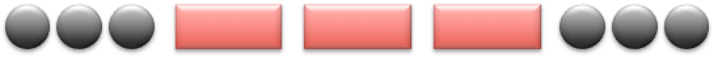
Modul 114: Zeichencodierung

Zeichencodierung vor dem Computerzeitalter

Alte Codierungen für Nachrichten sind z.B. ägyptische Hieroglyphen und indianische Rauchzeichen. Diese Codierung hat allerdings nicht einzelne Buchstaben codiert, sondern Begriffe.

Die erste Codierung für einzelne Buchstaben könnte der Morsecode gewesen sein. Dieser kennt für jeden Buchstaben (und alle Ziffern sowie Sonderzeichen) eine bestimmte Reihenfolge von kurzen und langen Signalen. Damit konnte mit einem unterbrechbaren Stromsignal oder per Lichtsignal Nachrichten übertragen werden.

Der bekannteste Morsecode ist sicher SOS: 

Standard ASCII Code S. 413

1963 wurde der "American Standard Code for Information Interchange" durch die "American Standards Association" definiert. Die einfache Idee war es, für jedes benötigte Zeichen eine Zahl zu definieren, welche dann (binär) codiert werden kann. Es wurden 7 Bits verwendet bzw. die Zahlen zwischen 0 und 127. Die 128 Zeichen enthielten folgende Zeichen:

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

(Quelle: ascii-table.com)

Bemerkenswert sind die Zeichen 0-31, dies sind nicht druckbare Zeichen und können im Fernmeldeverkehr gut genutzt werden. (Klingel, Bestätigung usw.) Für die Amerikaner hat das anfänglich gut ausgereicht, aber wie sollen wir damit das Wort "Küche" schreiben oder gar eine Email auf Französisch? Das System musst ausgebaut werden!

Erweiterter ASCII Code

Damit weitere länderspezifische Zeichen dargestellt werden können, wurde 1981 von IBM der erweiterte ASCII Code definiert. Dieser nutzt 8 Bits und kann 256 unterschiedliche Zeichen codieren. Die ersten 128 Zeichen wurden unverändert übernommen! 8 Bits sind sowieso praktischer, weil Prozessoren damals mit 8 Bits (1 Byte) rechnen konnten.
Eine komplette Liste der Zeichen finden wir unter ascii-code.com

ASCII ähnliche Normen

Eine international anerkannte Norm stellt der von der ISO (Internationale Standardisierungs Organisation) definierte **ISO 8859-1** Zeichensatz "**Latin 1**" dar. Dieser wurde für westeuropäische Sprachen zusammengestellt und hat nur wenige Abweichungen zum erweiterten ASCII. Insgesamt wurden 16 ISO 8859 Normen erstellt, darunter Osteuropäisch (-2), Arabisch (-6), Griechisch (-7), Thai (-11) usw.

Microsoft hat mit dem Windows Betriebssystem zwar auf die ISO normierten Zeichentabellen aufgebaut, diese aber teilweise leicht angepasst. Aus dem ISO 8859-1 und ISO 8859-15 wurde "**Windows Codepage** 1252 Westeuropäisch" gebildet. Auch bei den Windows Codepages gibt es sprachspezifische Varianten, nämlich 14 Stück. (874 Thai, 1251 Kyrillisch, 1256 Arabisch usw.)
Die Windows Codepages werden oft **ANSI Zeichensatz** genannt. Diese Bezeichnung basiert allerdings nicht auf der Normierung durch das ANSI. (American National Standards Institute)

Die Unterschiede können beim Datenaustausch zwischen unterschiedlichen Systemen einen Zusatzaufwand generieren. In der Praxis gibt es zum Glück gute Werkzeuge. ***➔1***

Weitere Informationen zu den Unterschieden innerhalb der Zeichensätze finden Sie in den PDF Dateien:

Verschiedene Zeichensätze ASCII und ANSI - PCtipp.ch.pdf
ISO 8859-1 – Wikipedia.pdf

Unicode

Wie wir jetzt wissen, ist die Auswahl des richtigen Zeichensatzes massgebend für die richtige Darstellung der als Zahlen codierten Zeichen. Zudem müssen wir dem Empfänger die Information mitgeben welcher Zeichensatz verwendet wurde, denn das ist in einer Textdatei nicht hinterlegbar. Bei der deutschen Sprache haben wir noch Glück, die 8 Bit (256 Zeichen-Speicherplätze) reichen aus für alle verwendeten Zeichen. In anderen Sprachen reichen diese 8 Bit nicht aus.

Wie könnte man das System also erweitern?

- Idee1: Wir zählen, wie viele Zeichen es weltweit gibt und passen dann die Anzahl Bits an für einen neuen, allumfassenden Zeichensatz
- Idee2: Wir zählen die verschiedenen Zeichensätze und sagen vor jedem Zeichen welchen Zeichensatz wir gerade verwenden

Diskutieren Sie die Vor- und Nachteile der beiden Varianten! (5 Minuten)

Bei Unicode handelt es sich um eine Mischung der beiden Ideen. Einerseits wurde die Menge an Zeichen von 8 Bit auf 16 Bit (2 Byte) erweitert.

2^{16} = 65'536 mögliche Zeichen

Um den weitverbreiteten ASCII Standard nicht über den Haufen zu werfen, wurden die ersten 128 Zeichen aus diesem übernommen.

Um für die Zukunft garantiert gerüstet zu sein wurden zudem 17 Ebenen (Planes), wovon jede davon 2^{16} Zeichen aufnehmen kann! Das bedeutet einen Zeichenraum von **1'114'112** Stück!

Kann das ausreichen? Denken Sie an:

- Verschiedene Sprachen
- Sprachen, in denen das gleiche Zeichen in X Varianten vorkommt (mit Punkt, mit mehreren Punkten)
- Mathematische Symbole
- Blindenschrift
- Phonetische Schrift (zeigt die richtige Aussprache an)

Die Antwort lautet ja, es reicht. Teilweise wurden skurrile Zeichen und Schriftsysteme aufgenommen. (Schriften von längst ausgestorbenen Kulturen usw.)

Wie werden Unicode Zeichen codiert?

	Markierung	Ebene	Zeichen	Unicode
von bis	U+	00 - 10	0000 - FFFF	
Bsp: X	U+	00	0058	U+0058
Bsp: ∞	U+	00	221E	U+221E
Bsp: 	U+	1	F640	U+1F640

Die Zahlen sind im Hexadezimalsystem angegeben.
Aufgaben zum Unicode ➔*2*

UTF (Unicode Transformation Format)

Während Unicode wie eine Liste aller Zeichen fungiert, wird UTF verwendet diese Liste möglichst schlau für die jeweilige Anwendung abzubilden. Dabei sollte nicht zu viel Speicherplatz verwendet werden.

Es gibt folgende verwendete UTF Codierungen: (Kurzübersicht)

Codierung	Beschreibung
UTF-8	Die am häufigsten verwendete Codierung. (HTML, Mail) Die ersten 128 Zeichen entsprechen ASCII Variable Länge 1-4 Byte Platzsparend beim lateinischen Alphabet
UTF-16	Die älteste Abbildungsvariante von Unicode Es werden 2 Byte (16 Bit) zur Codierung verwendet (teilweise auch 4 Byte) Teilweise kürzer als UTF-8 (bei nicht lateinischen Alphabeten) Es gibt 2 Varianten (Little und Big Endian)
UTF-32	Fixe Länge von 32 Bit (4 Byte) Braucht immer am meisten Speicherplatz Einfach zu gebrauchen, da fixe Länge Es gibt 2 Varianten (Little und Big Endian)

UTF-8 Codierung

Als Basis für die Codierung dient immer der Wert des Unicode Zeichens. Je nach Anzahl Stellen (binär) des Unicode Zeichens werden unterschiedlich viele Bytes verwendet für die Codierung.

Unicode Zeichen	UTF-8 Codierung	
0000 0000 - 0000 $007F_{16}$	0xxxx $xxxx_2$	ASCII Zeichensatz 0-127
0000 0067_{16} $110\ 0111_2$	$0110\ 0111_2$	Bsp: g (7 Bit)
0000 0080 - 0000 $07FF_{16}$	110x xxxx 10xx $xxxx_2$	Unicode Zeichen wird von rechts her aufgefüllt.
0000 $00FC_{16}$ $1111\ 1100_2$	$1100\ 0011\ 1011\ 1100_2$ $C3\ BC_{16}$	Bsp: ü (8 Bit)
0000 0800 - 0000 $FFFF_{16}$	1110 xxxx 10xx $xxxx_2$	Unicode Zeichen wird von rechts her aufgefüllt
0000 2623_{16} $0010\ 0110\ 0010\ 0011_2$	$1110\ 0010\ 1001\ 1000\ 1010\ 0011_2$ $E2\ 98\ A3_{16}$	Bsp: Biohazard 

(graue Zellen = Beispiele)

UTF-16 Codierung

Die Herstellung der UTF-16 Codierung funktioniert anders als bei UTF-8. Es gibt unterschiedliche Codierungen je nachdem in welcher *Ebene* sich das Zeichen befindet.

Unicode Zeichen	UTF-16 Codierung (Bits)	
0000 0000 - 0000 FFFF$_{16}$	xxxx xxxx xxxxx xxxx	Unicode Ebene 0 -> 2 Byte
0000 0067$_{16}$ 0110 0111$_2$	0067$_{16}$ 0000 0000 0110 0111$_2$	Bsp: g (Ebene 0)

0001 0000 - 0010 FFFF$_{16}$	1110 xxxx 10xx xxxx	Unicode Ebene 1 Berechnung: Unicode - 10000$_{16}$ Aufteilung auf 2 Byte
Unicode: 0001 F36B$_{16}$ Berechnung: - 0001 0000$_{16}$ F36B$_{16}$ 0000 1111 0011 0110 1011$_2$ -	1101 1000 0011 1100$_2$ 1101 1111 0110 1011$_2$ entspricht: D83C DF6B$_{16}$	Bsp: Schokolade 

Byte Reihenfolge
In der Praxis kommen zwei Darstellungsarten für UTF-16 vor, diese unterscheiden sich lediglich in der Reihenfolge:

Big Endian	Little Endian	
00 67$_{16}$	67 00$_{16}$	Bsp: g (Ebene 0)
D83C DF6B$_{16}$	DF6B D83C$_{16}$	Bsp: Schokolade 

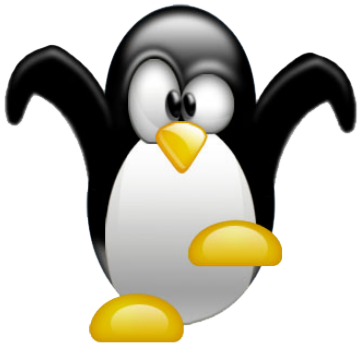
UTF-32 Codierung

Die Herstellung der UTF-32 Codierung ist denkbar einfach, es wird einfach der Code aus Unicode übernommen. Die Länge ist fix auf 32 Bits definiert.

Unicode Zeichen	UTF-32 Codierung (Bits)	
0000 0067$_{16}$ 0110 0111$_2$	0000 0067$_{16}$ 0000 0000 0000 0000 0000 0000 0110 0111$_2$	Bsp: g (Ebene 0)
Unicode: 0001 F36B$_{16}$	0000 0000 0000 0001 1111 0011 0110 1011$_2$	Bsp: Schokolade 

Auch bei der UTF-32 Codierung gibt es die beiden Varianten Big bzw. Little Endian. ➔*3*

Weiterführende Themen für Fortgeschrittene



Welche weiteren Codierungen kennen Sie?

- HTML Codierung, zum Beispiel €
- Unicode in HTML
- Domainnamen mit Umlauten, IDN Codierung

Erstellen Sie eine allgemeine Erklärung sowie 2 Beispiele zu den obigen Codierungsverfahren!

Was ist BOM?

Aufgabenteil

Installieren Sie ein Hilfsprogramm zur Analyse von Inhalten von Dateien und nutzen Sie dieses für die Erarbeitung der Lösungen dieses Arbeitsblattes.
Windows: WinHex, XVI32 oder ähnlich
Linux: Okteta oder ähnlich

1) 8 Bit Zeichensätze

Analysieren Sie die Dateien "Textdatei1.txt" und "Textdatei2.txt", welche Zeichensätze wurden verwendet?

Windows Tools:
Entdecken Sie die Möglichkeiten des Programms "Zeichentabelle". Welche Zeichensätze können Sie auswählen?

Installieren Sie das Programm "Notepad++" und schauen Sie die möglichen Codierungsarten an. (Menu -> Kodierung -> Zeichensatz).

Linux Tools:
Installieren Sie das Programm KCharSelect und entdecken Sie die Möglichkeiten.
Was bedeuten die grauen Felder?
Was ist der Unterscheid zwischen "Latin Grundschrift" und "Latin Ergänzung"?

Erstellen Sie einen kleinen Text mit dem Editor "Kate". Wie viele Codierungen können Sie beim Speichern auswählen? Welche Codierung wird standardmässig bei einem leeren Dokument gewählt?

2) Unicode

Informieren Sie sich über die Entstehung des Unicodes und die daraus entstandenen Versionen.

Verschaffen Sie sich einen Einblick (Überblick wäre zu viel verlangt) in die Zeichen des Unicode auf der Webseite http://unicode-table.com/de/
Spannend: Beim Scrollen durch die Zeichen werden rechts die involvierten Regionen angezeigt.
Wählen Sie auch einmal die Option "öffnen auf einer separaten Seite" an, um zu sehen wie die Blöcke eingeteilt sind!
Suchen Sie die bekannten Emoticon Zeichen.
Wählen Sie ein Zeichen aus und verändern Sie die Schriftart, was fällt auf?

Wie viele Ebenen werden aktiv genutzt? Welche Zeichen sind darin enthalten?

3) UTF

Was passiert wenn Sie mit einem nicht UTF-fähigen Gerät deutschen Text anzeigen, welcher in UTF codiert ist? Berücksichtigen Sie alle 3 Codierungsarten!

Was ist der Unterschied zwischen einem Smiley von "Wingdings" und "Emoji"?

In welchen UTF Codierungen können Sie anhand der Dateigrösse auf die Anzahl Zeichen rückschliessen?

Welche Codierungen können mit dem Windows Notepad verwendet werden?
Welche Codierungen können mit dem Linux Kate verwendet werden?

Was wurde hier codiert?

E2 9E AB E2 98 94 EF B8 8F 20 52 C3 A4 67 65

C2 A9 20 52 C3 AA 6E 65