Zahlensysteme

Das Dezimalsystem

Basis: 10					
Ziffernwerte: 0,1,2,3,4,5,6,7,8,9					
Stellenwerte...	10^3	10^2	10^1	10^0	
	T	H	Z	E	
Beispiel:	7	5	4	3	$= 7543_{10}$

Die Zahl kann auch so dargestellt werden:

$$7 \cdot 10^3 + 5 \cdot 10^2 + 4 \cdot 10^1 + 3 \cdot 10^0$$

= 7000 + 500 + 40 + 3

= 7543

Warum sind für uns weitere Zahlensysteme wichtig?
Die internen Abläufe aller Computer sind in Speichereinheiten, sogenannten Bytes organisiert. Um Daten aus der realen Welt mit einem Computer bearbeiten zu können, müssen sie im Arbeitsspeicher vorliegen. Dazu müssen sie so dargestellt werden, dass sie in die Speichereinheiten passen.
Als Formate für die digitale Darstellung von Daten sind vor allem 4 Zahlentypen von Bedeutung:

- Binärzahl
- Dezimalzahl
- Oktalzahl
- Hexadezimalzahl

Das Oktalsystem

Basis: 8					
Ziffernwerte: 0,1,2,3,4,5,6,7					
Stellenwerte...	8^3	8^2	8^1	8^0	
	512	64	8	1	
Beispiel:	7	5	4	3	$= 7543_8$

Die Zahl kann auch so dargestellt werden:

$$7 \cdot 512 + 5 \cdot 64 + 4 \cdot 8 + 3 \cdot 1$$

$= 3584 + 320 + 32 + 3 = 3939_{10} = 7543_8$

Das Dualsystem

Basis: 2						
Ziffernwerte: 0,1						
Stellenwerte...	2^5	2^4	2^3	2^2	2^1	2^0
	32	16	8	4	2	1
Beispiel:	1	0	1	1	0	1

Die Zahl 101101_2 kann auch so dargestellt werden:

$$1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$$
$$= \quad 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}$$

Das Hexadezimalsystem

Basis: 16					
Ziffernwerte: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F					
Stellenwerte...	16^4	16^3	16^2	16^1	16^0
	65536	4096	256	16	1
Beispiel:		B	1	0	C

Die Zahl $B10C_{16}$ kann auch so dargestellt werden:

$$11 \cdot 4096 + 1 \cdot 256 + 0 \cdot 16 + 12 \cdot 1$$
$$= \quad 45056 + 256 + 0 + 12 = 45324_{10}$$

Umrechnungen

Die Umrechnung einer Zahl zu einer beliebigen Basis ins Dezimalsystem ist sehr einfach.

$$1011011_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$
$$= 64 + 0 + 16 + 8 + 0 + 2 + 1 = 91_{10}$$

$$2375_8 = 2 \cdot 8^3 + 3 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0 = 2 \cdot 512 + 3 \cdot 64 + 7 \cdot 8 + 5 \cdot 1$$
$$= 1024 + 192 + 56 + 5$$
$$= 1277_{10}$$

$$FFFF_{16} = 15 \cdot 4096 + 15 \cdot 256 + 15 \cdot 16 + 15 \cdot 1$$
$$= 61440 + 3840 + 240 + 15 = 65535_{10}$$

Die Umrechnung einer Dezimalzahl in eine Zahl zu einer anderen Basis ist etwas aufwendiger.

91_{10} soll in eine Dualzahl verwandelt werden.

<table>
<tr><td>91:2 = 45 Rest 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>45:2 = 22 Rest 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>|</td></tr>
<tr><td>22:2 = 11 Rest 0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>|</td><td>|</td></tr>
<tr><td>11:2 = 5 Rest 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>5:2 = 2 Rest 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>2:2 = 1 Rest 0</td><td>-</td><td>-</td><td>-</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>1:2 = 0 Rest 1</td><td>-</td><td>-|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td></td><td></td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
</table>

$91_{10} = 1011011_2$

1277_{10} soll in eine Oktalzahl umgerechnet werden.

1277:8 = 159 Rest 5	-	-	-	-	-
159:8 = 19 Rest 7	-	-	-	-	|
19:8 = 2 Rest 3	-	-	-	|	|
2:8 = 0 Rest 2	-	-	|	|	|
		|	|	|	|
		2	3	7	5

$1277_{10} = 2375_8$

65535_{10} soll in eine Hexadezimalzahl verwandelt werden.

65535:16 = 4095 Rest 15 = F	-	-	-	-	-
4095:16 = 255 Rest 15 = F	-	-	-	-	|
255:16 = 16 Rest 15 = F	-	-	-	|	|
15 : 16 = 0 Rest 15 = F	-	-	|	|	|
		|	|	|	|
		F	F	F	F

$65535_{10} = FFFF_{16}$

Bezug zum Computer

Ein Byte besteht auf den meisten Computern aus acht Bits, die jeweils gesetzt oder ungesetzt sein können. Diesen beiden Zuständen kann einfach jeweils eine Zahl zugeordnet werden – die 1 für gesetzt und die 0 für ungesetzt – und damit ein Byte als eine achtstellige Binärzahl dargestellt werden.

$\underbrace{[1][1][1][1][1][1][1][1]}_{8\ \text{bit}}$ 1 Byte $11111111_2 = 255$

Eine achtstellige Binärzahl kann $2^8 = 256$ verschiedene Kombinationan von Nullen und Einsen enthalten und damit beispielsweise von 0 bis 255 zählen.

Wird das höchste Bit eines Bytes als Vorzeichen interpretiert, so wird die Menge aller Bytes auf die Zahlen –128 bis 127 abgebildet.

1 Byte pro Zahl: $0 \le x \le 255$ oder $-128 \le x \le 127$
2 Byte pro Zahl: $0 \le x \le 65535$ oder $-32768 \le x \le 32767$

Größere Zahlen benötigen mehr Bytes.

Fließkommazahl: $-2{,}157 \cdot 10^{12}$ einfach genau (float)

- benötigt werden 4 Bytes = 32 Bits
- 1 Bit für Vorzeichen
- 8 Bit für den Exponenten
- 23 Bit für die Mantisse

Fließkommazahl: doppelt genau (double)

- benötigt werden 8 Bytes = 64 Bits
- 1 Bit für Vorzeichen
- 11 Bit für den Exponenten
- 52 Bit für die Mantisse

Oktalzahlen werden oft zur Darstellung von Bytes verwendet
(0 .... 7 $\hat{=}$ 8 Bits)

Im Hexadezimalsystem entspricht ein Byte aus 8 Bits genau einer zweistelligen Hexadezimalzahl.

11110101	
1111	0101
F	5

= 245