Zeichensätze

Abriss über Zeichencodes von ASCII bis Unicode

- geschichtliche Entwicklung
- Eigenschaften
- Theorie
- abschliessende Bemerkungen

gehalten 2006-09-25 von Emil Obermayr bei hpcom in Braunschweig

ASCII

- **A** merican **S** tandard- **C** ode for **I** nformation **I** nterchange
- 7 Bit Code
- achtes Bit für Parität (Fehlerbehandlung)
- amerikanisches Alphabet in Klein- und Großbuchstaben
- Ziffern, Satzzeichen und wichtige mathematische Zeichen
- Steuercodes
- bis heute Grundlage aller gebräuchlichen Codes

"DIN"

- ISO 646
- ähnlich ASCII
- []{}\|~@ werden aufgegeben ...
- ... zugunsten von äöüÄÖÜß§

Codepage 437

- aka **OEM**
- 8 Bit Code
- Fehlerbehandlung wird dem System überlassen
- wichtigste europäische Sonderzeichen
- grafische Zeichen zum Darstellen von "Fenstern"
- basiert auf einem Zeichensatz von *Wang*
- später dann CP 850 mit mehr Sonderzeichen und weniger Grafik

ISO 8859-1

- aka **latin 1**
- gehört zu einer Gruppe von Codes für europäische Sprachräume
- enthält jeweils alle Sonderzeichen des Sprachraums
- keine grafischen Zeichen
- unter Windows auch bekannt als Codepage 1252
- derzeit aktueller Standard

quoted printable

- aka **"MIME"**
- 7 Bit Code
- achtes Bit entfällt für ASCII-Kompatibilität
- zugehöriger 8-Bit Zeichensatz muß angegeben werden
- Codierung mit '=' und 8-Bit-Zahl in hex , z.B. '=FC' für 'ü'
- benutzt für Email
- genau genommen kein Zeichensatz,
- ebenso wie die nun folgenden Beispiele,
- sonden ein " Encoding "

HTML- Entities

- Sonderzeichen werden mit Escape -Zeichen & eingeleitet...
- ... und mit ; abgeschlossen
- und bilden zusammen ein *Entitiy*
- Sonderzeichen können mit Namen benannt werden: **ä**
- oder mit Zahlen **‘** und auch in hex **ü**
- benutzt im HTML- Source des WWW
- aber nicht in URLs!

UTF -8

- 8 Bit Code
- kodiert komplettes 4-Byte-Unicode
- passt die Länge eines Zeichens an nötige Bitlänge an
- ASCII wird in 1 Byte übertragen
- ... latin -1 in 2 ...
- ... seltenere Zeichen in immer längeren Sequenzen
- bis zu 7 Byte in der Praxis
- aufkommender und zukünftiger Standard für Datenübertragung

UTF-16

- 16-Bit-Code
- mit *Surrogate* -Sequenz auf 32 Bit
- kodiert 20 Bit von den 32 des Unicode
- Standard für *Java*

UCS-2

- 16 Bit Code
- kodiert sauber 16 Bit von Unicode
- reicht für die meisten Texte
- erlaubt effektive Datenverarbeitung ähnlich zu latin 1
- Standard bei *MS SQL Server* **nchar**

UCS -4

- 32 Bit Code
- kodiert vollständiges Unicode sauber in 4 Byte
- belegt dadurch viel Speicher
- wird (noch) nicht zur Speicherung benutzt
- noch selten implementiert
- wenige Fonts verfügbar
- dies ist nun wieder ein "echter" Zeichensatz

Überblick

Name	Länge	%	fix	Raster	US	ä ĉ	ى	[]{}	⊨	텔	Standard	in ASCII	Kommentar
ASCII	1	-	+	+	+	-	-	+	-	-	+	++++	der Klassiker, 7-Bit-Code für das lateinische Alphabet, Ziffern und Satzzeichen
DIN	1	100	+	+	+	-/o	-	-	-	-	-	++	ein Versuch, 7-Bit-ASCII-Variante für deutsche Sonderzeichen
CP437	1	100	+	+	+	o	-	+	+	-	+	++	MS-DOS, ASCII-Erweiterung auf 8 Bit um europäische Sonderzeichen und Grafikzeichen
ISO	1	100	+	+	+	o/+	-	+	-	-	++	++	state of the art bzw. status quo, ASCII-Erweiterung auf 8 Bit um europäische Sonderzeichen
MIME	1;3	120	-	-	+	o/+	-	+	-	-	++	+	E-Mail, Codierung mit Escape-Sequenzen fester Länge aus ASCII-Zeichen
HTML	1;4-13	150	-	-	+	+	+	+	+	+	+++++	+ +++	WWW, Codierung mit Entities aus ASCII-Zeichen variabler Länge
UTF-8	1-7	110	-	-	+	+	+	+	+	+	+++++	+	die Zukunft...? 8-Bit Codierung mit variabler Länge
UTF-16	2;4	200	-	+	+	+	+	+	+	+	++++	-	... oder das hier? 16-Bit Codierung mit Escape-Sequenzen fester Länge für 20 Bit der UCS
UCS-2	2	200	+	+	+	+	+	+	+	-	+++	-	Für schnelle Verarbeitung von Texten, 16 Bit Codierung der BMP von UCS
UCS-4	4	400	+	+	+	+	+	+	+	+	+++++	-	eierlegende Wollmilchsau, aber einfach zu groß, 4-Byte-Codierung aller UCS

- Die erste Spalte ist der Stichwort-artige Name; darauf folgt die Code-Länge in Byte und der Prozentsatz um den sich ein Text mit 10% deutschen Umlauten „aufbläht“ in dieser Codierung; dann ob der Code eine feste Bytelänge hat und ob die Codes in ein festes Byte-Raster passen
- Es folgen Spalten für den amerikanischen Standardzeichensatz; für europäische Umlaute, nicht-europäische Umlaute; nicht-alphanumerische Zeichen aus ASCII, Grafikzeichen aus MSDOS und sehr selten benötigte Zeichen
- Standard ist ein Mass für die Fähigkeit Zeichen darzustellen; von einem + für ASCII bis 5 + für volle 32 Bit Unicode; umgekehrt ein Mass dafür wie gut sich Texte in dieser Codierung in einem reinem ASCII-Editor noch lesen lassen
- Die letzte Spalte beschreibt den Code kurz in Worten

Theorie

- quoted printable arbeite mit einem Escape-Zeichen **=** und konstanter Länge des Codes: **=FC**
- HTML- Entities arbeiten mit einem Start- und einem Stop -Zeichen der Escape-Sequenz: **&...;**

UTF-16 Theorie

- Normalerweise beschreibt UTF-16 genau wie UCS-2 die Zeichen-Plane 0, also die BMP
- 2 Bitkombinationen von je 6 Bit einem Byte sind in UCS-2 eigentlich "verboten"
- wenn diese vorkommen, ergeben die verbleibenden 2x2=4 Bit die gewünschte Plane minus 1
- Also können 17 Planes genutzt werden
- die folgenden 2 Byte sind der Code in der gewählten Plane
- "verschwendet" Codes in den unteren 16 Bit
- die "verbotenen" Bitkombinationen werden **Surrogates** genannt
- wurde zum Standard weil ursprünglich 16 Bit, also UCS -2, reichen sollten

UTF-8 Theorie

- führende 1-Bits im ersten Byte geben die Anzahl der benötigten Bytes an
- Folge-Bytes werden mit 10... gekennzeichnet
- führende 0 bedeutet "1 Byte in ASCII"
- ermöglicht mehrere Codierungen "kurzer" Codes
- per Konvention wird die kürzeste benutzt

Ausblick

- es gibt noch ganz andere Standard-Codes wie z.B. EBCDIC von IBM
- und solche die nur auf Rechnern eines Herstellers liefen wie von Sinclair und Commodore
- auch für Unicode gibt es weitere Codierungen wie UTF -7, Punycode, SCSU und BOCA
- Unicode kann noch viel mehr als nur Zeichen kodieren, wie z.B. Schreibrichtung festlegen
- mit Unicode ist Entwicklung der Zeichensätze zu einem vorläufigen, erfolgreichen Abschluß gekommen
- aber es gibt nach wie vor Entwicklung in diesem Bereich
- Details siehe http://unicode.org/ (vollständig, englisch und authoritive)
- auch viele Informationen: http://de.wikipedia.org/wiki/Unicode (anschaulich, einfach und auf deutsch)