LU02c - Binäre Zeichencodes

Einführung

Ein Zeichencode dient zur Speicherung von Schriftzeichen (Buchstaben, Ziffern, Sonderzeichen, ...) in einem Computersystem. Da ein Computer nur Zahlen verarbeiten kann, wird jedem Schriftzeichen eine eindeutige Zahl (Code) zugeordnet. Dieser Code wird dann im Computer als binäre Zahl gespeichert.

Ein wenig Geschichte

Solche Zeichencodes gab es schon lange bevor es Computersysteme gab. Zum Beispiel wird im Morsealphabet jedem Schriftzeichen eine Folge von Punkten und Strichen zugeordnet. Beim Morsen übersetzt der Sender seinen Text den Morsecode. Dieser Morsecode kann dann mit kurzen und langen Lichtimpulsen übermittelt werden. Der Empfänger empfängt die Lichtimpulse und übersetzt diese zurück in den entsprechenden Buchstaben. Natürlich kann dies nur funktionieren, wenn Sender und Empfänger den gleichen Zeichencode verwenden.

Morsecode ausprobieren

Zeichencodes am Computer

Wenn ich beim Schreiben dieses Wiki-Artikels auf eine Taste tippe, geschieht folgendes:

- Die Tastatur sendet die Information, welche Taste gedrückt wurde als elektrischen Impuls.
- Der Computer speichert den entsprechenden Zeichencode im Hauptspeicher.
- Für die Anzeige am Bildschirm wird dieser Zeichencode als eine Kombination aus Strichen und Kurven angezeigt.

Der Computer benötigt den Zeichencode, damit er den gewünschten Buchstaben darstellen kann.

Wenn Sie diese Wikiseite aufrufen (was Sie offensichtlich gerade tun), wird:

- Eine HTML-Datei an Ihren Webbrowser übermittelt. Jedes Schriftzeichen in dieser Datei wird als binäre Zahl übermittelt.
- Ihr Computer interpretiert die HTML-Seite und übersetzt die binären Zahlen mit Hilfe des Zeichencodes in die gewohnten Schriftzeichen.

Verbreitete Zeichencodes

ASCII

- *American_Standard_Code_for_Information_Interchange*

- *IT-Handbuch, 16.1 Textdateien und Zeichensätze*

Der **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange ist eine Zeichencodierung für die Verarbeitung von Daten. Die meisten heutigen Zeichencodierungen basieren auf der ASCII-Zeichencodierung.

Jedem Zeichen wird ein Code aus 7 Bits zugeordnet, womit insgesamt 128 unterschiedliche Zeichen zur Verfügung stehen:

- 95 druckbare Zeichen
  - Alle Buchstaben des lateinischen Alphabets (a-z und A-Z).
  - Die zehn arabischen Ziffern (0 - 9).
  - Einige Satzzeichen und Sonderzeichen.
- 33 nicht druckbare Steuerzeichen, zum Beispiel:
  - Zeilenumbruch

Im ASCII-Code entspricht jedes Zeichen einem Byte. Das vorderste Bit wurde ursprünglich als Paritätsbit zum Erkennen von Fehlern genutzt:

- Man zählt wie oft ein 1-Bit vorkommt: 100 0001 ⇒ Anzahl = 2
- Ist die Anzahl gerade, so ist das Paritätsbit '0'.
- Ist die Anzahl ungerade, so ist das Paritätsbit '1'.

Dadurch konnte man erkennen, wenn ein einzelnes Bit falsch übermittelt wurde.

ASCII-Zeichentabelle

American_Standard_Code_for_Information_Interchange#Zusammensetzung

Internationale Zeichencodes

Der ASCII-Code enthält keine Umlaute oder Buchstaben mit Akzent (z.B. é, â). Deshalb wurden Erweiterungen des ASCII-Codes entwickelt, die alle 8 Bits nutzen und somit 256 verschiedene Zeichen zur Verfügung stellten. Dabei wurde jeweils darauf geachtet, dass die ersten 128 Codes ('00'x bis '7F'x bzw. 0 bis 127) dem ASCII-Code entsprechen. Diese Zeichencodes werden als **Codepage** bezeichnet.

Der Standard ISO 8859-1 definiert einen solchen erweiterten Code für die westeuropäischen Sprachen. Er versucht, möglichst viele Zeichen der westeuropäischen Sprachen abzudecken.

Nachteile

- Für jeden Sprachraum muss ein eigener Zeichencode verwendet werden.
- Das gleiche Zeichen hat je nach eingesetztem Zeichencode einen anderen Wert.

Entwicklung

Der Unicode wurde entwickelt, um den Anforderungen der verschiedenen Sprachen gerecht zu werden. Diese Zeichencodierung verwendet bis zu 32 Bit (4 Byte) pro Zeichen und könnte damit fast 4.3 Milliarden verschiedener Zeichen abbilden.

UTF-8 ist eine 8-Bit-Codierung die auf Unicode basiert. Gleichzeitig ist UTF-8 abwärtskompatibel zum ASCII-Code.

Unicode

Siehe Unicode

Unicode ist ein internationaler Standard für die Codierung von Zeichen. Das Ziel von Unicode ist es, jedem Zeichen aus jedem Sprachraum einen bestimmten Code zuzuordnen.

Jedem Zeichen ist im Unicode-Standart ein Codepunkt zugeordnet. Dieser Codepunkt wird wie folgt dargestellt:

- Am Anfang wird U+ vorangestellt.
- Danach folgt der Codepunkt in hexadezimaler Schreibweise mit mindestens 4 Stellen.

Zum Beispiel `U+0040` für das Zeichen @.

In modernen Computersystemen ist der Unicode Basis für die verwendeten Zeichencodierungen UTF-8 und UTF-16. Eine Textvariable wird also nicht im Unicode gespeichert.

Nicht ganz ernst gemeint:

Echte Programmierer kennen die Binärcodes aller Unicode-Zeichen, auswendig!

UTF-8

//Siehe auch UTF-8 und http://www.utf8-zeichentabelle.de/ //

Die Abkürzung steht für **U**niversal Character Set **T**ranslation **F**ormat. Als universeller Zeichencode hat UTF-8 vor allem im Internet eine grosse Bedeutung. So waren im Juli 2014 mehr als 80% [1] aller Webseiten mit UTF-8 codiert.

Bei der UTF-8-Codierung wird jedem Zeichen eine Bitkette von 8 bis 32 Bit (1 bis 4 Byte) zugeordnet. Somit könnten theoretisch fast 4.3 Milliarden Zeichen abgebildet werden. Die ersten 128 Zeichen (Indizes 0 - 127) sind identisch mit dem ASCII-Code. Dadurch lassen sich insbesondere englische Texte mit nur einem Byte Speicherplatz pro Zeichen gespeichert werden. Somit können selbst Programme, die nicht UTF-8-fähig sind, die Daten bearbeiten.

Codierung

1 Byte lange Codes

Die ersten 127 Zeichen der UTF-8-Codetabelle entsprechen den ASCII-Zeichen. Diese Zeichencodes sind jeweils 8 Bit lang und beginnen mit einen Bit '0'.

Bits	Bedeutung
0xxx xxxx	Start-Byte, Länge des Codes: 1 Byte

2-4 Byte lange Codes

Allen Zeichen die nicht Teil der ASCII-Codetabelle sind, wird ein 16 - 32 Bit langer Zeichencodes zugeordnet.

- Bei einem Code mit mehr als 8 Bit (1 Byte) beginnt jedes Byte mit Bit '1'.
- Das erste Byte eines solchen Codes wird als „Start-Byte“ bezeichnet.
    - Das Start-Byte beginnt mit der Bitfolge '110', '1110' oder '1111 0'.
    - Die Anzahl der Bits '1' am Anfang des Start-Bytes zeigt die Gesamtlänge des Codes an.
- Das zweite bis vierte Byte wird als „Folge-Byte“ bezeichnet.
    - Ein Folge-Byte beginnt immer mit Bit '10'.

Die ersten Bits innerhalb des Start- und Folge-Bytes dienen zur Unterscheidung der Art und Länge des Codes. Die restlichen Bits (unten mit x dargestellt) stellen den eigentlichen Wert des Zeichens dar:

Bits	Bedeutung
10xx xxxx	Folge-Byte
110x xxxx	Start-Byte, Länge des Codes: 2 Byte
1110 xxxx	Start-Byte, Länge des Codes: 3 Byte
1111 0xxx	Start-Byte, Länge des Codes: 4 Byte

Beispiele

Bitkette: 0101 0101

- Bit '0' an der ersten Stelle:
    - Es handelt sich um ein Start-Byte.
    - Der Zeichencode ist 1 Byte lang.
    - Die Bits an den Positionen 2-8 stellen den Wert dar.
- '1010101':
    - Entspricht '55'x bzw. 85
    - Anhand der UTF-8-Zeichentabelle sehen Sie, dass es der Buchstabe **U** ist.
    - In der ASCII-Zeichentabelle finden Sie ebenfalls den Buchstaben **U** an dieser Stelle.

Bitkette: 11100010 10000010 10101100

- Das erste Byte beginnt mit '1110':
  - Es handelt sich um ein Start-Byte.
  - Der Zeichencode ist 3 Byte lang.
  - Die Bits an den Positionen 5-8 gehören zum Wert.
- Das zweite Byte beginnt mit '10':
  - Es handelt sich um ein Folge-Byte.
  - Die Bits an den Positionen 3-8 gehören zum Wert.
- Das dritte Byte beginnt mit '10':
  - Es handelt sich um ein Folge-Byte.
  - Die Bits an den Positionen 3-8 gehören zum Wert

Wir haben somit 16 Bits (4 aus dem Start-Byte und je 6 aus den Folge-Bytes) die den Zeichencode darstellen: `0010 000010 101100`

1. Ordnen Sie die Zeichen von rechts nach links in 4er Gruppen an: `0 0010 0000 1010 1100`
2. Übertragen Sie jede 4er Gruppe in das hexadezimale System: `20 AC`
3. Anhand der UTF-8-Codetabelle finden Sie das Unicode-Zeichen **€**.

UTF-16

Die UTF-16-Kodierung wird oft zur internen Speicherung von Zeichen auf dem Computer verwendet. Zum Beispiel werden `char`-Variablen in Java im UTF-16-Code abgelegt.

Bei UTF-16 wird jedem Zeichen eine Bitkette von 16 oder 32 Bit (2 oder 4 Byte) zugeordnet.

EBCDIC

Siehe Extended_Binary_Coded_Decimals_Interchange_Code

Der **E**xtended **B**inary **C**oded **D**ecimals **I**nterchange **C**ode (sinngemäss „erweiterter Austauschcode für binär codierte Dezimalziffern“) wurde von IBM entwickelt. Die Entwicklung fand zeitlich parallel zum ASCII-Code statt.

Aus Zeitgründen konnte IBM für seine Grossrechner keine ASCII-kompatiblen Ausgabegeräte entwickeln. Daher wurde der bereits bestehende 6 Bit lange BCD-Code erweitert.

Im Gegensatz zu ASCII (7 Bit) verwendet EBCDIC 8 Bits zur Speicherung eines Schriftzeichens. Heute wird der EBCDIC-Code fast nur noch von Grossrechnern (z.B. IBM z/System) verwendet.

---

m114-A1F, m114-A1E

Marcel Suter

1)

Quelle: http://w3techs.com/technologies/history_overview/character_encoding

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m114/learningunits/lu02/binaerezeichencodes

Last update: **2026/01/28 21:12**