char

Sistemi Operativi

A.A. 2020-2021

Teleprinter

- Una teleprinter (teletypewriter, Teletype o **TTY**) è un dispositivo elettromeccanico per inviare e ricevere messaggi. Inizialmente furono utilizzati nella telegrafia (1830).
- Furono adattate per fornire un'interfaccia utente ai primi computer mainframe e minicomputer, inviando dati digitati al computer e stampando la risposta.
- Potevano essere collegate su linee telefoniche tramite un modem.

ASCII

- ASCII: è uno standard di codifica dei caratteri per la comunicazione elettronica (prima edizione: 1963)
- Originariamente basato sull'**alfabeto inglese**, ASCII codifica 128 caratteri specificati in numeri interi a **sette bit** (da 0 a 127).

ASCII

- Novantacinque dei caratteri codificati sono stampabili: includono le cifre da 0 a 9, le lettere minuscole dalla a alla z, le lettere maiuscole dalla A alla Z e i simboli di punteggiatura.
- La specifica ASCII originale includeva 33 codici di controllo non stampabili originati con macchine **Teletype**; la maggior parte di questi sono ormai obsoleti, anche se alcuni sono ancora comunemente usati, come il ritorno a capo, l'avanzamento di riga e i codici di tabulazione.

Unicode

- Unicode è uno standard informatico per la codifica, la rappresentazione e la gestione del testo
- Prevede un massimo di 1.114.064 caratteri definibili ($2^{16} + 2^{20} – 2^{11}$)

- maggio 2019, Unicode 12.1: definisce 137.994 caratteri (composto da 137.766 caratteri grafici, 163 caratteri di formato e 65 caratteri di controllo)

- non tutti questi punti di codice corrispondono necessariamente a caratteri visibili; molti, ad esempio, sono assegnati a codici di controllo come il ritorno a capo

Unicode

- Unicode può essere implementato con diverse **codifiche di carattere** (character encoding). Lo standard Unicode definisce UTF-8, UTF-16 e UTF-32 (e altre).

- Le codifiche di carattere più comunemente utilizzate sono UTF-8, UTF-16 e UCS-2 (senza supporto completo per Unicode, un precursore di UTF-16); GB18030 è standardizzato in Cina e implementa pienamente Unicode (ma non è uno standard Unicode ufficiale)

Unicode

- Lo standard Unicode definisce uno «spazio di codice» (codespace) di valori numerici che vanno da 0 a $10FFFF_{16}$, chiamato «punti di codice» e indicato come **U+0000** fino a **U+10FFFF**
- "U+" più il valore del punto di codice in esadecimale, anteposto con zeri iniziali necessari per ottenere un minimo di quattro cifre

- Tabella unicode ad esempio consultabile qui (e molti altri siti):
- https://unicode-table.com/en/

Unicode - esempi

‘$’: ASCII 36 (decimale), **Unicode U+0024** (esadecimale)

‘A’: ASCII 65, **Unicode U+0041**

‘è’ : **U+00E8** (non presente in ASCII) [decimale 232]

😁 : **U+1F601** ‘Grinning Face with Smiling Eyes Emoji’

😸 : **U+1F638** ‘Grinning Cat Face with Smiling Eyes Emoji’

Unicode - esempi

- ‘$’: https://unicode-table.com/en/0024/
- ‘A’: https://unicode-table.com/en/0041/
- ‘è’: https://unicode-table.com/en/00E8/
- 😁 : https://unicode-table.com/en/1F601/
- 😼 : https://unicode-table.com/en/1F638/

UTF-8

- Può codificare tutti i 1.114.064 caratteri Unicode
- codifica a lunghezza variabile (1,2,3 o 4 bytes)
- Nasce per essere compatibile all'indietro verso ASCII
- utilizza un byte per i primi 128 «punti di codice» (~=caratteri) e fino a 4 byte per altri caratteri
- I primi 128 punti di codice Unicode rappresentano i caratteri ASCII => qualsiasi testo ASCII è anche un testo UTF-8

UTF-8

- UTF-8 è la codifica principale sul World Wide Web (utilizzata in oltre il 94% dei siti Web a novembre 2019)
- il W3C raccomanda UTF-8 come codifica predefinita in XML e HTML
- Linux Kernel usa UTF-8

Esercizio: provare il comando:

touch 😈.txt

$ Unicode U+0024 (ASCII 36)

Encoding

Encoding	hex	dec (bytes)	dec	binary
UTF-8	24	36	36	00100100
UTF-16BE	00 24	0 36	36	00000000 00100100
UTF-16LE	24 00	36 0	9216	00100100 00000000
UTF-32BE	00 00 00 24	0 0 0 36	36	00000000 00000000 00000000 00100100
UTF-32LE	24 00 00 00	36 0 0 0	603979776	00100100 00000000 00000000 00000000

https://unicode-table.com/en/0024/

Little endian, big endian

- Intel: little endian (LE)
- ARM: big endian (BE)

è U+00E8

Encoding

Encoding	hex	dec (bytes)	dec	binary
UTF-8	C3 A8	195 168	50088	11000011 10101000
UTF-16BE	00 E8	0 232	232	00000000 11101000
UTF-16LE	E8 00	232 0	59392	11101000 00000000
UTF-32BE	00 00 00 E8	0 0 0 232	232	00000000 00000000 00000000 11101000
UTF-32LE	E8 00 00 00	232 0 0 0	3892314112	11101000 00000000 00000000 00000000

😸 U+1F638 ‘Grinning Cat Face with Smiling Eyes Emoji’

Encoding

Encoding	hex	dec (bytes)	dec	binary
UTF-8	F0 9F 98 B8	240 159 152 184	4036991160	11110000 10011111 10011000 10111000
UTF-16BE	D8 3D DE 38	216 61 222 56	3627933240	11011000 00111101 11011110 00111000
UTF-16LE	3D D8 38 DE	61 216 56 222	1037580510	00111101 11011000 00111000 11011110
UTF-32BE	00 01 F6 38	0 1 246 56	128568	00000000 00000001 11110110 00111000
UTF-32LE	38 F6 01 00	56 246 1 0	955646208	00111000 11110110 00000001 00000000

UTF-16

- codifica a lunghezza variabile: 16 o 32 bit per codificare un carattere
- Può codificare tutti i 1.114.064 caratteri unicode
- Usato da Windows, **Java**
- Raramente usato su Unix/Linux, macOS
- È un successore di UCS-2 (2-byte Universal Character Set, codifica «solo» 65536 caratteri)

esempio

- https://replit.com/@MarcoTessarotto/unicode-sample

- msg_array[0] = 😸
- msg_array [0][0] = f0
- msg_array [0][1] = 9f
- msg_array [0][2] = 98
- msg_array [0][3] = b8
- msg_array [0][4] = 0