ASCII und UTF-8

ASCII ist die Abkürzung für *"American Standard Code for Information Interchange"* und ist ein Standard für die Codierung von Text. Er definiert 95 druckbare Zeichen (printable characters) und 33 Steuerzeichen (control characters).

Printable characters:

0x20 / 32 space	0x40 / 64 @	0x60 / 96 `
0x21 / 33 !	0x41 / 65 A	0x61 / 97 a
0x22 / 34 "	0x42 / 66 B	0x62 / 98 b
0x23 / 35 #	0x43 / 67 C	0x63 / 99 c
0x24 / 36 $	0x44 / 68 D	0x64 / 100 d
0x25 / 37 %	0x45 / 69 E	0x65 / 101 e
0x26 / 38 &	0x46 / 70 F	0x66 / 102 f
0x27 / 39 '	0x47 / 71 G	0x67 / 103 g
0x28 / 40 (	0x48 / 72 H	0x68 / 104 h
0x29 / 41)	0x49 / 73 I	0x69 / 105 i
0x2a / 42 *	0x4a / 74 J	0x6a / 106 j
0x2b / 43 +	0x4b / 75 K	0x6b / 107 k
0x2c / 44 ,	0x4c / 76 L	0x6c / 108 l
0x2d / 45 -	0x4d / 77 M	0x6d / 109 m
0x2e / 46 .	0x4e / 78 N	0x6e / 110 n
0x2f / 47 /	0x4f / 79 O	0x6f / 111 o
0x30 / 48 0	0x50 / 80 P	0x70 / 112 p
0x31 / 49 1	0x51 / 81 Q	0x71 / 113 q
0x32 / 50 2	0x52 / 82 R	0x72 / 114 r
0x33 / 51 3	0x53 / 83 S	0x73 / 115 s
0x34 / 52 4	0x54 / 84 T	0x74 / 116 t
0x35 / 53 5	0x55 / 85 U	0x75 / 117 u
0x36 / 54 6	0x56 / 86 V	0x76 / 118 v
0x37 / 55 7	0x57 / 87 W	0x77 / 119 w
0x38 / 56 8	0x58 / 88 X	0x78 / 120 x
0x39 / 57 9	0x59 / 89 Y	0x79 / 121 y
0x3a / 58 :	0x5a / 90 Z	0x7a / 122 z
0x3b / 59 ;	0x5b / 91 [	0x7b / 123 {
0x3c / 60 <	0x5c / 92 \	0x7c / 124 |
0x3d / 61 =	0x5d / 93]	0x7d / 125 }
0x3e / 62 >	0x5e / 94 ^	0x7e / 126 ~
0x3f / 63 ?	0x5f / 95 _	

Control characters:

0x00	\0	null character
0x01		start of header
0x02		start of text
0x03		end of text
0x04		end of transmission
0x05		enquiry
0x06		acknowledge
0x07		bell
0x08	\b	backspace
0x09	\t	horizontal tabulation
0x0a	\n	line feed (LF)
0x0b	\v	vertical tabulation
0x0c	\f	form feed
0x0d	\r	carriage return (CR)
0x0e		shift out
0x0f		shift in
0x10		data link escape
0x11		device control #1
0x12		device control #2
0x13		device control #3
0x14		device control #4
0x15		negative acknowledgement
0x16		synchronous idle
0x17		end of transmission block
0x18		cancel
0x19		end of medium
0x1a		substitute
0x1b	(\e)	escape
0x1c		file separator
0x1d		group separator
0x1e		record separator
0x1f		unit separator
0x7f		delete

Details

Die Verwendung aller ASCII-Zeichen in den Sprachen C und C++ ist auf der Zeichen-Seite detailiert aufgeführt.

Ursprünglich entstand der ASCII-Standard aus der ISO 646 Zeichencodierung, welche auch als *"Invariant Code Set"* bekannt ist. ASCII entspricht der US-Erweiterung von ISO 646 und ist somit

programmiert werden können. Heutzutage kann jedoch der ASCII-Standard grundsätzlich vorausgesetzt werden.

ASCII basiert zwar auf einem der ältesten Standards in der Computerindustrie, doch er ist weltweit in nahezu jedem System eingebaut, vom Supercomputer bis hin zur Waschmaschine. ASCII hat sich bis heute bewährt und wird als universelles Rohformat für Textdateien angesehen. Aus diesem Grund werden auch heute noch Programmcode, Logfiles oder auch beispielsweise wissenschaftliche Rohdaten in ASCII-Dateien geschrieben. ASCII-Dateien haben weder Formatierung noch Bilder, Grafiken, Tabellen, Schriften, usw. Viele Programme bieten Schnittstellen für den Import und Export von ASCII-Dateien an.

ASCII ist eine 7-Bit-Codierung, wobei alle 128 Zeichen einen definierten Wert haben. Ein Byte besteht heutzutage normalerweise aus 8 Bits. Somit werden die 7 Bits der ASCII-Codierung in 8 Bits hineincodiert. Das überzählige (höchstwertige) Bit ist nicht definiert und wird unterschiedlich genutzt. Dadurch ergeben sich unterschiedliche Zeichenbelegungen, was als *"Encoding"* bezeichnet wird. Heutzutage hat sich UTF-8 als Standard-Encoding durchgesetzt. Je länger je seltener finden sich andere Encodings wie beispielsweise die Standard-Encodings für Windows und MacOS. Siehe unten.

Die Zeichen 0 - 31 und das Zeichen 127 hatten zu früheren Zeiten insbesondere die Aufgabe, die Ein- und Ausgabe von Text zu steuern und werden deswegen *"Steuerzeichen"* (control characters) genannt. Die meisten Steuer-Zeichen werden heutzutage nicht mehr benutzt, einige davon sind jedoch nach wie vor bei bei der Arbeit mit Strings oder allgemein in der Programmierung von grosser Wichtigkeit. Die wichtigsten Steuer-Zeichen sowie einige weitere Zeichen können in C und C++ mittels Escape-Sequenzen umschrieben werden.

Die Zeichen 32 bis 126 haben eine visuelle Representation und werden deswegen *"druckbare Zeichen"* (printable characters) genannt. Die druckbaren Zeichen beinhalten zum einen die arabischen Ziffern, lateinische Gross- und Kleinbuchstaben sowie eine Auswahl an Satzzeichen und sonstigen Zeichen, die insbesondere beim Programieren verwendet werden. In Textdateien können alle druckbaren Zeichen bedenkenlos auftreten. Sie haben im Gegensatz zu den Steuerzeichen keinerlei besondere Bedeutung sondern stellen genau nur das Zeichen dar, dass hier beschrieben ist. Beim Zeichen 32 handelt es sich um den Leerschlag (Leertaste).

Newlines

Ein Zeilenumbruch (auf Englisch *"Line break"*, im Programmier-Alltag jedoch häufiger *"Newline"*) wird je nach System anders codiert:

```
Unix:     LF     \n
Windows:  CR+LF  \r\n
```

Es gibt noch andere Systeme, welche wiederum andere Kombinationen verwenden. Für die alltägliche Programmierung sind heutzutage jedoch im Allgemeinen nur noch die zwei Zeilenenden-Versionen *"Unix"* und *"Windows"* wichtig.

Es ist zu beachten, dass jede Zeilenende-Variante unabhängig vom Encoding auftreten kann und teilweise innerhalb einer einzigen Datei sogar durchmischt auftritt. Leider hat sich auch bei modernen Format-Standards noch keine Variante durchsetzen können und erschwert nach wie vor das Programmieren. Probleme können beispielsweise auftreten, wenn in C eine Datei mittels `fopen()` im ASCII-Modus geöffnet wird. Hierbei werden die Zeilenenden automatisch umkonvertiert, was beispielsweise bei Index-Berechnungen zu unterschiedlichen Ergebnissen auf

oder als ein einziges Zeichen gezählt werden soll.

UTF-8 Encoding

Durch die 7-Bit Codierung von ASCII können nur wenige Zeichen abgebildet werden. Dieses Problem wurde durch eine Erweiterung auf 8 Bits nur leicht entschärft, und schuf gleichzeitig neue Probleme bei der Kompatibilität verschiedener Systeme welche noch bis heute auftreten.

Durch die Einführung von Unicode wurde ein Standard mit über einer Million Zeichen geschaffen. Diese Menge an Zeichen ist ausreichend für eine Vielzahl von Zeichen wie beispielsweise die griechischen Buchstaben, asiatische Schriftzeichen, Piktogramme, mathematische Symbole, Musiknoten und vieles mehr. Unicode wurde aufgrund der breiten Abdeckung auch als Universal Character Set (UCS) bezeichnet.

Unicode ist mittlerweile weit verbreitet und der Standard von vielen Systemen. Ein Problem, welches Unicode selbst jedoch nicht löste war die Kompatibilität zum weltweit akzeptierten ASCII-Standard. Unicode definiert momentan 21 Bits, was mit den 7 Bits von ASCII nicht vereinbar ist. Um diesem Problem entgegenzukommen wurde das Encoding UTF-8 entworfen, welches ASCII aufwärtskompatibel zu Unicode macht.

UTF-8 ist die Abkürzung für 8-Bit-UCS-Transformation-Format und ist sowohl fähig, alle Zeichen des Unicodes darzustellen, als auch die 7-Bit-ASCII-Zeichen ohne Konvertierung abzubilden. Die 7 Bits eines ASCII-Zeichens wurden in den gängigen Systemen stets mit 8 Bits gespeichert, wobei das führende Bit mit 0 gesetzt war. UTF-8 hat diese Eigenschaft übernommen und gleichzeitig definiert, dass bei Auftreten einer 1 beim führenden Bit es sich nicht mehr um ein ASCII-Zeichen handelt, sondern um ein Unicode-Zeichen.

Die nicht-ASCII-Zeichen werden bei UTF-8 mittels Hintereinanderreihung mehrerer Bytes realisiert, wobei das erste Byte eines Zeichens als das Start-Byte gilt. Je nachdem, wie die führenden Bits des Start-Bytes gesetzt sind, können eine unterschiedliche Anzahl an Folge-Bytes hinzukommen. Eine solche Codierung wird auch *"Multibyte-Codierung"* genannt. Das Schema für die Umwandlung ist in folgender Tabelle dargestellt:

```
0xxxxxxx                                ASCII-Zeichen
110xxxxx 10xxxxxx                       Unicode U+00080 bis U+007ff
1110xxxx 10xxxxxx 10xxxxxx              Unicode U+00800 bis U+0ffff
11110xxx 10xxxxxx 10xxxxxx 10xxxxxx     Unicode U+10000 bis U+1ffff
```

Die x bezeichnen dabei beliebige Bits, welche bei der Umwandlung in Unicode direkt aneinandergereiht werden und so die Zeichen ergeben, wie sie in der rechten Spalte aufgelistet sind. Dabei ist zu beachten, dass niederwertige Zeichen theoretisch auf unterschiedliche Weise codiert werden könnten. Der UTF-8 Standard schreibt jedoch vor, dass nur die Codierung mit minimalem Platzverbrauch zulässig ist.

Die spezielle Codierung mit den führenden Bits erlaubt es zudem, ein Byte direkt als Start- oder aber als ein Folge-Byte zu identifizieren. Dies ist insbesondere wichtig bei der Wiederherstellung von defekten Daten oder aber auch beim Zählen der Anzahl Zeichen in einem String.

Mittels UTF-8 werden die in der westlichen Welt häufig verwendeten Zeichen automatisch mit weniger Bytes gespeichert, was in kleineren Dateien resultiert. Der weitaus wichtigere Vorteil dieser Codierung ist jedoch, dass bestehende ASCII-Dateien direkt als UTF-8 Dateien interpretiert werden können. Aus diesem Grund hat sich UTF-8 weltweit verbreitet und verhalf indirekt auch zur Verbreitung von Unicode.

Windows Encoding

Window benutzte früher je nach Sprachregion verschiedene Encodings. Das am weitesten verbreitete war Windows-1252 Western European, eine Erweiterung des Encodings ISO-8859-1, besser bekannt als Latin-1. Die Encodings Windows-1252 Western European und ISO-8859-1 unterscheiden sich in den Zeichen 128 - 159, welche unter der ISO-8859 Klasse als nicht definiert gelten. Da diese Zeichen nur selten gebraucht werden, wurden fälschlicherweise die beiden Encodings oft vermischt.

Die Angabe ISO-8859-1, beziehungsweise Latin-1, ist heutzutage weitaus verbreiteter als Windows-1252 und wurde sogar für die Zeichenbelegung der Zeichen 128 - 255 in Unicode übernommen. Dennoch sind nach wie vor viele Dateien in Windows-1252 codiert, weswegen beispielsweise gemäss dem neuen HTML5-Standard das Encoding ISO-8859-1 als Windows-1252 interpretiert werden soll.

Nebst ISO-8859 gibt es noch den Standard ISO 8859 (kein Bindestrich). Dieser definiert ebenfalls keine druckbaren Zeichen für 128 - 159, ersetzt diese jedoch durch zusätzliche Steuerzeichen.

MacOS Encoding

Das Macintosh-System vor der Version X benutzte ein Encoding, welches unter dem Namen MacRoman bekannt war. Die Zeichentabelle kann bei anderen Quellen nachgelesen werden. In Mac OS X jedoch wurde dieses Encoding durch die Einführung von Unicode nach und nach verdrängt. Heutzutage werden Dateien grundsätzlich in UTF-8 abgespeichert.