

Informationsdarstellung im Rechner

Dr. Christian Herta

15. Oktober 2005

Darstellung von Information im Computer

Alle Daten und Informationen, die im Computer verarbeitet werden, müssen in einer für den Computer verständlichen Form vorliegen und somit umgewandelt (*codiert*) werden.

Codierung: Als Codierung bezeichnet man die Umwandlung von Symbolen von einer Darstellung in eine andere.

Beispiel: Morsecode

A: · -

B: - · · ·

Im Computer werden Informationen stets binär (zweiwertig) codiert, d.h. sie werden als Abfolge von Nullen und Einsen repräsentiert.

Dies hat mit den physikalischen Aufbau realer Computer zu tun. Die beiden Zahlen entsprechen den physikalischen Zuständen (z.B. Spannung an bzw. aus).

Binäre Darstellung

Jede einzelne Stelle in der Abfolge von Nullen und Einsen bezeichnet man als **Bit** oder Binärzeichen.

Bit (Abk. für **binary digit**): Informationsgehalt einer einzelnen Ja-Nein Entscheidung (hier: entweder 0 oder 1).

Will man z.B. alle Buchstaben des Alphabets mit Bits codieren, muss man verschiedene Bits zu Bitgruppen zusammenfassen. Die Kombination der Werte einer Bitgruppe codiert die Information. Jedes hinzugefügte Bit zur Gruppe erhöht die Anzahl der codierbaren Zustände um den Faktor zwei.

Anzahl der Bits	Mögliche Zustände
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$

Bitgruppen

Als zweckmäßig haben sich Bit-Gruppen mit den Längen 8, 16 und 32 erwiesen. Diese werden als **Byte**, **Wort** und **Langwort** bezeichnet. Das Bit, das am weitesten rechts steht, wird als niederwertigstes Bit (least significant bit - LSB) bezeichnet. Das am weit links stehende Bit heißt höchst wertigstes Bit (most significant bit - MSB)

Name der Bitgruppe	Anzahl der Bits	Mögliche Zustände
Bit	1	2
Halbbyte(Tetrade)	4	16
Byte	8	256
Wort	16	65536
Langwort	32	4294967296

Beachten Sie die exponentielle Zunahme der Anzahl der Zustände mit der Anzahl der Bits.

Darstellung von Buchstaben und Zeichen

Standards zur Codierung von Zeichen

Die gebräuchlichste Form, Zeichen zu codieren, ist der **ASCII**-Code (American Standard Code for Information Interchange). ASCII beschreibt einen Sieben-Bit-Code. D.h. es können $2^7 = 128$ Zeichen mit den ASCII-Code dargestellt werden. Steuerzeichen (wie z.B.: Zeilenvorschub) sind im ASCII-Code ebenfalls erhalten.

Beispiel: Buchstabe A : 1000001

Durch Hinzunahme des 8-ten Bit des Bytes können nochmal 128 Zeichen dargestellt werden. Diese können für länderspezifische Zeichen genutzt werden. Dabei gibt es aber keinen einheitlichen Standard. Dies führt zu Inkompatibilitäten zwischen den verschiedenen Ländercodierungen bzw. Computersystemen (z.B. bei deutschen Umlauten).

Unicode

Lösung: **Unicode**

Aus [Wikipedia](#), der freien Enzyklopädie:

Unicode ist ein internationaler Standard, in dem langfristig für jedes sinntragende Zeichen bzw. Textelement aller bekannten Schriftkulturen und Zeichensysteme ein digitaler Code festgelegt wird. Er will das Problem der verschiedenen inkompatiblen Kodierungen in den unterschiedlichen Ländern beseitigen.

Die Speicherung und Übertragung von Unicode erfolgt in unterschiedlichen Formaten. Wobei [utf-8](#) am verbreitetsten ist.

Text-Dateien

(Text-)Daten können als Dateien (*Files*) im ASCII-Code bzw. Unicode auf einem Rechner gespeichert werden. Diese Dateien können ohne größere Schwierigkeiten zwischen verschiedenen Computersystemen ausgetauscht werden und dort angezeigt und weiterverarbeitet werden.

Dabei können jedoch Probleme bei der Handhabung des Zeilenendes auftreten. DOS und Windows verwendet zur Darstellung des Zeilenendes die Kombination Wagenrücklauf-Zeilenvorschub. Bei Apple ist die Reihenfolge umgekehrt. Unix und Linux nutzen nur den Zeilenvorschub.

Beispiele für Text-Dateien:
CSV, Emails, Webseiten.

Darstellung von Zahlen(-werten)

Unterscheidung zwischen einer Zahl als Text und einer Zahl als Zahlenwert.

Beispiel: ASCII-Code der Zahl -27: 00111101 00110010 00110111

Textdarstellung einer Zahl ist ungeeignet zum Rechnen.

Stellenwertprinzip:

Beispiel für Dezimalsystem (10er System) hat 10 Ziffern: 0,1,..9

Basis 10 ($|_{10}$):

$$362,5 = 3 * 100 + 6 * 10 + 2 * 1 + 5 * 0.1 =$$

$$3 * 10^2 + 6 * 10^1 + 2 * 10^0 + 5 * 10^{-1}$$

Dualsystem, Oktalsystem und Hexadezimalsystem

Für eine Binärzahlenfolge ist das **Dualsystem** geeignet.

Basis 2 ($|_2$):

$$362,5|_{10} \equiv 1*256+0*128+1*64+0*32+0*16+1*8+0*4+1*2+0*1+1*1/2 =$$

$$1*2^8+0*2^7+1*2^6+0*2^5+0*2^4+1*2^3+0*2^2+1*2^1+0*2^0+1*2^{-1} = 101001010,1$$

Bei großen Zahlen wird das Dualsystem schnell unübersichtlich. Daher verwendet man auch das Oktalsystem (Basis 8) und das Hexadizimalsystem (Basis 16) (Machen Sie sich Gedanken: Warum gerade diese?).

Zahlensystem	Symbole
Dualsystem	0,1
Oktalsystem	0,1,2,3,4,5,6,7
Hexadizimalsystem	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Beispiel: $14|_{10} \equiv 1110|_2 \equiv 16|_8 \equiv E|_{16}$

Umwandlung zwischen den Zahlensystemen

- Zwischen Systemen, deren Basis ein Vielfaches von 2 ist: Durch Gruppieren (eventuell über Dualsystem).
- Dual-, Oktal- oder Hexadezimal-System ins Dezimalsystem z.B. mit dem Horner-Schema
- Vom Dezimalsystem ins Dual-, Oktal- oder Hexadezimal-System: z.B. mit der Umkehrung des Horner-Schema (eventuell über Dualsystem)

Negative Dualzahlen

Darstellung des Vorzeichen mit einem Bit

Konvention z.B.:

Vorzeichen wird im *most significant Bit* codiert.

Vorrangestellte Null - Positive Zahlen

Vorrangestellte Eins - Negative Zahlen

Beispiel:

$$+5|_{10} \equiv 0101|_2$$

$$-5|_{10} \equiv 1101|_2$$

Beachte: Die Null kommt doppelt vor (+0, -0).

Zweierkomplement

Um Rechner-Schaltungen einfach zu halten soll, wurde ein Verfahren entwickelt, die Subtraktion von Zahlen auf eine Addition zurückzuführen.

Dies führt auf die Darstellung von negativen Dual-Zahlen im **Zweierkomplement**.

Erzeugen des Zweierkomplements um negative Zahlen darzustellen:

1. Aufschreiben der Zahl als positive Dualzahl (mit vorrangestellter Null für positives Vorzeichen)
2. Inversion aller Stellen ($0 \rightarrow 1$; $1 \rightarrow 0$)
3. Addition von 1 (\rightarrow nur eine Null). (wird dieser Schritt weggelassen spricht man vom Einerkomplement)

Bei Rechnungen ist zu beachten, dass man bzw. der Computer üblicherweise mit einer festen Stellenzahl operiert. Bei Bedarf sind also eventuell führende Nullen (vor der Inversion) aufzufüllen.

Reele Zahlen - Festkommadarstellung

Problem: Position des Kommas?

Festkommadarstellung:

Vereinbarung eines konstanten Faktors mit dem der Ganzzahlenwert multipliziert wird. D.h. Multiplikation des Ganzzahlenwertes mit dem Faktor 2^{-k} ; k -Stellen stehen so hinter dem Komma.

Reele Zahlen - Gleitkommadarstellung

Gleitkommadarstellung

Zahl Z soll dargestellt werden: $Z = X * B^Y$

X: Mantisse

Y: Exponent

Nur X, Y werden im Rechner dargestellt: [Mantisse X, Exponent Y].

Folgende Angaben sind zur Eindeutigkeit notwendig:

- Anzahl der Stellen der Mantisse und des Exponenten
- Darstellungsweise der Mantisse und des Exponenten (z.B. ob diese im Zweierkomplement vorliegen)
- Lage des Kommas bei der Mantisse
- Die Gleitkommabasis B

Zusammenfassung

- Daten können im Computer unterschiedlich repräsentiert werden:
 - Buchstaben und andere Zeichen können in Bitgruppen in unterschiedlichen Codierungen dargestellt werden (ASCII, Unicode).
 - Alphanumerische Daten (zur Weiterverarbeitung) werden in verschiedenen Formaten (je nach Typ und Aufgabe) gespeichert.
 - * Basis hierzu ist das Dualsystem, Oktal und Hexadezimalsystem.
 - * Zweierkomplement, Festkomma-Darstellung, Gleitkomma-Darstellung