Editoria Elettronica

Vincenzo Gervasi
Dipartimento di Informatica
Università di Pisa

email: gervasi@di.unipi.it
www: http://www.di.unipi.it/~gervasi

Logistica del corso

- Orario delle lezioni:
  - Prima parte (Ottobre)
    - **Lunedì 16:00 aula A1**
    - **Giovedì 16:00 aula A1**
  - Seconda parte (Novembre, Dicembre)
    - **Lunedì 16:00 aula A1**
    - **Giovedì 16:00 laboratorio I-Lab**

Logistica del corso

- Ricevimento studenti:
  - **Lunedì 14:00 studio 333** (Dip. Inf.)
  - **Lunedì 18:00 aula A1** (dopo lezione)
  - **Giovedì 18:00 studio 333** (Dip. Inf.)
- Per casi particolari:
  - su appuntamento, spedire email a gervasi@di.unipi.it
  - durante le pause delle lezioni

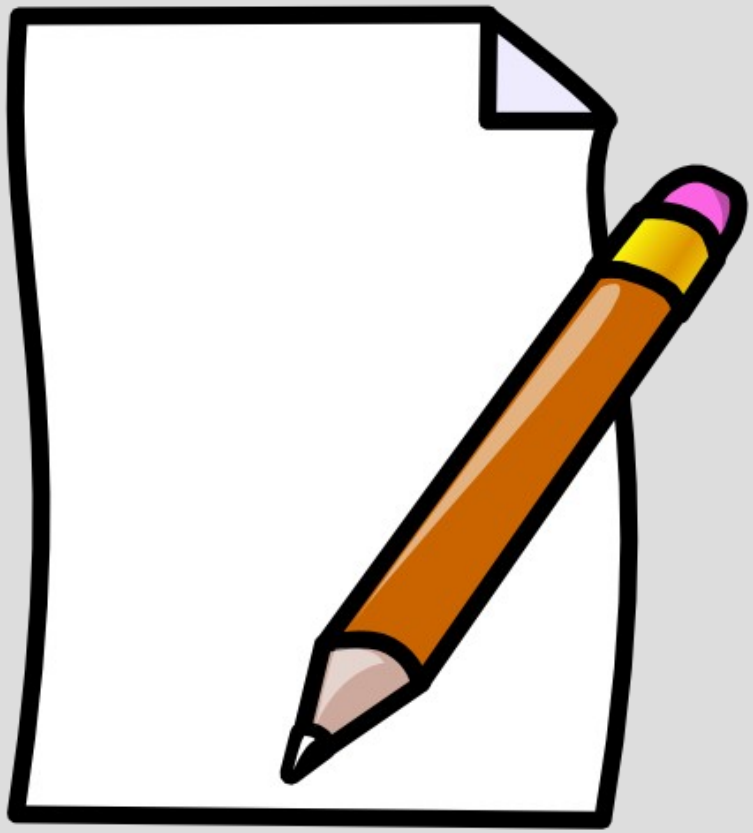
Logistica del corso

- Il corso (5 CFU) viene offerto a:
  - studenti del Corso di Laurea (triennale) in Informatica Umanistica
  - studenti del Corso di Laurea Specialistica in Informatica Umanistica (percorso “Editoria Elettronica”)
  - partecipanti al Modulo Professionalizzante
- **Solo per questi ultimi** è richiesta la firma di presenza
  - (fate circolare il modulo!)

Logistica del corso

- Valutazione
  - prove in itinere (esercitazioni)
  - un elaborato finale (grafico o programma)
  - una prova orale (teorica)
- Materiale didattico
  - dispense del docente (lucidi, senza animazioni!)
  - altro materiale, anche online, verrà indicato di volta in volta durante il corso

Obiettivi del corso



- Formare **figure professionali** capaci di operare in **ambienti digitali** per la **gestione** e la **pubblicazione** di **informazioni strutturate** e **non strutturate** (testi e dati).

Obiettivi del corso

- Formare **figure professionali** capaci di operare in ambienti digitali per la gestione e la pubblicazione di informazioni strutturate e non strutturate (testi e dati).

- Il corso intende mettere i partecipanti in condizione di operare a livello professionale nel mondo dell'editoria
- Forniremo quindi molte informazioni di rilevanza pratica (e un po' di teoria)
- Parte del corso sarà dedicata all'acquisizione diretta di esperienze

Obiettivi del corso

- Formare figure professionali capaci di operare in **ambienti digitali** per la gestione e la pubblicazione di informazioni strutturate e non strutturate (testi e dati).

- La gran parte dell'attività editoriale tipica si svolge oggi con l'ausilio di **mezzi elettronici/digitali**
- La gestione di **informazioni e contenuti digitali** è parte essenziale della **società della conoscenza**
- L'intero **ciclo di vita dell'informazione** è ormai digitale!

Obiettivi del corso

- Formare figure professionali capaci di operare in ambienti digitali per la **gestione** e la **pubblicazione** di informazioni strutturate e non strutturate (testi e dati).

- **Non** ci occuperemo della **generazione** dell'informazione
  - giornalismo, scrittura creativa, scrittura tecnica, ecc.
- Ci occuperemo invece della sua **gestione**:
  - trasformazione
  - presentazione
  - diffusione
  - ...

Obiettivi del corso

- Formare figure professionali capaci di operare in ambienti digitali per la gestione e la pubblicazione di **informazioni strutturate** e **non strutturate** (testi e dati).

- **Non** ci occuperemo di **contenuti** in generale
  - niente film, musica, video musicali, partite di calcio, ...
- Tratteremo invece informazione:
  - **strutturata** (basi di dati, archivi, ecc.)
  - **non strutturata** (testo libero, immagini)
- **Rappresentabile graficamente**

Obiettivi del corso

- Formare figure professionali capaci di operare in ambienti digitali per la gestione e la pubblicazione di informazioni strutturate e non strutturate (testi e dati).

- Il corso avrà un approccio **tecnologico** ai problemi citati
- Studieremo standard, algoritmi, codifiche, ...
- Altri corsi si occuperanno del **contenuto** e della **forma**
  - in particolare, questo corso si coordina con **Progettazione grafica e web design**

Programma (di massima)

- **Informazione: natura, codifica, rappresentazione**
  - codifica dei testi
    - codifiche di caratteri, codifiche di documenti
  - codifica delle immagini
    - codifiche bitmap, formati, algoritmi di compressione
    - codifiche vettoriali, formati, trasformazioni
    - codifica dei colori, modelli colore, usi tipici

Programma (di massima)

- **Tecnologie di acquisizione, memorizzazione e presentazione**
  - tecnologie di scansione, lettura ottica, importazione di documenti
  - tecnologie di memorizzazione
  - tecnologie di stampa
  - tecnologie di presentazione video
  - linguaggio Postscript e formato PDF
  - formati Office proprietari (.doc) e liberi (Open Document Format)

Programma (di massima)

- **Tipografia digitale**
  - storia, design e tecnologie per i font
  - la formattazione del testo
    - algoritmi di formattazione di paragrafo, sillabazione, formattazione di pagina
  - cenni di design e grafica
  - ambienti di word processing e desktop publishing

Programma (di massima)

- **Esercitazioni**
  - programmazione di algoritmi
    - transcodifica
    - formattazione
    - estrazione di informazione
  - disegno di font
    - per il corpo testo (leggibilità)
    - decorativi (titolazione, “dingbats”)
    - effetti speciali

Programma (di massima)

- **Esercitazioni**
  - disegno vettoriale
    - creazione di un logo
  - disegno bitmap / fotoritocco
    - “ripulitura”, fotomontaggio
  - creazione di un documento complesso
    - manuale tecnico
    - testo letterario
    - manifesto o annuncio pubblicitario
- (valutazione delle *capacità tecniche* dimostrate)

Momento buono per ulteriori domande e chiarimenti...

La codifica dei testi

- Un **testo** è una **sequenza** di **simboli**
  - Nota: un **testo** è una cosa diversa da un **documento**
- I **simboli** possono essere:
  - **caratteri** provenienti da un **alfabeto** predeterminato; hanno una rappresentazione grafica
  - **codici di controllo**, anch'essi tratti da un insieme predeterminato, solitamente non hanno rappresentazione grafica, ma alterano la rappresentazione dei caratteri

Esempio di testo

- Perché, finora cosa abbiamo proiettato??

< 'P', 'e', 'r', 'c', 'h', 'é', ',', ' ', 'f', 'i', 'n', … , '?' >

- Nota: stiamo *scrivendo* un testo che *descrive* un testo
  - ci serve un **meta-livello** descrittivo per distinguere i caratteri **di cui** parliamo da quelli **tramite i quali** parliamo
  - qui abbiamo usato il colore

L'ambigua nozione di carattere

- Spesso si tende a identificare un **carattere** ...
  - ... con una lettera dell'alfabeto
  - ... con il segno astratto che rappresenta la lettera
  - ... con la forma concreta che tale segno assume in un certo font
  - ... con l'informazione che tale segno porta
  - ... altre opzioni?

Facciamo chiarezza!

- Un **glifo,** dal greco γλύφω (glýphō), "incidere", è un qualunque **segno** grafico (anticamente, inciso o dipinto)
- Un **grafema** è un segno elementare, non ulteriormente divisibile, del **linguaggio scritto**
- Il grafema coincide tipicamente con una **lettera** dell'alfabeto (in senso generale) del linguaggio in questione

Facciamo chiarezza!

- Un **fonema** è invece l'unità elementare del **linguaggio parlato** (suono)
- La corrispondenza fra fonemi, grafemi, lettere e glifi può essere complessa:
  - “gn” rappresenta un solo fonema, due grafemi (digrafo), due lettere
  - “fi” consta di due grafemi, ma spesso è rappresentato da un solo glifo “**ﬁ**” (legatura) che rappresenta due lettere e due fonemi (analoga legatura si ha per “ffi”)
  - “c/o” consta di due lettere e un simbolo, dunque tre grafemi, un solo glifo “℅”, una lunga serie di fonemi
  - “--” è spesso usato per indicare il trattino lungo “—“; qui due glifi codificano un solo grafema che non corrisponde a nessun fonema

Facciamo chiarezza!

- Un glifo è dunque la **rappresentazione astratta** di un grafema (es., una lettera, un ideogramma cinese), di una parte di un grafema (es., un accento) o di più grafemi (es., una legatura)
- Il grafema è una unità **di testo**
- Il glifo è una unità **grafica**

Facciamo chiarezza!

- Ad ogni glifo (unità grafica astratta) possono corrispondere molteplici disegni (unità grafica concreta)
- Esempio: glifo “A”
  - 
- Raccolte di glifi disegnati con lo stesso stile grafico vengono dette **font** (**fonti** o **tipi**)... ma di queste ci occuperemo più avanti

Una nozione più precisa di carattere

- I **caratteri** (simboli) di cui ci occuperemo trattando della codifica dei testi sono dunque i **grafemi**
- Infatti:
  - abbiamo caratteri che non sono lettere: %
  - abbiamo caratteri che non rappresentano fonemi: !
  - abbiamo più caratteri per la stessa lettera: a, A
  - però: grafemi multi-glifo come “--” contano come due caratteri distinti

Codifica dei testi

- Com'è noto, i calcolatori elettronici sono in grado di trattare soltanto i simboli 0 e 1 (*on* e *off*, *vero* e *falso*,...)
- Sequenze di 0 e 1 costituiscono *numeri binari*; con essi si possono esprimere numeri di dimensione arbitraria
- Una codifica assegna un **codice numerico** a ogni unità testuale
  - tipicamente, **un numero per carattere**

Codifica dei testi

- I codici storicamente più importanti sono:
  - codice **ASCII** (oggi di uso universale)
  - codice **EBCDIC** (oggi quasi scomparso)
- Codici moderni basati sull'ASCII
  - **ISO-8859-xxx**
  - Windows “Code Pages” (**CP-xxx**)
- Codici “del futuro”
  - **UNICODE** (e varianti di codifica)

Il codice ASCII

- **ASCII** = **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
- Una codifica standard creata nel 1967 (ultima revisione del 1986)
- Basato sull'alfabeto inglese:
  - niente lettere accentate
  - niente caratteri tipici di altre lingue (ß, Ç, ñ)
  - alcuni simboli tipici dell'inglese commerciale ($, %, @, &)

Il codice ASCII

- L'ASCII codifica:
  - **33** codici di controllo (oggi molti sono in disuso)
  - **95** caratteri
- In totale si hanno quindi 128 codici, che possono essere memorizzati in **7 bit** (cifre binarie) di informazione
- Visto che quasi tutti i computer lavorano con unità di 8 bit, un bit veniva lasciato libero per altri usi

Codici di controllo ASCII

Binario	Ottale	Decimale	Hex	Nome	Immissione da tastiera	Codice escape (C, Java)	Descrizione	
0000 0000	0	0	0	NUL	^@	\0	Null character	Carattere “nullo”
0000 0001	1	1	1	SOH	^A		Start of Header	
0000 0010	2	2	2	STX	^B		Start of Text	
0000 0011	3	3	3	ETX	^C		End of Text	
0000 0100	4	4	4	EOT	^D		End of Transmission	
0000 0101	5	5	5	ENQ	^E		Enquiry	
0000 0110	6	6	6	ACK	^F		Acknowledgment	
0000 0111	7	7	7	BEL	^G	\a	Bell	Campanella
0000 1000	10	8	8	BS	^H	\b	Backspace	Ritorno indietro di un carattere
0000 1001	11	9	9	HT	^I	\t	Horizontal Tab	Tabulazione orizzontale
0000 1010	12	10	0A	LF	^J	\n	Line feed	Nuova riga
0000 1011	13	11	0B	VT	^K		Vertical Tab	Tabulazione verticale
0000 1100	14	12	0C	FF	^L	\f	Form feed	Nuova pagina
0000 1101	15	13	0D	CR	^M	\r	Carriage return	Ritorno carrello
0000 1110	16	14	0E	SO	^N		Shift Out	
0000 1111	17	15	0F	SI	^O		Shift In	
0001 0000	20	16	10	DLE	^P		Data Link Escape	
0001 0001	21	17	11	DC1	^Q		Device Control 1 (oft. XON)	
0001 0010	22	18	12	DC2	^R		Device Control 2	
0001 0011	23	19	13	DC3	^S		Device Control 3 (oft. XOFF)	
0001 0100	24	20	14	DC4	^T		Device Control 4	
0001 0101	25	21	15	NAK	^U		Negative Acknowledgement	
0001 0110	26	22	16	SYN	^V		Synchronous Idle	
0001 0111	27	23	17	ETB	^W		End of Trans. Block	
0001 1000	30	24	18	CAN	^X		Cancel	
0001 1001	31	25	19	EM	^Y		End of Medium	
0001 1010	32	26	1A	SUB	^Z		Substitute	
0001 1011	33	27	1B	ESC	^[	\e	Escape	Codice di escape
0001 1100	34	28	1C	FS	^\		File Separator	
0001 1101	35	29	1D	GS	^]		Group Separator	
0001 1110	36	30	1E	RS	^^		Record Separator	
0001 1111	37	31	1F	US	^_		Unit Separator	
0111 1111	177	127	7F	DEL	^?		Delete	Cancella (carattere successivo)

Caratteri ASCII

Simboli, numeri, punteggiatura				Lettere maiuscole				Lettere minuscole			
Binario	Decimale	Hex	Glifo	Binario	Decimale	Hex	Glifo	Binario	Decimale	Hex	Glifo
0010 0000	32	20	*(spazio)*	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(	0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29	)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	116	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	121	79	y
0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	z
0011 1011	59	3B	;	0101 1011	91	5B	[	0111 1011	123	7B	{
0011 1100	60	3C	<	0101 1100	92	5C	\	0111 1100	124	7C	|
0011 1101	61	3D	=	0101 1101	93	5D	]	0111 1101	125	7D	}
0011 1110	62	3E	>	0101 1110	94	5E	^	0111 1110	126	7E	~
0011 1111	63	3F	?	0101 1111	95	5F	_				

Esempio di codifica ASCII

- Vediamo come viene codificata la frase “Ciao, mamma!”:

Carattere	C	i	a	o	,		m	a	m	m	a	!
Codice	67	105	97	111	44	32	109	97	109	109	97	33

- È importante ricordare che il calcolatore “vede” solo i codici numerici
- Serve un altro sistema per sapere quale codice si sta usando
  - tipicamente, è dato implicitamente
  - a volte, si specifica l'*encoding* in testa al documento

Codifica di testi in ASCII

- Un testo ASCII è codificato da una serie di caratteri, con interposti codici di controllo
- Spesso, gli unici codici di controllo usati sono quelli che indicano gli a-capo
  - ad ogni fine riga, se il testo è **preformattato** (*a-capo fisici*)
  - solo a fine paragrafo, se il testo **non è formattato** (*a-capo logici*)
    - in questo caso, il programma che visualizza il testo dovrà riformattarlo al volo

Codifica di testo in ASCII

- **Attenzione** a una trappola comune:
  - UNIX (Linux) e molti altri sistemi usano il codice **LF** (13) per indicare a-capo
  - Macintosh usa il codice **CR** (10)
  - Windows e MS-DOS usano entrambi i codici, ovvero la sequenza **CR+LF** (10+13)
- Se il testo è stato generato su un S.O. diverso e programmi usati non fanno la conversione automaticamente, può capitare di avere a che fare con linee **estremamente** lunghe!

Il codice EBCDIC

- Prodotto da IBM (tentando di competere con ASCII) nel 1963-64
- Usato su tutti i *mainframe* IBM dal System/360 in poi (tranne che da Linux su zSeries)
- Codice a 8 bit, contiene più caratteri rispetto all'ASCII, ma...
  - poco standardizzato
  - codifica bislacca (alfabeto disgiunto!)
- Interesse solo storico

Il codice EBCDIC

Hex	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
4-			â	ä	à	á	ã	å	ç	ñ	[	.	<	(	+	!
5-	&	é	ê	ë	è	í	î	ï	ì	ß	]	$	*	)	;	^
6-	-	/	Â	Ä	À	Á	Ã	Å	Ç	Ñ	¦	,	%	_	>	?
7-	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	@	'	=	"
8-	Ø	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	«		»	ð	ý	þ	±
9-	°	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	ª	º	æ	¸	Æ	¤
A-	µ	~	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**	¡	¿	Ð	Ý	Þ	®
B-	¢	£	¥	·	©	§	¶	¼	½	¾	¬	|	¯	¨	´	×
C-	**{**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**		ô	ö	ò	ó	õ
D-	**}**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	¹	û	ü	ù	ú	ÿ
E-	\	÷	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	²	Ô	Ö	Ò	Ó	Õ
F-	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	³	Û	Ü	Ù	Ú	

CCSID500, una delle varianti di EBCDIC

Svantaggi di ASCII e EBCDIC

- ASCII, limitato a 95 caratteri, non era sufficiente per linguaggi diversi dall'Inglese (e anche per quello...)
  - Es: perche' invece di perché
- EBCDIC, mai standardizzato davvero, era inutilizzabile come formato di scambio fra macchine diverse
  - persino il codice di “a capo” poteva essere diverso su macchine diverse!

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
160	A0	NO-BREAK SPACE	
161	A1	INVERTED EXCLAMATION MARK	¡
162	A2	CENT SIGN	¢
163	A3	POUND SIGN	£
164	A4	EURO SIGN	€
165	A5	YEN SIGN	¥
166	A6	CAPITAL LETTER S WITH CARON	Š
167	A7	PARAGRAPH SIGN	§
168	A8	SMALL LETTER S WITH CARON	š
169	A9	COPYRIGHT SIGN	©
170	AA	FEMININE ORDINAL INDICATOR	ª
171	AB	LEFT ANGLE QUOTATION MARK	«
172	AC	NOT SIGN	¬
173	AD	SOFT HYPHEN	-
174	AE	REGISTERED TRADE MARK SIGN	®
175	AF	MACRON	¯
176	B0	DEGREE SIGN, RING ABOVE	°
177	B1	PLUS-MINUS SIGN	±
178	B2	SUPERSCRIPT TWO	²
179	B3	SUPERSCRIPT THREE	³
180	B4	CAPITAL LETTER Z WITH CARON	Ž
181	B5	MICRO SIGN	µ
182	B6	PILCROW SIGN	¶
183	B7	MIDDLE DOT	·
184	B8	SMALL LETTER Z WITH CARON	ž
185	B9	SUPERSCRIPT ONE	¹
186	BA	MASCULINE ORDINAL INDICATOR	º
187	BB	RIGHT ANGLE QUOTATION MARK	»
188	BC	CAPITAL OE DIGRAPH	Œ
189	BD	SMALL OE DIGRAPH	œ
190	BE	CAPITAL LETTER Y WITH DIAERESIS	Ÿ
191	BF	INVERTED QUESTION MARK	¿
192	C0	CAPITAL LETTER A WITH GRAVE ACCENT	À
193	C1	CAPITAL LETTER A WITH ACUTE ACCENT	Á
194	C2	CAPITAL LETTER A WITH CIRCUMFLEX ACCENT	Â
195	C3	CAPITAL LETTER A WITH TILDE	Ã
196	C4	CAPITAL LETTER A WITH DIAERESIS	Ä
197	C5	CAPITAL LETTER A WITH RING ABOVE	Å
198	C6	CAPITAL DIPHTHONG A WITH E	Æ
199	C7	CAPITAL LETTER C WITH CEDILLA	Ç
200	C8	CAPITAL LETTER E WITH GRAVE ACCENT	È
201	C9	CAPITAL LETTER E WITH ACUTE ACCENT	É
202	CA	CAPITAL LETTER E WITH CIRCUMFLEX ACCENT	Ê
203	CB	CAPITAL LETTER E WITH DIAERESIS	Ë
204	CC	CAPITAL LETTER I WITH GRAVE ACCENT	Ì
205	CD	CAPITAL LETTER I WITH ACUTE ACCENT	Í
206	CE	CAPITAL LETTER I WITH CIRCUMFLEX ACCENT	Î
208	D0	CAPITAL ICELANDIC LETTER ETH	Ð
209	D1	CAPITAL LETTER N WITH TILDE	Ñ
210	D2	CAPITAL LETTER O WITH GRAVE ACCENT	Ò
211	D3	CAPITAL LETTER O WITH ACUTE ACCENT	Ó
212	D4	CAPITAL LETTER O WITH CIRCUMFLEX ACCENT	Ô
213	D5	CAPITAL LETTER O WITH TILDE	Õ
214	D6	CAPITAL LETTER O WITH DIAERESIS	Ö
215	D7	MULTIPLICATION SIGN	×
216	D8	CAPITAL LETTER O WITH OBLIQUE STROKE	Ø
217	D9	CAPITAL LETTER U WITH GRAVE ACCENT	Ù
218	DA	CAPITAL LETTER U WITH ACUTE ACCENT	Ú
219	DB	CAPITAL LETTER U WITH CIRCUMFLEX ACCENT	Û
220	DC	CAPITAL LETTER U WITH DIAERESIS	Ü
221	DD	CAPITAL LETTER Y WITH ACUTE ACCENT	Ý
222	DE	CAPITAL ICELANDIC LETTER THORN	Þ
223	DF	SMALL GERMAN LETTER SHARP s	ß
224	E0	SMALL LETTER a WITH GRAVE ACCENT	à
225	E1	SMALL LETTER a WITH ACUTE ACCENT	á
226	E2	SMALL LETTER a WITH CIRCUMFLEX ACCENT	â
227	E3	SMALL LETTER a WITH TILDE	ã
228	E4	SMALL LETTER a WITH DIAERESIS	ä
229	E5	SMALL LETTER a WITH RING ABOVE	å
230	E6	SMALL DIPHTHONG a WITH e	æ
231	E7	SMALL LETTER c WITH CEDILLA	ç
232	E8	SMALL LETTER e WITH GRAVE ACCENT	è
233	E9	SMALL LETTER e WITH ACUTE ACCENT	é
234	EA	SMALL LETTER e WITH CIRCUMFLEX ACCENT	ê
235	EB	SMALL LETTER e WITH DIAERESIS	ë
236	EC	SMALL LETTER i WITH GRAVE ACCENT	ì
237	ED	SMALL LETTER i WITH ACUTE ACCENT	í
238	EE	SMALL LETTER i WITH CIRCUMFLEX ACCENT	î
239	EF	SMALL LETTER i WITH DIAERESIS	ï
240	F0	SMALL ICELANDIC LETTER eth	ð
241	F1	SMALL LETTER n WITH TILDE	ñ
242	F2	SMALL LETTER o WITH GRAVE ACCENT	ò
243	F3	SMALL LETTER o WITH ACUTE ACCENT	ó
244	F4	SMALL LETTER o WITH CIRCUMFLEX ACCENT	ô
245	F5	SMALL LETTER o WITH TILDE	õ
246	F6	SMALL LETTER o WITH DIAERESIS	ö
247	F7	DIVISION SIGN	÷
248	F8	SMALL LETTER o WITH OBLIQUE STROKE	ø
249	F9	SMALL LETTER u WITH GRAVE ACCENT	ù
250	FA	SMALL LETTER u WITH ACUTE ACCENT	ú
251	FB	SMALL LETTER u WITH CIRCUMFLEX ACCENT	û
252	FC	SMALL LETTER u WITH DIAERESIS	ü
253	FD	SMALL LETTER y WITH ACUTE ACCENT	ý
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	Codica 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*	21 33	22 34	23 35	24 36	25 37	26 38	27 39	28 40	29 41	2A 42	2B 43	2C 44	2D 45	2E 46	2F 47
	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
3.	30 48	31 49	32 50	33 51	34 52	35 53	36 54	37 55	38 56	39 57	3A 58	3B 59	3C 60	3D 61	3E 62	3F 63
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
4.	40 64	41 65	42 66	43 67	44 68	45 69	46 70	47 71	48 72	49 73	4A 74	4B 75	4C 76	4D 77	4E 78	4F 79
	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
5.	50 80	51 81	52 82	53 83	54 84	55 85	56 86	57 87	58 88	59 89	5A 90	5B 91	5C 92	5D 93	5E 94	5F 95
	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
6.	60 96	61 97	62 98	63 99	64 100	65 101	66 102	67 103	68 104	69 105	6A 106	6B 107	6C 108	6D 109	6E 110	6F 111
	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	⌂
7.	70 112	71 113	72 114	73 115	74 116	75 117	76 118	77 119	78 120	79 121	7A 122	7B 123	7C 124	7D 125	7E 126	2302 127
	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
8.	C7 128	FC 129	E9 130	E2 131	E4 132	E0 133	E5 134	E7 135	EA 136	EB 137	E8 138	EF 139	EE 140	EC 141	C4 142	C5 143
	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
9.	C9 144	E6 145	C6 146	F4 147	F6 148	F2 149	FB 150	F9 151	FF 152	D6 153	DC 154	A2 155	A3 156	A5 157	20A7 158	192 159
	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
A.	E1 160	ED 161	F3 162	FA 163	F1 164	D1 165	AA 166	BA 167	BF 168	2310 169	AC 170	BD 171	BC 172	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.	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
	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
E.	3B1 224	DF 225	393 226	3C0 227	3A3 228	3C3 229	B5 230	3C4 231	3A6 232	398 233	3A9 234	3B4 235	221E 236	3C6 237	3B5 238	2229 239
	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	
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	207F 252	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 pagere 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** diveri 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”
  - finezze 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...

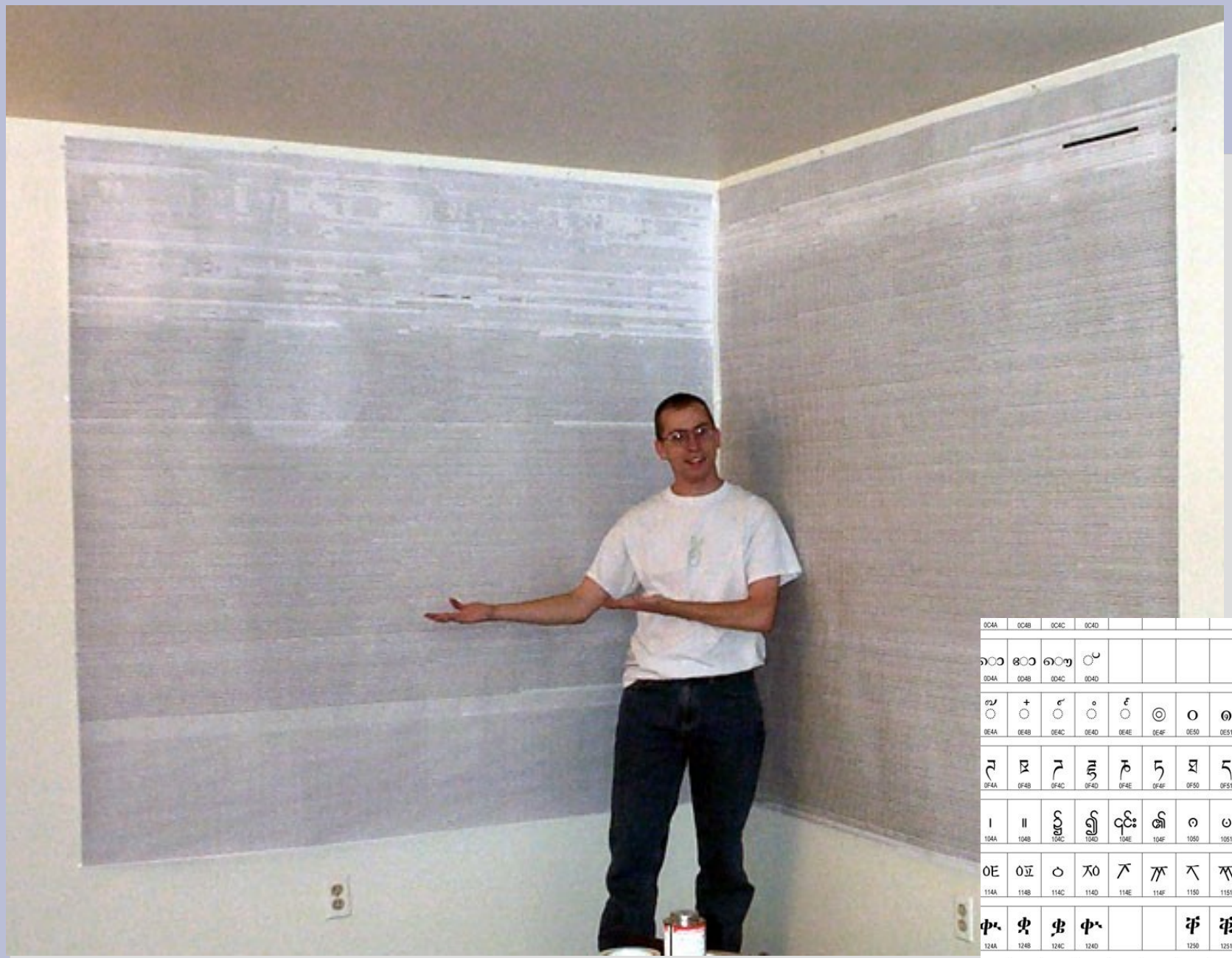
Struttura di UNICODE

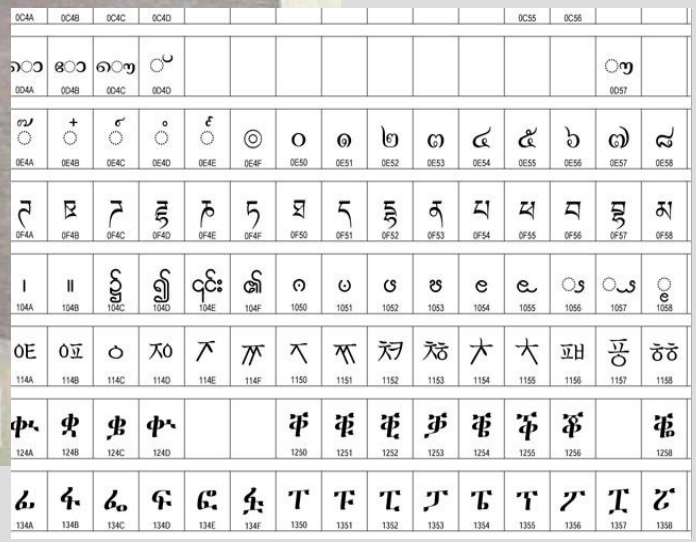
- I primi 256 code point, da 0 a FF, sono identici ai caratteri ISO-8859-1; ciò semplifica grandemente la conversione fra UNICODE e ISO-Latin-1 (e -15)
- UNICODE ha spazio per 16.777.216 “pagine” di 256 caratteri; molti standard precedenti sono inclusi tali e quali per comodità
- Moltissimi caratteri 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 precombinate sono disponibili per maggiore comodità
- In alcune lingue (arabo, hindi, ecc.) le regole per le legature sono molto complesse: le versioni precombinate risparmiano al software di rendering decisioni penose

Dimensioni di UNICODE





Applicazioni di UNICODE

- Con UNICODE, è possibile creare e gestire senza troppa pena documenti multilingue:
  - A, Δ, Й, ק ,م ,ঙ, 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 lettere 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	00000101 11010000	11010111 10010000	D7 90
	00000yyy yyzzzzzz	110yyyyy 10zzzzzz	

- 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 10FFFD (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 0000xxxx xxxxxxyy yyyyyyyy
  - il suo codice UTF-16 è 110110xx xxxxxxxx 110111yy 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
119070 (hex 1D11E)	chiave di Sol (musica)	D834 DD1E	(simbolo della 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

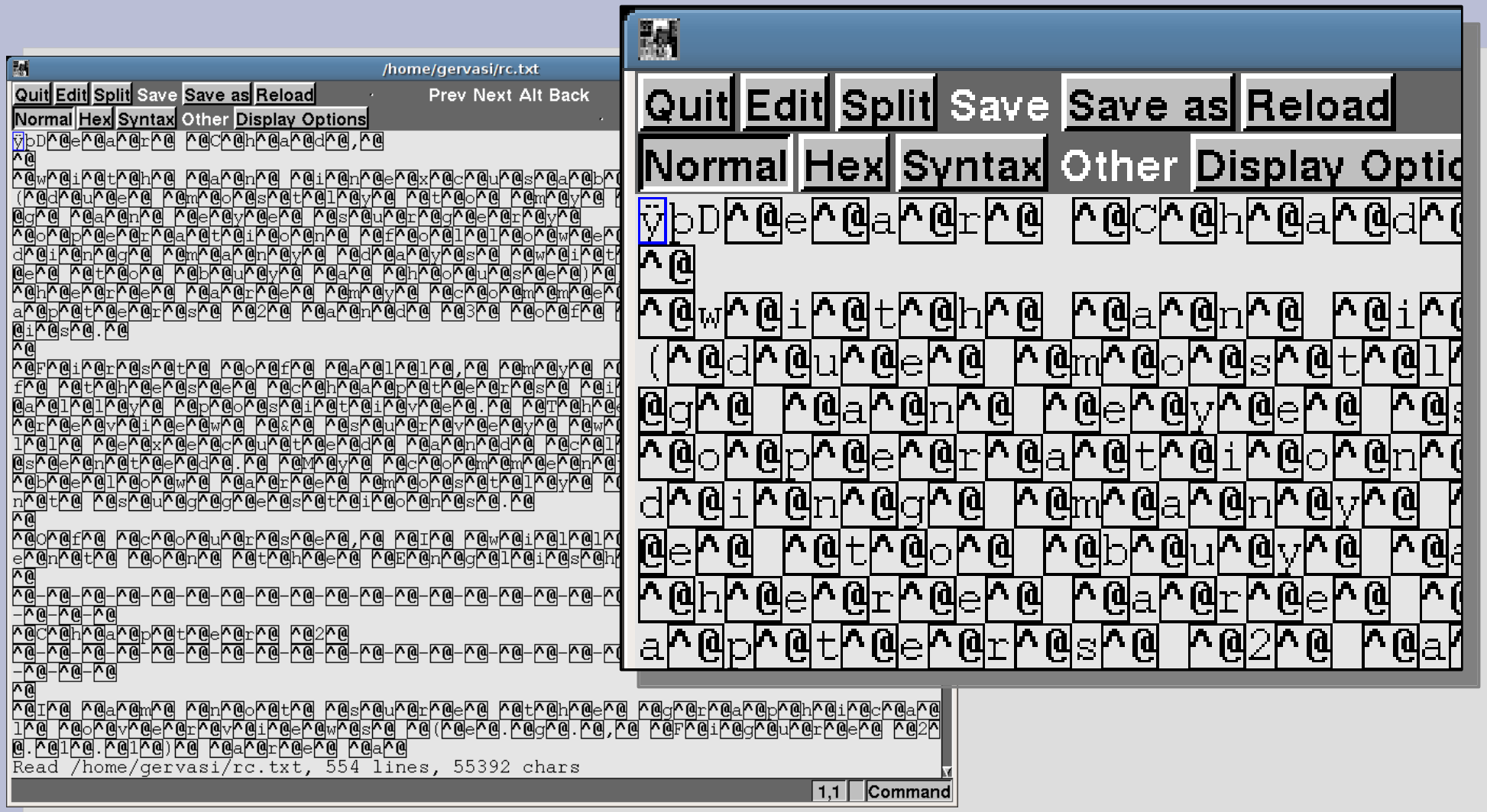
Alcune considerazioni pratiche

- 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

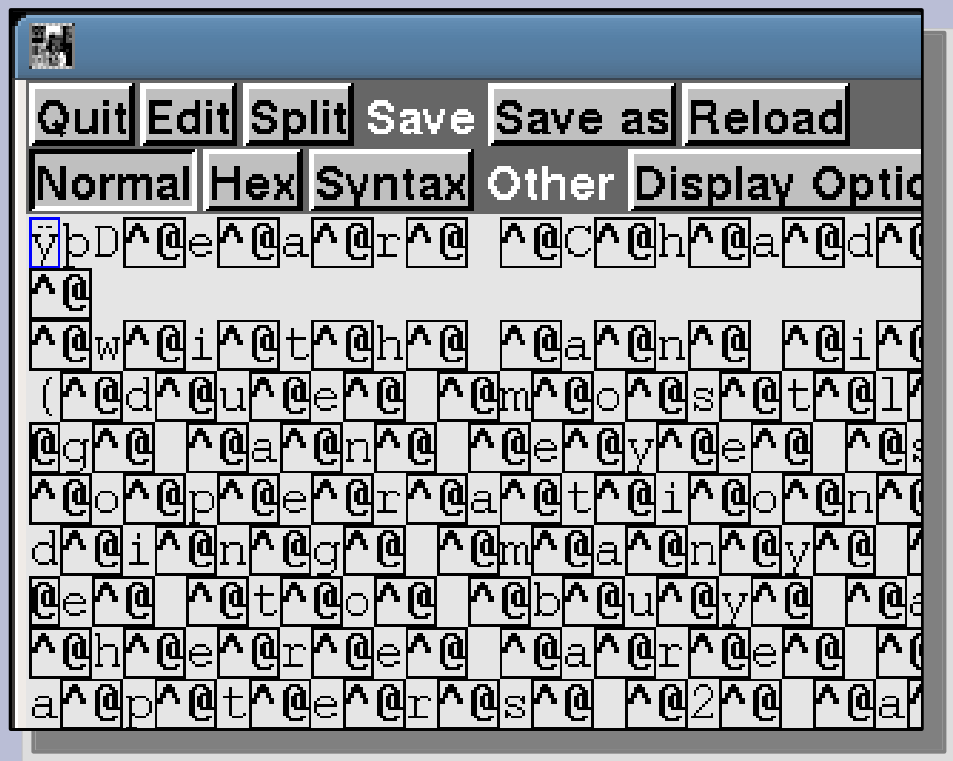
- Cosa accade quando la codifica non viene riconosciuta correttamente?
  - Se la codifica non è compatibile ASCII (come UTF-16, UCS-4), non c'è salvezza!
  - Anche il documento più semplice apparirà irrimediabilmente “rovinato”
  - In realtà, basterebbe aprirlo con un programma che sia in grado di gestire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

Un esempio



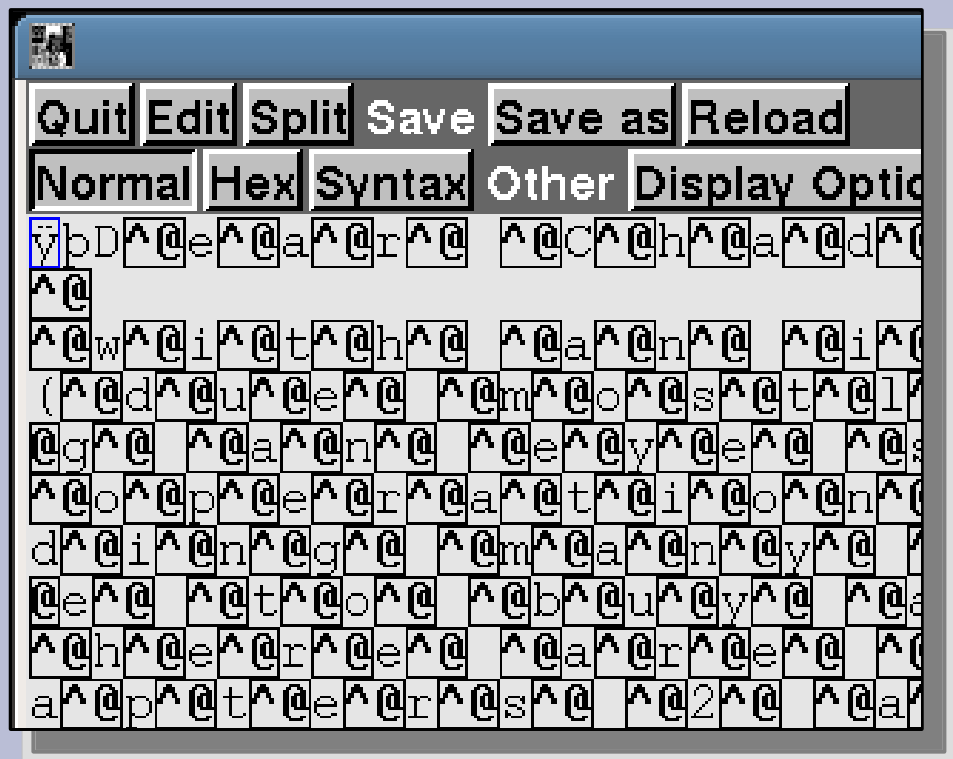
- Aita! Soccorso! Ch'accade?? Che fu??

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...

Un esempio

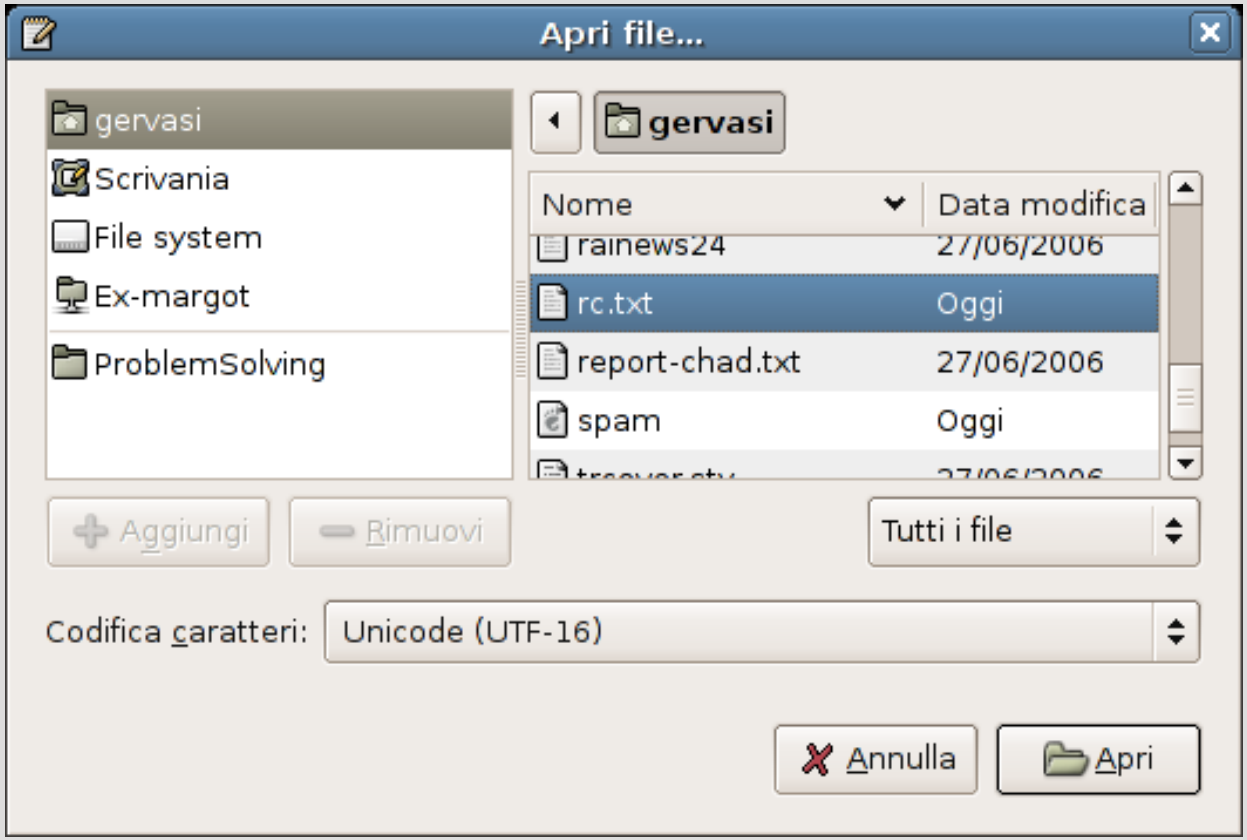


Terribile sospetto: sarà mica che il documento è codificato in UTF-16?

- 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...

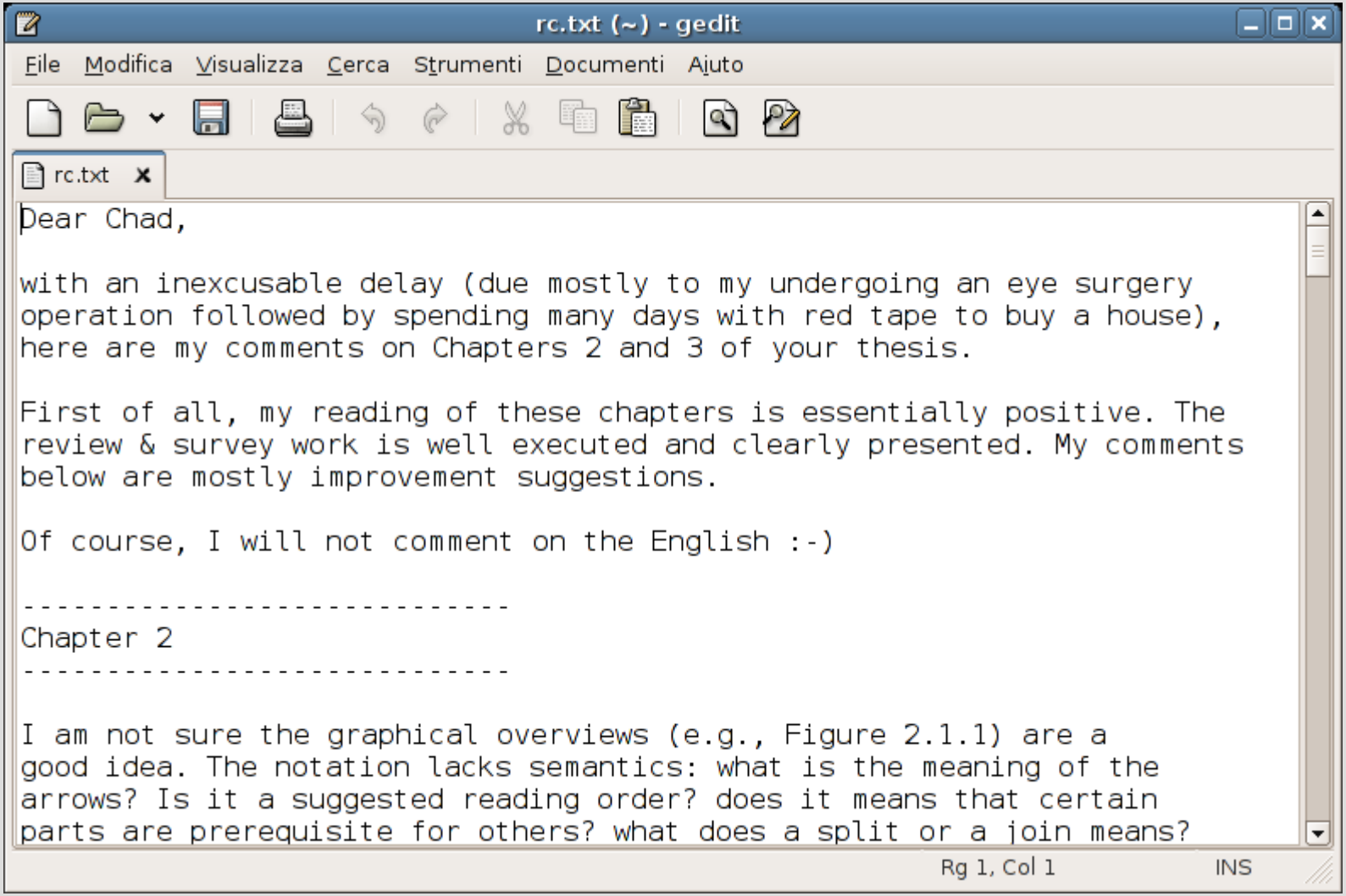
Un esempio

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



Un esempio

- Et voilà!



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)