

LU01c - Zahlensysteme

Überblick

Vor etwa 30'000 Jahren begannen die Menschen zu zählen. Zu Beginn nutzte man dazu Gegenstände wie Steine oder Hölzchen. Mit der Zeit entstand das Bedürfnis, die Zahlen festzuhalten.

Die einfachsten Zahlensysteme bestanden einfach aus so vielen Zeichen, wie die Anzahl war. Zum Beispiel wurden 15 Kerben in einen Stein geritzt. Ein solches System ist bei grösseren Zahlen sehr aufwändig und unübersichtlich.

Ein verbessertes System nutzte spezielle Symbole, die einem bestimmten Wert entsprachen. Die Zahl entstand durch Addition der Werte aller Symbole.

Eine wesentliche Verbesserung brachten die Stellenwertsysteme, welche ursprünglich in Indien entwickelt wurden. Bei diesen Zahlensystemen hängt der Wert eines Symbols von seiner Position innerhalb der Zahl ab. Unser Dezimalsystem ist ein solches Stellenwertsystem.

Zahlen in der Informatik

Ein Computer kennt nur die beiden Zustände „kein Strom“ / „Strom“. Diese beiden Zustände bilden wir mit den Ziffern 0 und 1 ab. Somit ist das Binärsystem (Dualsystem oder 2er-System) die Grundlage der Informatik.

Stellenwertsysteme

Alle Stellenwertsysteme basieren auf den gleichen Regeln. Das Wichtigste ist dabei die Basis:

- Die Basis legt fest, wie viele Ziffern-Symbole das System kennt.
- Welcher Faktor für den Wert der Stellen angewandt wird.

Um die Systeme zu unterscheiden, schreiben Sie immer die Basis tiefgestellt neben die Zahl.



- Binäre Zahlen können auch durch ein **b** nach der Zahl identifiziert werden.
- Hexadezimale Zahlen werden häufig durch ein **x** nach der Zahl identifiziert.

Eine Zahl ohne Angabe des Systems ist als Dezimalzahl zu interpretieren. Also

- $1A7_{16}$ oder $1A7x$
- 10101011_2 oder $10101011b$



Dezimalsystem

Dezimalsystem

Wir betrachten zunächst das Ihnen vertraute Dezimalsystem im Detail. Daraus können wir Schlüsse für andere, in der Informatik relevante Systeme ziehen.

Das Dezimalsystem (10er System),

- kennt 10 verschiedene Zahl-Symbole,
 - In Europa verwenden wir Symbole die aus Indien über den arabischen Raum kamen: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
Sie werden deshalb als *arabische Ziffern* bezeichnet.
 - Je nach Gebiet werden andere Symbole verwendet.
- hat die Basis 10 für die Stellen einer Zahl.
 - Die Stelle direkt vor dem Dezimalpunkt hat den Wert 10^0 bzw. 1.
 - Nach links hat jede Stelle den zehnfachen Wert der davorliegenden Stelle.
 - Nach rechts hat jede Stelle einen Zehntel des Werts der davorliegenden Stelle.

Ziffer	7	3	4	.	2	5
Wert	10^2	10^1	10^0		10^{-1}	10^{-2}
	100	10	1		0.1	0.01

- Jede Ziffer wird mit dem Wert seiner Position multipliziert.
- Alle Produkte (Ziffer * Wert) werden addiert.

$$734.25 = 7*100 + 3*10 + 4*1 + 2*0.1 + 5*0.01$$

Binärsystem

Das Binärsystem oder Dualsystem verwendet die Basis 2. Daraus ergibt sich:

- Es gibt 2 unterschiedliche Symbole für die Ziffern (z.B. 0 und 1)
- Der Wert der Positionen verdoppelt sich von rechts nach links, bzw. halbiert sich von links nach rechts.

Ziffer	1	1	0	1	0	.	1	0	1
Wert	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
	16	8	4	2	1		1/2	1/4	1/8

$$11010.101_2 = 1*16 + 1*8 + 0*4 + 1*2 + 0*1 + 1*(1/2) + 0*(1/4) + 1*(1/8)$$

Oktal

Das Oktalsystem verwendet die Basis 8.

- 8 unterschiedliche Symbole für die Ziffern (0, 1, 2, 3, 4, 5, 6, 7)
- Der Wert der Positionen erhöht sich um den Faktor 8 (bzw. wird um den Faktor 8 verkleinert).

Ziffer	4	0	6	.	4	1
Wert	8^2	8^1	8^0		8^{-1}	8^{-2}
	64	8	1		1/8	1/64

$$406.41_8 = 4 \cdot 64 + 0 \cdot 8 + 6 \cdot 1 + 4 \cdot (1/8) + 1 \cdot (1/64)$$

Im Oktalsystem werden jeweils 3 binäre Stellen (Bits) zusammengefasst. Dadurch lassen sich die Zahlen kompakter schreiben. Gleichzeitig ist die Umrechnung zwischen Binärsystem und Oktalsystem relativ einfach.

Hexadezimal

Das Hexadezimal-System verwendet die Basis 16.

- 16 unterschiedliche Symbole für die Ziffern (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
- Der Wert der Positionen erhöht sich um den Faktor 16 (bzw. wird um den Faktor 16 verkleinert).

Ziffer	3	A	2	.	0	C
Wert	16^2	16^1	16^0		16^{-1}	16^{-2}
	256	16	1		1/16	1/256

$$3A2.0C_{16} = 3 \cdot 256 + 10 \cdot 16 + 2 \cdot 1 + 0 \cdot (1/16) + 12 \cdot (1/256)$$

Das Hexadezimalsystem (kurz Hex) fasst jeweils 4 binäre Stellen (Bits) zusammen. Damit können Sie ein Byte (8 Bits) mit zwei hexadezimalen Ziffern abbilden.



Zum Schmunzeln: Echte Programmierer haben Mühe zwischen Halloween (31. Oktober) und Weihnachten (25. Dezember) zu unterscheiden. Denn $31_{\text{okt}} = 25_{\text{Dez}}$.

Fließkommazahlen

Gleitkommazahl

Die Fließkommazahlen oder Gleitkommazahlen wurden entwickelt, um sehr grosse bzw. sehr kleine Zahlen effizient darzustellen. Zum Beispiel beträgt der Abstand zwischen Erde und Sonne ca. 148300000000 Meter. Als Fließkommazahl geschrieben: $1.483 \cdot 10^{11}$ Meter.

Dabei wird neben den eigentlichen Ziffern auch noch ein Exponent als Faktor für den Wert der Ziffern gespeichert. Eine Fließkommazahl setzt sich also aus zwei Teilen zusammen:

- Die Ziffernstellen (Mantisse) z.B. 1.483
- Der Exponent mit Basis 10, z.B. $10^{11} = 11$

Um eine Fließkommazahl ins Dezimalsystem zu übertragen, multiplizieren Sie die Ziffernstellen mit 10^{Exponent} .

Zum Beispiel:

- $1.4356 * 10^3 = 1435.6$
- $0.756 * 10^{-4} = 0.0000756$

Computerprogramme

Eine Variable in einem Computerprogramm hat immer einen festgelegten Speicherbedarf und kann somit bestimmte Anzahl Stellen speichern. Bei einer Variable mit einem festen Dezimalpunkt gehen Ihnen irgendwann die Stellen aus. Selbst eine Variable mit 64-Bit Speicherplatz kann „nur“ Zahlen mit rund 19 dezimalen Stellen speichern.

Bei einer Fließkommazahl kann der Wertebereich dynamisch angepasst werden. Auch eine Variable vom Datentyp `double` hat nur 64-Bit Speicherplatz zur Verfügung. Damit können aber Zahlen mit bis zu 308 Stellen vor bzw. 324 Stellen nach dem Dezimalpunkt gespeichert werden.

Übung

Sie können das Problem mit den Festkommazahlen in einem einfachen Java-Programm ausprobieren. Kopieren Sie dazu den Code in den Online Java Compiler

<https://www.jdoodle.com/online-java-compiler/>:

```
public class MyClass {
    public static void main(String args[]) {
        byte meineZahl = 0;

        while (meineZahl >= 0) {
            meineZahl++;
            System.out.println(meineZahl);
        }
    }
}
```

Frage: An welchem Punkt sind der Variable die Stellen ausgegangen?

Wieso Java und nicht Python



Python kennt keine maximale Grösse für Ganzzahlen. Deshalb lässt sich dieses Verhalten nicht in Python ausprobieren.

Exponentiale Schreibweise

Die exponentiale Schreibweise dient der vereinfachten Notation einer Fließkommazahl. Dabei wird zwischen den Ziffernstellen und dem Exponenten der Buchstabe **E** eingefügt.

Lesen Sie die Zahl 1.4356E3 indem Sie das **E** durch „mal 10 hoch“ ersetzen. Sie erhalten dann „1.4356 mal 10 hoch 3“ oder „ $1.4356 \cdot 10^3$ “.

Vereinfachte "Umrechnung"

Anstatt die Zahlen umzurechnen, können Sie auch den Dezimalpunkt verschieben. Die Regel dazu lautet:

1. Verschiebe den Dezimalpunkt um so viele Stellen nach rechts oder links, wie der Exponent angibt.
2. Positiver Exponent = Rechts; Negativer Exponent = Links
3. Ergänze, soweit nötig, die fehlenden Nullen.

Beispiele

- 2.5678E2
 - Wir verschieben den Dezimalpunkt um 2 Stellen nach rechts; Das ergibt 256.78
- 1.87E5
 - Wir verschieben den Dezimalpunkt um 5 Stellen nach rechts; Das ergibt 187...
 - Wir ergänzen die fehlenden Nullen vor dem Dezimalpunkt; 187000
- 5.6213E-3
 - wir verschieben den Dezimalpunkt um **-3** Stellen nach rechts, also um 3 Stellen nach links; Das ergibt 0.0056213

m114-A1G



Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m114/learningunits/lu01/zahlensysteme>

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

