

Unicode unter Linux

Johannes Weiß

Dieses Dokument steht unter der GNU Free Documentation License (GFDL)

Geschichtliches

- 1838 Morsekodex: eines Zeichen-Kodierungen
- 1874 Baudotcode: erster echter Binärkode (5-bit)
- 1880 Hollerith-Code auf Lochkarten (80 12-Bit Zeichen auf einer Karte)
- 1963 ASA (später ANSI) entwickelt ASCII (American Standard Code for Information Interchange)
wird weltweit als Standard akzeptiert, nur IBM geht mit EBCDIC
(Extended Binary Coded Decimal Interchange Code) einen eigenen Weg

- da ASCII nur die englische Sprache abdeckt, entwickelte sich eine Vielzahl von 8-Bit Erweiterungen
 - Microsoft Codepages
 - ISO-8859 (ISO-8859-[1-16], 8859-1 wird auch Latin-1 genannt)
 - AppleMac
 - ...

Probleme:

- keine Zeichen aus verschiedenen Zeichensätzen gleichzeitig in einem Dokument
- Austausch von Text über das Internet sehr aufwändig oder unmöglich

Entstehung von Unicode

- Mitte der 80er Jahre: zwei Ansätze zur Schaffung eines einheitlichen Zeichensatzes
 - ISO 10646 von der International Organization for Standardization
 - Unicode Projekt, hauptsächlich organisiert von US-Firmen
- Unicode Consortium setzt sich gegenüber der ISO durch
- ISO und Unicode Standards entsprechen sich von nun an in etwa (z.B. Unicode 3.0 und ISO 10646-1:2000)

Vorteile von Unicode

- statt vielen hundert Zeichensätzen nur noch ein einziger
- Zeichen aus allen Sprachen vereint
- genug Platz für eventuelle zukünftige Zeichen
- Austausch mit anderen Schriftkulturen stark vereinfacht
- Zeichen, die in mehreren Alphabeten vorkommen, werden nur einmal erfasst
- Kompatibilität mit älteren etablierten Standards
- Stabilität durch Abwärtskompatibilität zu älteren Versionen

Begriffserklärungen

- (abstrakter) Zeichensatz
eine Menge von Zeichen (Buchstaben, Ziffern, Interpunktionszeichen, Leerzeichen, Kontrollcodes, ...)
- Kodetabelle
eine Tabelle, in dem jedem Zeichen aus dem Zeichensatz eine Nummer (Kodeposition) zugewiesen wird
- Kodierungsformat
die Kodeposition wird als Folge von Kodeeinheiten (z.B. 1 Byte), also als Bitmuster, dargestellt. Die Kodelänge kann fest oder variabel sein.
- Kodierungsschema legt fest, wie das Kodierungsformat als Folge von Bytes repräsentiert wird. z.B. ein BOM (Byte Order Mark) am Anfang des Textes, um die Endianess festzulegen

Designprinzipien und Aufbau

- Zeichensatz heißt UCS (Universal Character Set)
- Koderaum (31-Bit) wird unterteilt in verschieden große Blöcke
- erster Block (0-127) entspricht ASCII, der zweite (128-255) Latin-1
- die ersten 16-Bit nennt man BMP (Basic Multilingual Plane)
- große Blöcke für private Benutzung reserviert
- zwei Arten von Zeichen: Gewöhnliche und Kombinierbare
- jedem Zeichen wird in der Kodetabelle neben der Kodeposition ein Name und semantische Informationen zugeteilt

Kodierungsformate

- UCS-4 (UTF-32)
jede Kodeposition wird in 4 Bytes (32 Bit) gespeichert, z.B.
'A' (65) : 00000000 00000000 00000000 01000001
'α' (945) : 00000000 00000000 00000011 10110001
 - Vorteil: leicht zu handhaben, weil feste Kodelänge
 - Nachteil: Latin-1 Texte werden 4-mal so groß wie bisher
- UCS-2 (ähnlich wie UTF-16)
jede Kodeposition wird in 2 Bytes (16 Bit) gespeichert, z.B.
'A' (65) : 00000000 01000001
'α' (945) : 00000011 10110001
 - Vorteil: ebenfalls leicht zu handhaben, weniger Platzverbrauch bei Latin-1 Sprachen
 - Nachteil: nur die ersten 65536 Zeichen können kodiert werden

UTF-8

alle Kodierungsformate mit Null-Bytes oder Slash ('/') an beliebigen Stellen in einem String nicht für POSIX-Systeme geeignet

UTF-8 schafft hier Abhilfe

- entwickelt 1992 von Ken Thompson
- Kodepositionen von 0 bis 127 (ASCII) werden mit einem Byte dargestellt
- Kodeposition über 127 werden mit 2-6 Bytes gebildet, wobei das erste Byte die Länge der gesamten Byte-Sequenz enthält
- Vorteile:
 - alle 2^{31} Zeichen können kodiert werden
 - 100% kompatibel mit ASCII
 - weniger Platzverbrauch, je gebräuchlicher die eingesetzten Zeichen sind (Englisch: 1 Byte, Westeuropäisch: 2 Bytes, BMP: 3 Bytes)

- Nachteil: Verarbeitung schwieriger, da dynamische Kodelänge

Bitmuster:

bis 2^7 : 0xxxxxxx
 bis 2^{11} : 110xxxxx 10xxxxxx
 bis 2^{16} : 1110xxxx 10xxxxxx 10xxxxxx
 bis 2^{21} : 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
 bis 2^{26} : 111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
 bis 2^{31} : 1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Beispiel:

'A' (65) : 01000001
 'α' (945) : **11001110 10110001** (945 = 01110 110001)

Linux-System auf Unicode umstellen

- benötigte Software
 - glibc 2.3 mit locale, wchar und iconv Unterstützung
 - XFree86 4.2, libncurses 5.3 mit wchar-Funktionen
- eventuell passende locale generieren:
`locale -a`
`localedef -v -c -i de_DE -f UTF-8 de_DE.UTF-8`
- Umgebungsvariable LANG setzen (in der `~/.bashrc` oder `/etc/profile`):
`export LANG=de_DE.UTF-8`
- Test: 'locale charmap' sollte UTF-8 ausgeben

Linux-Konsole

- kbd (Keyboard Utilities) 1.08 installieren
- Keymap laden, z.B.:
loadkeys
/usr/share/kbd/keymaps/i386/qwertz/de-latin1-noadkeys.map.gz
- eventuell bessere Konsolen-Schriftarten herunterladen
- für Consolen-Ausgabe und Tastatur Unicode-Modus aktivieren:
unicode_start [Schriftart]

XFree86

- Unicode Schriftarten herunterladen
(in neueren XFree86-Versionen enthalten)
- benötigte Zeilen in `/usr/X11R6/lib/X11/locale/compose.dir`:
`en_US.UTF-8/Compose de_DE.UTF-8`
`en_US.UTF-8/Compose: de_DE.UTF-8`
- benötigte Zeilen in `/usr/X11R6/lib/X11/locale/locale.dir`:
`en_US.UTF-8/XLC_LOCALE de_DE.UTF-8`
`en_US.UTF-8/XLC_LOCALE: de_DE.UTF-8`
- Einstellungen in der `~/.Xresources`:
`XTerm*utf8: 1`
`XTerm*font: -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10646-1`
`XTerm*VT100*font: -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10646-1`
`XTerm*VT100*wideFont: -misc-fixed-medium-r-normal-ja-13-125-75-75-c-120-iso10646-1`
`XTerm*VT100*boldFont: -misc-fixed-bold-r-semicondensed--13-120-75-75-c-60-iso10646-1`

Literatur

- The Unicode HOWTO
<http://www.tldp.org/HOWTO/Unicode-HOWTO.html>
- UTF-8 and Unicode FAQ for Unix/Linux
<http://www.cl.cam.ac.uk/~mgk25/unicode.html>
- Debian: Introduction to i18n
<http://www.debian.org/doc/manuals/intro-i18n>
- Unicode Project (FAQ, Technical Introduction, ...)
<http://www.unicode.org>
- A Short History of Character Sets
<http://www.gnomedia.com/cw/articles/article003.php>
- Wikipedia (Unicode, UTF-8, ASCII, ...)
<http://www.wikipedia.org>
- A Brief History of Character Codes
<http://tronweb.super-nova.co.jp/characcodehist.html>
- Unicode for developers
<http://www.joelonsoftware.com/articles/Unicode.html>
- Das Modell der strukturierten Dokumente
<http://www11.informatik.tu-muenchen.de/~brueggem/Vorlesungen/ep2000Winter/Skript/kurseinheit1.htm>