UNIVERSITÀ
DEGLI STUDI
DI MILANO

Codifica del Testo

Paolo Ceravolo

paolo.ceravolo@unimi.it

Editoria Digitale

SESAR LAB

UNIVERSITÀ
DEGLI STUDI
DI MILANO

IL FILE

UNIVERSITÀ DEGLI STUDI DI MILANO

- Un **file** è l'**unità di dati elementare** gestita dal file system ed è caratterizzato da:
  - un contenuto, cioè una sequenza di byte
  - un identificatore, unico per ogni file
- I dati contenuti nel file devono essere opportunamente interpretati:
- Le regole con cui interpretare il file sono chiamate **formato** del file
  - i formati dei file corrispondono a codifiche convenzionali di informazioni adottate dalle applicazioni

FILE BINARI

UNIVERSITÀ DEGLI STUDI DI MILANO

- Un file binario è un file che può contenere qualsiasi tipo di dati, codificato in codice binario a scopo di archiviazione o utilizzo, per essere utilizzato richiede un'**applicazione** capace di comprendere la **codifica utilizzata**
- I file binari sono solitamente concepiti come sequenze di byte: le singole cifre bit che costituiscono il file sono raggruppate in gruppi di otto
- Alcuni file binari contengono header, cioè contenitori di metadati usati dai programmi associati ai file per riconoscerne ed interpretarne il contenuto. Ad esempio, un file GIF può contenere più immagini, e gli header sono utilizzati per identificare e descrivere ciascun blocco di dati

FILE DI TESTO

UNIVERSITÀ DEGLI STUDI DI MILANO

- Un file di testo contiene solo **caratteri alfanuemrici**, che compongono un testo leggibile direttamente dagli utenti senza bisogno di installare programmi appositi
- Questi file sono molto utilizzati sia perché possono essere letti direttamente dall'utente, sia per documenti che per il codice sorgente di un linguaggio di programmazione, inoltre garantiscono una buona portabilità da un OS ad un altro
  - A livello di archiviazione sono anch'essi codificati come sequenze di byte ma costituiscono una **categoria a parte** prece sono supportati da **tutti gli OS** e utilizzati per la scrittura di **codice sorgente**

FILE ESEGUIBILE

UNIVERSITÀ DEGLI STUDI DI MILANO

- Un file eseguibile contiene un programma eseguibile per un computer, ovvero un programma scritto in **linguaggio macchina** nel formato adatto ad essere caricato dal sistema operativo
- Gli eseguibili sono dipendenti dalla piattaforma: per esempio, un file eseguibile per un sistema Microsoft Windows non è direttamente utilizzabile in sistemi Unix o Mac OS (a meno di non usare un software di emulazione). Questa restrizione è dovuta a tre motivi:
  - processori diversi supportano linguaggi macchina diversi
  - sistemi operativi diversi usano formati diversi per i file eseguibili
  - per effettuare operazioni di base (input/output) i programmi eseguibili devono avvalersi delle primitive fornite dal sistema operativo

PORTABILITA'

UNIVERSITÀ DEGLI STUDI DI MILANO

- Abbiamo visto nelle lezioni precedenti l'importanza della **portabilità** nei sistemi informatici
- Chiediamoci quindi cosa rende un file portabile
  - **Il livello di supporto nei sistemi operativi**
    - I file di testo sono supportati da tutti i sistemi operativi e concorrono al funzionamento del sistema stesso
  - **Il livello di diffusione di uno standard**
    - PDF e DOC sono ad esempio molto diffusi
  - **Il livello di variabilità nelle versioni di uno standard**
    - Vecchie versione di DOC non sono più supportate oggi, vecchie version di PDF sono ancora compatibili con gli interpreti attuali

PORTABILITA'

- Abbiamo visto nelle lezioni precedenti l'importanza della **portabilità** nei sistemi informatici
  - L'AGID pubblica linee guida raccomandazioni riguardo i formati di file da utilizzare
  - Esiste inoltre un International Comparison of Recommended File Formats group (ICRF) che pubblica un confronto dei formati dei file consigliati a livello internazionale
  - Queste linee guida, tuttavia, fotografano l'esistente e non ci aiutano a valutare le tendenze di adozione dei formati di file

CODIFICA

UNIVERSITÀ DEGLI STUDI DI MILANO

- Parlare di **formato** di file significa parlare dello **schema di codifica** necessario per interpretarlo
- La codifica è quel processo che ci permette di

esprimere informazioni e messaggi mediante le regole e i simboli di un sistema convenzionale (il codice) stabilito concordemente dall'emettitore e dal ricevitore dei messaggi allo scopo di trasmettere o elaborare automaticamente le informazioni - Treccani

- I formati di testo sono quelli più portabili e sono stati storicamente utilizzati per facilitare l'interoperabilità dei sistemi
  - ASCII nel modello ISO/OSI
  - HTML nel WWW
  - JSON nel Web 2.0

CODIFICA DEL TESTO

- Nei calcolatori il testo è memorizzato come sequenza binaria, la codifica del testo si occupa di definire la corrispondenza tra una sequenza binaria e il corrispondente carattere alfanumerico
  - Prendiamo la codifica `Windows-1252`:
    - Il carattere “é” è salvato in memoria usando la sequela binaria che in esadecimale corrisponde al valore “E9”
  - Per la codifica `DOS latin-1` il valore “E9” corrisponde al carattere “Ú”
- La difficoltà è quindi duplice:
  - È necessario scegliere con cura, a seconda dell'applicazione di destinazione, la codifica da utilizzare quando si salva un testo
  - Quando è il momento di visualizzare un testo, è necessario poter determinare la codifica utilizzata

LA CODIFICA ASCII

- Se la **quantità di informazione** da rappresentare (considerando l'alfabeto inglese): 26 lettere + le 26 lettere maiuscole + 10 cifre decimali + 30 caratteri circa di uso comune (., *, %, etc.) + alcuni caratteri di controllo (spazi, interruzione di riga) ovvero **120 caratteri** circa
- Numero di BIT necessari: **bastano 7 bit** poiché $27 = 128 > 120$
- Su questa base è stato creato il codice ASCII, basato appunto sull'uso di 7 bit per ogni carattere
- Esempio:
  - A diventa 1000001
  - B diventa 1000010
  - a diventa 1100000

LA CODIFICA ASCII

UNIVERSITÀ DEGLI STUDI DI MILANO

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

LIMITI DI ASCII

UNIVERSITÀ DEGLI STUDI DI MILANO

- La **globalizzazione** di Internet ha proposto il problema di rendere correttamente gli alfabeti di **migliaia di lingue** nel mondo
  - Poiché i 7 bit usati da ASCII consentono di rappresentare solo 128 caratteri diversi, sono stati definiti nel tempo altri codici per consentire la rappresentazione di più caratteri
  - ASCII esteso: usa 8 bit (un BYTE) per carattere. Consente quindi di rappresentare anche caratteri accentati e lingue diverse dall'inglese.

ALTRE CODIFICHE STORICHE

UNIVERSITÀ DEGLI STUDI DI MILANO

- Extended Binary Characters for Digital Interchange Code (EBCDIC)
- Codifica proprietaria IBM del 1965 a 8 bit. IBM è molto sicura della superiorità dei suoi chip e si azzarda fin dagli anni cinquanta ad usare tutti e 8 i bit del byte
- ISO 646
- Una codifica ISO del 1991 che permettere l'uso di caratteri nazionali europei in un contesto sostanzialmente ASCII. Sono lasciati 12 codici liberi per le versioni nazionali dei vari linguaggi europei. Ogni versione nazionale li sostituisce con caratteri propri. I caratteri sacrificati sono: # $ @ \ ¬ ` { | } ~
- ISO 8859/1 (ISO Latin 1)
- Un estensione standard dell'ASCII che comprende un certo numero di caratteri degli alfabeti europei. È compatibile all'indietro con ASCII: estende i soli caratteri >127

UNICODE

- Lo standard Unicode definisce tre codifiche che consentono a uno stesso dato di essere trasferito utilizzando un vocabolario basato su 1, 2 o 4 byte: UTF-8, UTF-16, UTF-32. Queste tre codifiche descrivono gli stessi caratteri e possono essere trasformate efficientemente una nell'altra
- **Composizione dinamica**. Alcuni caratteri (in arabo, in cinese, ma anche, banalmente, le lettere accentate o con modificatori degli alfabeti europei) sono composizioni di codifiche indipendenti. In certi casi ricorrere ad un doppio codice per un carattere composto è eccessivo. Per i simboli di uso frequente esiste un singolo carattere equivalente
- **Semantica**. Ogni carattere possiede un suo significato preciso (la ß tedesca è diversa dalla ß greca) nonché proprietà come direzione, esigenze di spaziatura, capacità di combinazione
- **Convertibilità**. Esiste un meccanismo di conversione tra Unicode e altre codifiche precedenti, in modo da garantire compatibilità all'indietro

UNICODE

- UTF-8
- I caratteri più frequenti sono codificati con 1 byte, poi 2, 3 e 4 per i meno frequenti. Usato molto nel WEB. Ha il vantaggio che il set di caratteri ASCII mantiene la stessa codifica, può quindi essere usato su vecchie applicazioni. Ma non si accede direttamente a una codifica prima si deve codificare come raggruppare i byte
- UTF-32
- Ogni carattere è codificato da 4 byte. Usato quando non si ritiene di avere problemi di memoria oppure se si fa uso di vocabolari poco frequenti
- UTF-16
- Caratteri codificati da copie di byte, 2 copie per i meno frequenti. Usato per situazioni dove si vuole bilanciare l'uso di memoria con la velocità di accesso ai dati

UNICODE

UNIVERSITÀ DEGLI STUDI DI MILANO

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

COMPOSIZIONE DEI BYTE IN UTF-8

UNIVERSITÀ DEGLI STUDI DI MILANO

- UTF-8 segue un meccanismo basato su pattern specifici dei bit iniziali di ogni byte. A seconda del numero di bit necessari per rappresentare il carattere Unicode, UTF-8 utilizza 1, 2, 3 o 4 byte
  - 1 byte (7 bit utili):
    - Viene utilizzato per i caratteri con codice Unicode da 0 a 127 (0x00 - 0x7F).
  - 2 byte (11 bit utili):
    - Viene utilizzato per i caratteri con codice Unicode da 128 a 2047 (0x80 - 0x7FF). Il primo byte inizia con 110xxxxx, e il secondo con 10xxxxxx.
  - 3 byte (16 bit utili):
    - Viene utilizzato per i caratteri con codice Unicode da 2048 a 65535 (0x800 - 0xFFFF). Il primo byte inizia con 1110xxxx, il secondo e il terzo con 10xxxxxx.
  - 4 byte (21 bit utili):
    - Viene utilizzato per i caratteri con codice Unicode da 65536 a 1114111 (0x10000 - 0x10FFFF). Il primo byte inizia con 11110xxx, gli altri tre byte con 10xxxxxx.

PROBLEMI DI CODIFICA

UNIVERSITÀ DEGLI STUDI DI MILANO

- Se UNICODE è uno standard universale, in grado di rappresentare i caratteri di tutti gli alfabeti, perché ci sono ancora problemi di codifica?
  - Il motivo principale è che i vecchi sistemi non si sono necessariamente evoluti contemporaneamente alla rivoluzione Unicode
  - Microsoft si è presa la libertà di creare le proprie tabelle di caratteri derivate dalle tabelle ISO-8859-x. Pertanto, l'invio di un file di testo Windows a un server Linux o a un'applicazione proprietaria può facilmente generare disalineamenti
  - Data l'ampia numerosità di caratteri presenti in UNICODE, i font dei caratteri non sempre sono prodotti per l'intero set, quindi alcuni font possono essere applicati solo a specifici sottoinsiemi di UNICODE

PROBLEMI DI CODIFICA

UNIVERSITÀ DEGLI STUDI DI MILANO

- Se UNICODE è uno standard universale, in grado di rappresentare i caratteri di tutti gli alfabeti, perché ci sono ancora problemi di codifica?
  - Il Byte Order Mark (BOM) è una sequenza di byte UNICODE non stampabili posta all'inizio di un testo UNICODE per facilitarne l'interpretazione. BOM non è né standard né obbligatorio, ma rende più facile per le applicazioni compatibili determinare il sottotipo del formato Unicode e definire la direzione di lettura dei byte. Questo spesso causa problemi di compatibilità perché non tutte le applicazioni sanno come gestire BOM. Per le applicazioni non compatibili, questa sequenza di byte viene decodificata in ASCII esteso

DETERMINER LA CODIFICA DI UN FILE

UNIVERSITÀ DEGLI STUDI DI MILANO

- Indipendentemente dall'origine di un file, può essere utile verificare con assoluta certezza il suo formato e mostrare l'eventuale tag BOM
- Se all'inizio del file è presente un tag BOM, si tratta di un testo in formato Unicode:

```
UTF-8 = EF BB BF
UTF-16 Big Endian = FE FF
UTF-16 Little Endian = FF FE
UTF-32 Big Endian = 00 00 FE FF
UTF-32 Little Endian = FF FE 00 00
```

https://validator.w3.org/i18n-checker/

- L'assenza del BOM non consente di stabilire che il file non sia codificato in UNICODE
- Rimuoverlo può aumentare la compatibilità con le applicazioni.