236	φ 3C6 237	ε 3B5 238	η 2229 239
F.	≡ 2261 240	± B1 241	≥ 2265 242	≤ 2264 243	∫ 2320 244	∫ 2321 245	÷ F7 246	≈ 2248 247	° B0 248	· 2219 249	· B7 250	√ 221A 251	ⁿ 207F 252	² B2 253	■ 25A0 254	A0 255

Il problema con le Codepage

- Spesso la definizione segue criteri più commerciali che linguistici
 - La CP850 venne definita in quel modo (faccine, notine, semi delle carte da gioco), per dimostrare che il PC poteva far girare un programma simile a una nota macchina di word processing Wang (!)
 - CP diverse sono state definite per paesi con alfabeti simili, ma economie diverse
 - versioni del S.O. per il paese più ricco venivano fatte pagare più care rispetto ai “vicini poveri”

Il problema con le Codepage

- A volte CP diverse sono del tutto identiche, a parte uno o due caratteri
 - È estremamente difficile riconoscere a prima vista quale CP è stata usata per codificare un certo testo
- Più CP possono essere usate per la stessa lingua
 - incompatibilità anche all'interno dello stesso paese e fra testi prodotti sulla stessa macchina

Problemi comuni a ISO e CP

- È possibile codificare al più 256 caratteri distinti
 - mantenendo la compatibilità ASCII, ancora meno!
 - lingue ideografiche (cinese, giapponese, coreano) o complesse sono escluse
- Lo **stesso codice** numerico può corrispondere a **caratteri diversi** in codifiche diverse
- Lo **stesso carattere** può corrispondere a **codici diversi** in codifiche diverse

Approfondimenti

- Il sito <http://homepages.cwi.nl/~dik/english/codes/stand.html> contiene un elenco di codifiche “storiche” e delle loro successive versioni
- Il sito <http://www.iana.org/assignments/character-sets> elenca i 250 set di caratteri ufficialmente riconosciuti da IANA (Internet Assigned Numbers Authority)
- Il sito <http://www.i18nguy.com/unicode/codepages.html> contiene molte tabelle di codifiche e link utili sulle varie codepage e altre codifiche
- Su Linux, i comandi “man ascii” e “man latin1” mostrano le rispettive tabelle (analogamente per altre codifiche)

La Soluzione Finale: UNICODE

- Chiaramente, l'aggiunta di codifiche su codifiche, ciascuna di al più 256 caratteri (8 bit), non può andare lontano...
- Una soluzione radicale è quella proposta dallo **standard UNICODE:**
 - abbandonare gli standard 8 bit
 - codificare ogni carattere con **32 bit**
 - oltre **4 miliardi di caratteri** diversi codificabili!

UNICODE

- UNICODE assegna ad ogni carattere un proprio codice numerico
 - indipendentemente dalla piattaforma
 - indipendentemente dal programma
 - indipendentemente dal linguaggio
- Con oltre 4 miliardi di codici assegnabili, si può “scialare”
 - finzze tipografiche
 - lingue morte o rarissime

UNICODE

- Definito inizialmente nel 1991, UNICODE ha avuto varie versioni
- L'ultima, UNICODE 5.0, è del Luglio 2006
- Ogni versione “include” le precedenti, e solitamente aggiunge altri caratteri
- UNICODE definisce inoltre:
 - mappature (mappings)
 - codifiche (encodings)
 - regole di ordinamento (collation)

UNICODE

- UNICODE supporta praticamente tutte le lingue e i sistemi di scrittura esistenti al mondo (e alcuni non esistenti), fra cui ovviamente le lingue occidentali e:

Arabic

Armenian

Bengali

Braille embossing patterns

Canadian Aboriginal Syllabics

Cherokee

Coptic

Cyrillic

Devanāgarī

Ethiopic

Georgian

Greek

Gujarati

Gurmukhi (Punjabi)

Han (Kanji, Hanja, Hanzi)

Hangul (Korean)

Hebrew

Hiragana and Katakana (Japanese)

International Phonetic Alphabet (IPA)

Khmer (Cambodian)

Kannada

Lao

Latin

Malayalam

Mongolian

Myanmar (Burmese)

Oriya

Syriac

Tamil

Telugu

Thai

Tibetan

Tifinagh

Yi

Zhuyin (Bopomofo)

UNICODE

- Sono previste poi:
 - alcune lingue morte (per la scrittura di testi accademici)
 - Cuneiforme (Assiro-Babilonese), Deseret, Lineare B (Cretese), Ogham, Etrusco, Fenicio, Runico, Shavian, Ugaritico, ...
 - alcune lingue fantastiche
 - Klingon, Ferengi (proposti ma ritirati)
 - Elfico, Tengwar e Cirth (tuttora in considerazione)
 - alcune altre notazioni comuni
 - simboli matematici
 - simboli musicali

Struttura di UNICODE

- Un **carattere** UNICODE è caratterizzato dal suo codice numerico, detto **code point**, solitamente rappresentato con 8 cifre esadecimali
 - Per esempio, fi (legatura di f e i) è rappresentato dal codice 0000FB01, mentre il simbolo “do doppio diesis strumentale” della notazione musicale greca antica (simile a una lambda maiuscola con una gambetta) è 0001D235
- È uso omettere gli “0” iniziali...

Struttura di UNICODE

- I primi 256 code point, da 0 a FF, sono identici ai caratteri ISO-8859-1; ciò semplifica grandemente la conversione fra UNICODE e ISO-Latin-1 (e -15)
- UNICODE ha spazio per 16.777.216 “pagine” di 256 caratteri; molti standard precedenti sono inclusi tali e quali per comodità
- Moltissimi caratteri sono quindi ripetuti più volte in pagine diverse

Struttura di UNICODE

- Similmente, benché UNICODE contempli la possibilità di “legare” caratteri (per esempio, lettere e accenti), molte versioni precombinata sono disponibili per maggiore comodità
- In alcune lingue (arabo, hindi, ecc.) le regole per le legature sono molto complesse: le versioni precombinata risparmiano al software di rendering decisioni penose

Applicazioni di UNICODE

- Con UNICODE, è possibile creare e gestire senza troppa pena documenti multilingue:
 - A, Δ, Ў, ρ, م, ๓, あ, 叶, 葉
- In particolare, tutti gli standard W3C (incluso HTML) supportano UNICODE; è quindi possibile usare qualunque carattere su una pagina web
 - riga precedente: A, Δ, Й, ק, م, ๗, あ, 叶, 葉
 - occorre naturalmente che il browser abbia accesso ai font corrispondenti...

Il problema della codifica

- Abbiamo visto che UNICODE può codificare 4.294.967.296 caratteri distinti
- Ogni carattere occupa 32 bit (contro gli 8 delle altre codifiche); i documenti richiedono quindi 4 volte lo spazio
- Ma la stragrande maggioranza (quasi totalità) dei documenti usa da 60 a 1000 caratteri, per cui basterebbero da 6 a 10 bit... quanto spazio sprecato!

Il problema della codifica

- Per ovviare a questo problema, e garantire maggiore compatibilità con S.O. e applicazioni che non sono in grado di gestire facilmente 32 bit per carattere, UNICODE definisce vari formati di codifica più o meno compatti
- UTF-7, UTF-8, CESU-8, UTF-16, UTF-32, UTF-EBCDIC, SCSU, UCS
- Vediamo alcuni dei più comuni...

UTF-8

- UTF-8 (8-bit UCS/Unicode Transformation Format) è una **codifica a lunghezza variabile** fra **una sequenza di valori a 8 bit** e **una sequenza di caratteri UNICODE**
- Creata da Ken Thompson and Rob Pike (due degli autori originali di UNIX)
- È una codifica diffusa, con molte proprietà utili

UTF-8

- I primi 128 caratteri di UNICODE (0-7F), equivalenti ai caratteri ASCII, sono codificati con il loro codice “naturale”
- Tutti gli altri caratteri sono codificati con due, tre o quattro valori a 8 bit (byte)
 - lo schema garantisce che in questi casi tutti i byte abbiano un valore superiore a 127, in modo che non ci sia confusione con i codici precedenti

UTF-8

- In dettaglio:

Range caratteri UNICODE	Code point (in binario)	Codifica UTF-8 (in binario)	Note
000000–00007F 128 caratteri	0zzzzzzz sette bit z	0zzzzzzz sette bit z	range equivalente all'ASCII; tutti i byte iniziano con un bit a 0
000080–0007FF 1920 caratteri	00000yyy yyzzzzzz tre bit y; due bit y, sei bit z	110yyyyy 10zzzzzz cinque bit y, sei bit z	il primo byte inizia con 110, il successivo con 10.
000800–00FFFF 63488 caratteri	xxxxyyyy yyzzzzzz quattro bit x, quattro bit y; due bit y, sei bit z	1110xxxx 10yyyyyy 10zzzzzz quattro bit x; sei bit y; sei bit z	il primo byte inizia con 1110, i successivi con 10.
010000–10FFFF 1048576 caratteri	000wwwxx xxxxyyyy yyzzzzzz tre bit w, due bit x; quattro bit x, quattro bit y; due bit y, sei bit z	11110www 10xxxxxx 10yyyyyy 10zzzzzz tre bit w; sei bit x; sei bit y; sei bit z	il primo byte inizia con 11110, i successivi con 10

- Proprietà interessante: il codice si auto-sincronizza
 - guardando i primi bit di un byte, so a che punto sono e come gestire i byte successivi

UTF-8: esempio di codifica

- Consideriamo il carattere א (aleph), la prima lettera dell'alfabeto ebraico
 - Il suo code point è 5D0 (000005D0), quindi ricade nel secondo caso della tabella precedente:

Code point	Code point (in bin)	UTF-8 (in bin)	UTF-8 (in hex)
00 00 05 D0	00000 101 11010000	110 10111 10 010000	D7 90
	00000 yyy yyzzzzzz	110 yyyyy 10 zzzzzz	

- Per la decodifica si applica in direzione opposta lo stesso meccanismo

Usi notevoli di UTF-8

- Nel linguaggio di programmazione Java (e derivati), le stringhe sono codificate con UTF-8; i programmi Java sono quindi in grado di gestire nativamente UNICODE
 - C'è qualche piccola differenza; per esempio il code point 0 non è codificato con 00000000 ma con 11000000 10000000
- I file system Macintosh, DVD, e alcuni su UNIX usano UTF-8 per i nomi dei file
- Gli standard relativi al Web e alla e-mail richiedono che un programma compatibile supporti *almeno* UTF-8 come standard di codifica
- Quasi tutti i programmi che trattano testi ASCII sono UTF-8 compatibili!

UTF-16

- UTF-16 (16-bit UCS/Unicode Transformation Format) è una **codifica a lunghezza variabile** fra **una sequenza di valori a 16 bit** e **una sequenza di caratteri UNICODE**
- I primi 65.536 valori di UTF-16 coincidono con i primi 65.536 caratteri di UNICODE, e con la codifica UCS-2

UTF-16

- In dettaglio:
 - caratteri fra 0000 e FFFF vengono rappresentati, su due byte, tali e quali
 - caratteri fra 10000 e 10FFFF (il limite superiore attuale di UNICODE) vengono scomposti e rappresentati con due parole a 16 bit:
 - La prima è data dall'OR fra D800 e i 10 bit alti del code point
 - La seconda è data dall'OR fra DC00 e i 10 bit bassi del code point

UTF-16

- In altre parole:
 - se il valore binario del code point è
00000000 00000000 **xxxxxxxx** **yyyyyyyy**
 - il suo codice UTF-16 è
xxxxxxxx **yyyyyyyy**
 - se invece il valore binario del code point è
00000000 0000**xxxx** **xxxxxxxxyy** **yyyyyyyy**
 - il suo codice UTF-16 è
110110**xx** **xxxxxxxx** 110111**yy** **yyyyyyyy**

UTF-16: esempio di codifica

code point	carattere	codifica UTF-16	glifo
122 (hex 7A)	z (alfabeto latino)	007A	Z
27700 (hex 6C34)	acqua (ideogramma cinese)	6C34	水 (simbolo della chiave di Sol)
119070 (hex 1D11E)	chiave di Sol (musica)	D834 DD1E	chiave di Sol

- Anche in questo caso, per la decodifica si applica lo stesso meccanismo in direzione inversa
 - Si noti che però il codice non si auto-sincronizza (i byte intermedi di un codice su due parole potrebbero a loro volta contenere le sequenze speciali)

Usi notevoli di UTF-16

- Windows (NT, 2000, XP, CE) usa UTF-16 internamente per rappresentare le stringhe
- La macchine virtuali Java e .NET e il S.O. Symbian (quello dei telefonini) fanno altrettanto
- I motori grafici di Cocoa (Macintosh) e QT (Linux) usano UTF-16

UTF-16 e UCS-2

- UCS-2 (Universal Character Set – 2 bytes) è una **codifica a lunghezza fissa** fra **valori a 16 bit** e **i primi 65536 caratteri** di UNICODE
 - caratteri fra 0000 e FFFF vengono rappresentati, su due byte, tali e quali
 - tutti gli altri caratteri **non sono rappresentabili** – punto e basta!
- Codifica superata e resa obsoleta da UTF-16

Usi notevoli di UCS-2

- Le versioni di Windows NT sviluppate prima di Windows 2000 supportavano solo UCS-2, e potevano quindi gestire solo alcune decine di migliaia di caratteri
- Per tutti gli altri usi (es., Cinese) si usavano sistemi ad-hoc

UCS-4

- UCS-4 (Universal Character Set –4 bytes) è una **codifica a lunghezza fissa** fra **valori a 32 bit** e **tutti i caratteri** di UNICODE
 - Ogni carattere viene rappresentato dal suo code point, espresso su 4 byte
 - Si tratta in effetti di una “non-codifica” che lascia tutto come sta
 - Grande spreco di spazio!
- Noto anche come UTF-32

Alcune considerazioni pratiche

- Cosa accade quando la codifica non viene riconosciuta correttamente?
 - Se il testo contiene solo caratteri ASCII (codici 0-127) e la codifica è compatibile con l'ASCII (es. UTF-8, ISO-8859-1/15), si potrebbe non notare nulla
 - Ma se il testo contiene lettere accentate, simboli matematici, lingue straniere... cominciano i guai!
 - Alcuni caratteri appariranno “sbagliati”

Alcune considerazioni pratiche

- Cosa accade quando la codifica non viene riconosciuta correttamente?
 - Se la codifica non è compatibile ASCII (come UTF-16, UCS-4), non c'è salvezza!
 - Anche il documento più semplice apparirà irrimediabilmente “rovinato”
 - In realtà, basterebbe aprirlo con un programma che sia in grado di gestire la codifica corretta

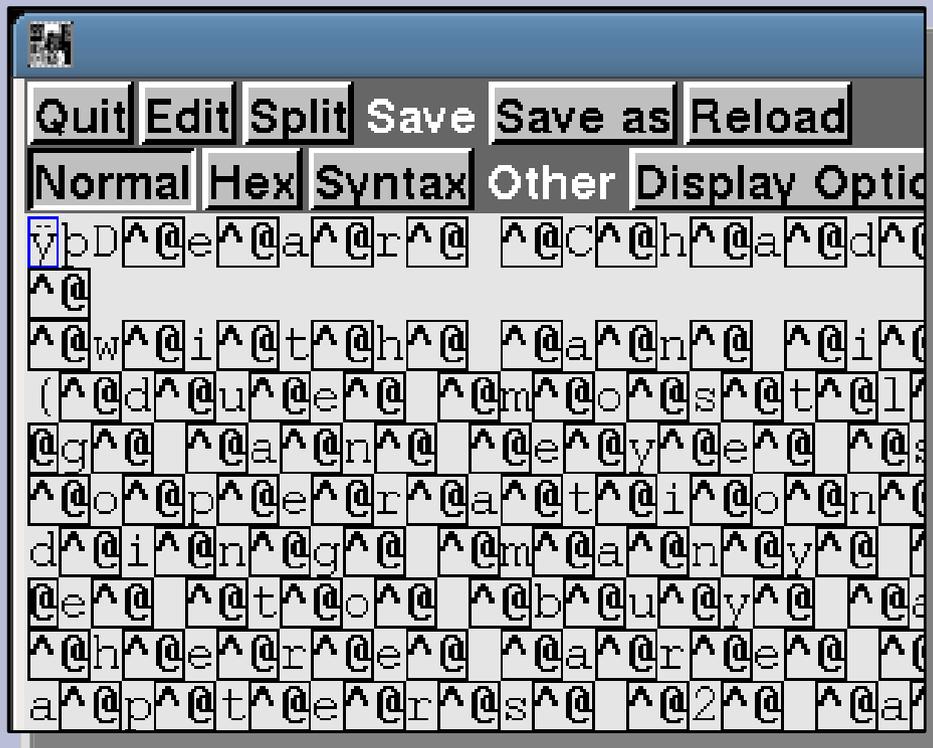
Un esempio

The screenshot shows a terminal window with a file editor. The title bar reads "/home/gervasi/rc.txt". The menu bar includes "Quit", "Edit", "Split", "Save", "Save as", and "Reload". The toolbar shows "Prev", "Next", "Alt", and "Back". The text menu has "Normal", "Hex", "Syntax", "Other", and "Display Options". The main text area contains a file that has become a "mojibake" of the original text, with characters appearing as garbled symbols. At the bottom, a status bar indicates "Read /home/gervasi/rc.txt, 554 lines, 55392 chars".

This screenshot is similar to the first one, showing a terminal window with a file editor. The title bar reads "/home/gervasi/rc.txt". The menu bar includes "Quit", "Edit", "Split", "Save", "Save as", and "Reload". The toolbar shows "Prev", "Next", "Alt", and "Back". The text menu has "Normal", "Hex", "Syntax", "Other", and "Display Options". The main text area contains a file that has become a "mojibake" of the original text, with characters appearing as garbled symbols. At the bottom, a status bar indicates "1,1 Command".

- Aita! Soccorso! Ch'accade?? Che fu??

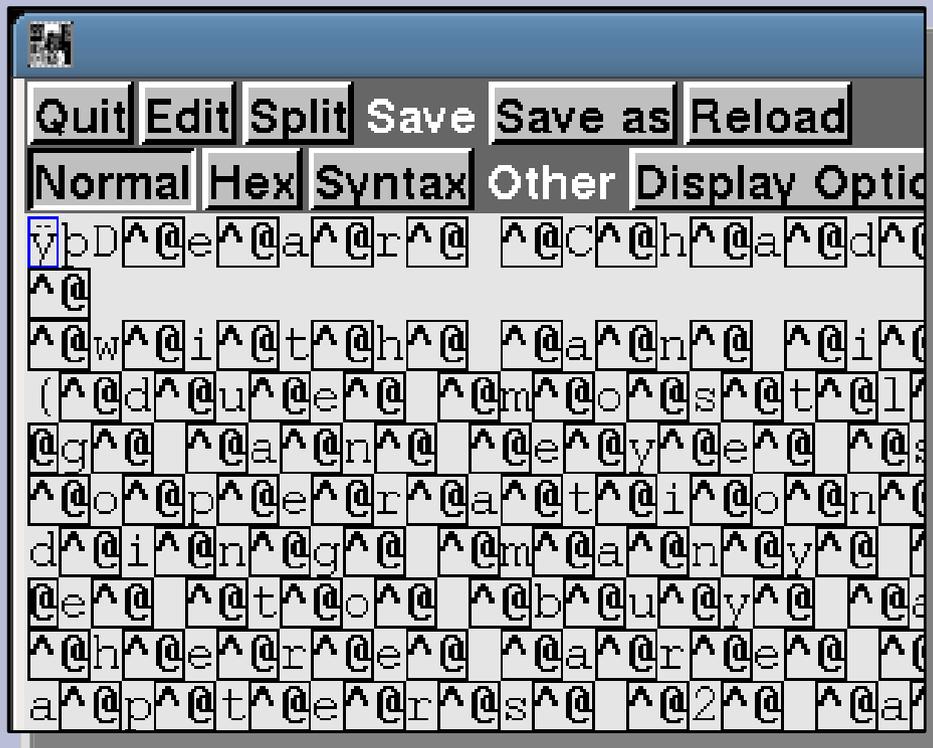
Un esempio



The screenshot shows a text editor window with a menu bar containing 'Quit', 'Edit', 'Split', 'Save', 'Save as', and 'Reload'. Below the menu bar, there are tabs for 'Normal', 'Hex', 'Syntax', 'Other', and 'Display Options'. The main text area displays a document where every character is followed by a caret and an at-sign (^@), representing a NUL character. The visible text is: `y b D ^@ e ^@ a ^@ r ^@ ^@ C ^@ h ^@ a ^@ d ^@ ^@ ^@ w ^@ i ^@ t ^@ h ^@ ^@ a ^@ n ^@ ^@ i ^@ (^@ d ^@ u ^@ e ^@ ^@ m ^@ o ^@ s ^@ t ^@ l ^@ @ g ^@ ^@ a ^@ n ^@ ^@ e ^@ y ^@ e ^@ ^@ s ^@ o ^@ p ^@ e ^@ r ^@ a ^@ t ^@ i ^@ o ^@ n ^@ d ^@ i ^@ n ^@ g ^@ ^@ m ^@ a ^@ n ^@ y ^@ @ e ^@ ^@ t ^@ o ^@ ^@ b ^@ u ^@ y ^@ ^@ a ^@ h ^@ e ^@ r ^@ e ^@ ^@ a ^@ r ^@ e ^@ ^@ a ^@ p ^@ t ^@ e ^@ r ^@ s ^@ ^@ 2 ^@ ^@ a ^@`

- Notiamo che il documento contiene caratteri “normali”, intervallati da ^@
- Ma ^@ è il simbolo ASCII di NUL, ovvero un byte di valore 0...
- Sembra quindi che **ogni** carattere sia codificato in **due** byte anziché uno...

Un esempio

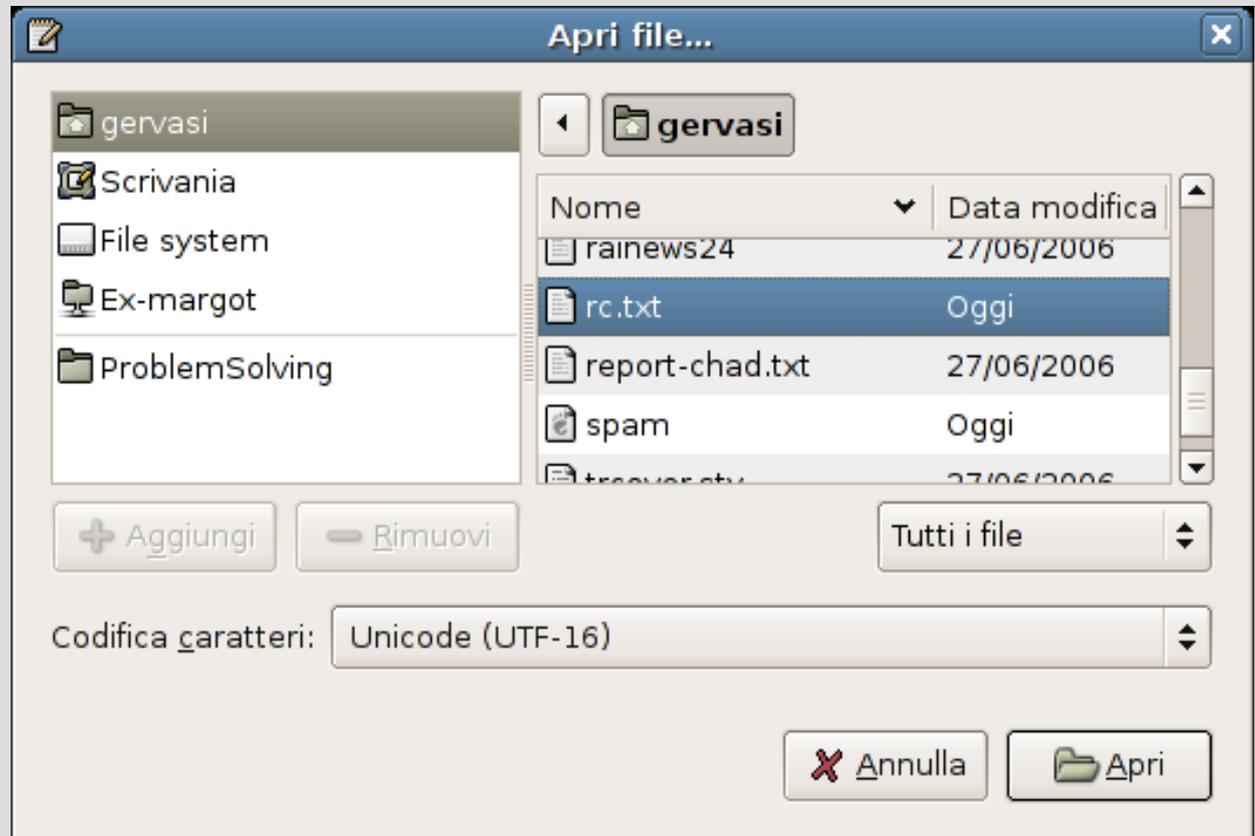


- Notiamo che il documento contiene caratteri “normali”, intervallati da ^@
- Ma ^@ è il simbolo ASCII di NUL, ovvero un byte di valore 0...
- Sembra quindi che **ogni** carattere sia codificato in **due** byte anziché uno...

*Terribile sospetto:
sarà mica che il
documento è
codificato in UTF-16?*

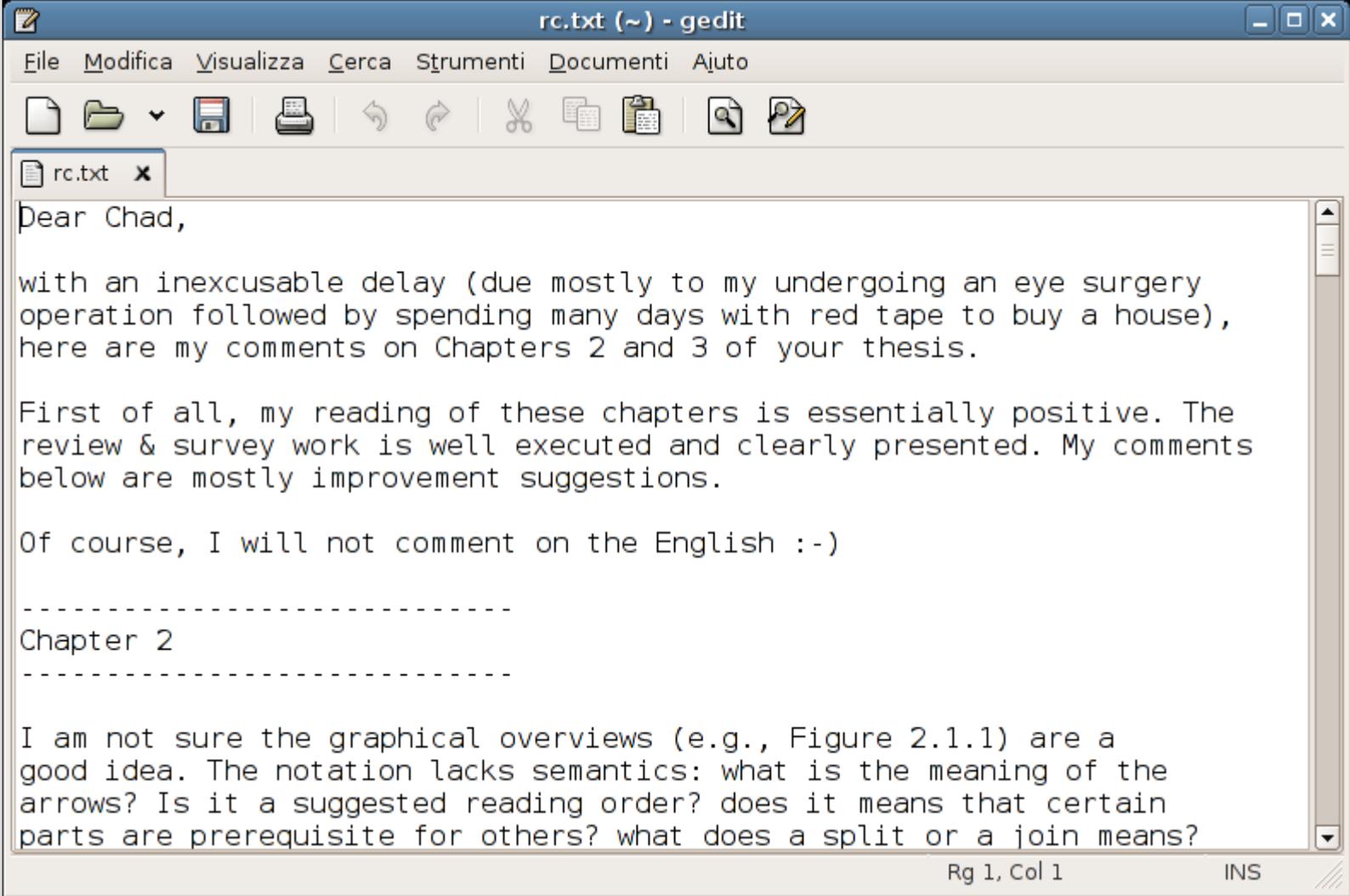
Un esempio

- Apriamo lo stesso file con un'applicazione che ci permette di scegliere la codifica da usare...



Un esempio

- Et voilà!



The screenshot shows a gedit text editor window titled "rc.txt (~) - gedit". The menu bar includes "File", "Modifica", "Visualizza", "Cerca", "Strumenti", "Documenti", and "Ajuto". The toolbar contains icons for file operations like opening, saving, printing, undo, redo, cut, copy, paste, search, and help. A single tab labeled "rc.txt" is open. The text content is as follows:

```
Dear Chad,  
  
with an inexcusable delay (due mostly to my undergoing an eye surgery  
operation followed by spending many days with red tape to buy a house),  
here are my comments on Chapters 2 and 3 of your thesis.  
  
First of all, my reading of these chapters is essentially positive. The  
review & survey work is well executed and clearly presented. My comments  
below are mostly improvement suggestions.  
  
Of course, I will not comment on the English :-)  
  
-----  
Chapter 2  
-----  
  
I am not sure the graphical overviews (e.g., Figure 2.1.1) are a  
good idea. The notation lacks semantics: what is the meaning of the  
arrows? Is it a suggested reading order? does it means that certain  
parts are prerequisite for others? what does a split or a join means?
```

The status bar at the bottom right shows "Rg 1, Col 1" and "INS".

Esercizio

- Riuscite a riconoscere la codifica e il contenuto del testo descritto dalla sequenza di codici (in hex)

C3 88 20 6C 27 6F 72 61 21 0A

Approfondimenti

- Il sito principale su UNICODE è <http://www.unicode.org>; la pagina <http://unicode.org/charts/> è particolarmente affascinante
- L'ultima versione dello standard è pubblicata come *"The Unicode Standard, Version 5.0"*, Addison-Wesley Professional, ISBN: 0321480910 (1472 pagine, \$40)
- Alcuni alfabeti registrati "per uso privato" (inclusi Klingon, Elfico, lingua dei segni, ecc.) sono elencati all'URL <http://www.evertype.com/standards/csur>
- Un confronto fra le varie codifiche di UNICODE è presentato alla pagina "Comparison of Unicode encodings" di Wikipedia (<http://en.wikipedia.org>)