

Speichern von Daten im Computer

© R. Wittye – 18.01.2011

Inhalt

- Wie werden Informationen am PC gespeichert?
- Zahlensysteme
- Umrechnung zwischen Zahlensystemen
- Untersuchung von Dateien
- Zeichencodierung

Information und Daten

Welche Arten von Information werden in einem Rechner verarbeitet?

Aus Anwendersicht:

- Texte
- Zahlen
- Bilder
- Töne, etc...

Aus technischer Sicht gilt jedoch:

Alles ist eine Zahl.



Bits und Bytes

- **1 Bit** (= Binary Digit)
 - kleinste Dateneinheit
 - nimmt nur **Wert 1** oder **Wert 0** an
- **1 Byte** = geordnete Folge von 8 Bits
- Beispiel:

8 Bits
 $\underbrace{1101\ 1100}_{1\ \text{Byte}}\ \underbrace{1100\ 0001}_{1\ \text{Byte}}\ \underbrace{1010\ 1111}_{1\ \text{Byte}}$

Exkurs: Zahlensysteme

- Dienen zur Darstellung von Zahlen
- Heute im Alltag üblich: Dezimalsystem (Zehnersystem)
- Wie gerade gesehen: Bei Computern ist das Dualsystem (Binärsystem, Zweiersystem) wichtig
 - nur zwei Werte (0 und 1) werden unterschieden

Zahlensysteme: Stellenwert

- Zum einfachen Rechnen: Stellenwert entscheidend
 - D.h. jede Ziffer hat je nach ihrer Position in der Zahl eine andere Bedeutung
 - Wenn Ziffer 1 an 1. Stelle von rechts im Dezimalsystem → Wert: 1
 - Wenn Ziffer 1 an 2. Stelle von rechts im Dezimalsystem → Wert: 10
 - Jede Stelle im Dezimalsystem weiter links ist das 10-fache der Stelle rechts davon

Zahlensystem: Stellenwert

- Ein Stellenwertsystem hat eine Basis b .
- Jede Zifferposition hat einen Wert, der einer Potenz der Basis entspricht.
- Für die n -te Position (von rechts) hat man einen Wert von b^{n-1} .
- Berechnung des Zahlenwertes: Multiplikation der einzelnen Ziffern z_i mit den zugehörigen Stellenwerten b_i und Summation dieser Produkte:
Zahlenwert = $z_n \cdot b^n + \dots + z_1 \cdot b^1 + z_0 \cdot b^0$
- Kein Stellenwert: Römische Zahlen wie MXCVII

Dezimalsystem

- Auch Zehnersystem genannt
- Basis 10
- Verwendete Ziffern: 0,1,2,3,4,5,6,7,8,9
- Zahl $123 = 123_{10}$ bedeutet
 $1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 = 1 \cdot 100 + 2 \cdot 10 + 3 \cdot 1$

Dualsystem

- Auch Binärsystem, Zweiersystem genannt
- Basis: 2
- Verwendete Ziffern: 0,1

Umrechnung: Dualzahl $\rightarrow$ Dezimalzahl

Beispiele:

$$\mathbf{0101}_2 = \mathbf{0} \cdot 2^3 + \mathbf{1} \cdot 2^2 + \mathbf{0} \cdot 2^1 + \mathbf{1} \cdot 2^0 = 0 + 4 + 0 + 1 = 5_{10}$$

$$\begin{aligned} \mathbf{0011\ 1101}_2 &= \mathbf{0} \cdot 2^7 + \mathbf{0} \cdot 2^6 + \mathbf{1} \cdot 2^5 + \mathbf{1} \cdot 2^4 + \mathbf{1} \cdot 2^3 + \mathbf{1} \cdot 2^2 + \mathbf{0} \cdot 2^1 + \mathbf{1} \cdot 2^0 \\ &= 0 + 0 + 32 + 16 + 8 + 4 + 1 = \mathbf{61}_{10} \end{aligned}$$

Aufgabe 1:

- Wandle die Zahl $1001\ 1001_2$ in die entsprechende Dezimalzahl um.
- $\mathbf{1001\ 1001}_2 =$
 $\mathbf{1} \cdot 2^7 + \mathbf{0} \cdot 2^6 + \mathbf{0} \cdot 2^5 + \mathbf{1} \cdot 2^4 + \mathbf{2}^3 + \mathbf{0} \cdot 2^2 + \mathbf{0} \cdot 2^1 + \mathbf{1} \cdot 2^0 = \mathbf{153}_{10}$

Umrechnung: Dezimalzahl $\rightarrow$ Dualzahl

Dazu verwendet man das folgende Verfahren:

Beispiel: $26_{10} = 0001\ 1010_2$

26:2=13	Rest 0
13:2= 6	Rest 1
6:2= 3	Rest 0
3:2= 1	Rest 1
1:2= 0	Rest 1

Ergebnis: 0 0 0 1 1 0 1 0

Mit Nullen aufgefüllt,
da es sich um ein
8-Bit-Wert handelt.

Aufgabe 2:

- Rechne die Dezimalzahl 170_{10} ins Zweiersystem um.
- $170_{10} = 1010\ 1010_2$

Probleme bei Dualzahlen

- Sehr lange Zahlen
- Recht schnell unübersichtlich
- Kürzere Darstellung notwendig, die schnell aus Binärdarstellung ableitbar
- Byte wird halbiert $\rightarrow$ 4 Bits
 - $\rightarrow$ können 16 unterschiedliche Zustände speichern
 - $\rightarrow$ 16er-System (Hexadezimalsystem)

Hexadezimalsystem

- Hexadezimalzahlen werden kurz Hex-Zahlen genannt
- Basis: 16
- Verwendete „Ziffern“: 0,1,2,3,4,5,6,7,8,9 und A,B,C,D,E,F
- Dabei A = 10, B = 11, ... , F = 15

Umrechnung: Hexzahl $\rightarrow$ Dezimalzahl

Beispiel:

$$\begin{aligned} \text{■ } C9_{16} &= „C“ \cdot 16^1 + 9 \cdot 16^0 = 12 \cdot 16^1 + 9 \cdot 16^0 \\ &= 12 \cdot 16 + 9 \cdot 1 = 192 + 9 = 201 \end{aligned}$$

Umrechnung: Dezimalzahl $\rightarrow$ Hexzahl

Dazu verwendet man das Verfahren wie bei Dualzahlen:

Beispiel: $77_{10} = 4D_{16}$

$$\begin{aligned} 77:16 &= 4 \text{ Rest } 13 \text{ („D“)} \\ 4:16 &= 0 \text{ Rest } 4 \end{aligned}$$

Ergebnis: 4 D

Arbeitsphase

- Aufgabe 1 und 2 auf dem Blatt fertig bearbeiten

Lösungen der Aufgaben 1

- Aufgabe 1
 - $1001\ 1001_2 = 153_{10}$
 - $55_{16} = 85_{10}$
 - $A9_{16} = 169_{10}$
- Aufgabe 2
 - $170_{10} = 1010\ 1010_2 = AA_{16}$

Arbeitsphase

Aufgabe 3 – 7 auf dem Blatt schriftlich!

Zeichencodierung

- In Computer werden auch Buchstaben als Zahlen gespeichert.
- Ein Buchstabe entspricht immer der gleichen Zahl (Zeichencodierung)
- Folgende Zeichencodierungen sind üblich:
 - ASCII
 - ANSI (ISO 8859)
 - Unicode

ASCII-Code

- ASCII steht für American Standard Code for Information Interchange
- Wurde erstmals 1967 veröffentlicht
- 7-Bit Zeichencode
- enthält also nur 127 Zeichen!
- Keine Umlaute oder Akzente.
- In DOS länderspezifische Ergänzungen durch sogenannte Codepages

ASCII-Code

ASCII-Codetabelle, Nummerierung in Hex

Code	...0	...1	...2	...3	...4	...5	...6	...7	...8	...9	...A	...B	...C	...D	...E	...F
0...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1...	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2...	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3...	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4...	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5...	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6...	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7...	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Quelle: http://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange

ASCII-Code

- Von 0_{10} bis 31_{10} : „nichtdruckbare“ Steuerzeichen
- Von 32 – 64: Sonderzeichen und Ziffern
- Von 65 – 90: Großbuchstaben
- Von 91 – 96: Sonderzeichen
- Von 97 – 122 :Kleinbuchstaben
- Von 123 – 127: Sonderzeichen
- Keine Umlaute usw. vorhanden

ANSI (ISO-8859)

- Ergänzung des ASCII-Codes um 8. Bit
- Wurde von Microsoft Windows (bis Windows ME/2000) verwendet
- Ab Windows 2000 Unicode möglich
- Umlaute und viele andere länderspezifische Zeichen enthalten
- Ausnahme: Asiatische Zeichen

Vergleich ASCII - ANSI

[illegible]

Quelle: <http://www.torsten-horn.de/techdocs/ascii.htm>

Unicode

- Zeichencodierung, die aus mehreren Bytes besteht
- Beginnt wie die ASCII-Zeichencodierung
- Restliche Codierung komplizierter: siehe Links bzw. Quellenangaben
- Eine Zeichencodierung weltweit möglich
- C# nutzt Unicode
- Am meisten verbreitet sind die Unicode-Varianten **UTF-8** oder **UTF-16** (Windows)

Quellen

- Einige Folien basieren auf einer Präsentation von M. Diehm (Informatik-Kurs 13 am LMG vom 07.12.2005)
- <http://de.wikipedia.org/wiki/Zeichenkodierung>
- <http://de.wikipedia.org/wiki/Ascii>
- <http://de.wikipedia.org/wiki/ASCII-Tabelle>
- <http://www.torsten-horn.de/techdocs/ascii.htm>
- <http://de.wikipedia.org/wiki/Ansi>
- <http://de.wikipedia.org/wiki/ISO-8859>
- <http://de.wikipedia.org/wiki/Unicode>
- <http://de.wikipedia.org/wiki/UTF-8>
- <http://de.wikipedia.org/wiki/UTF-16>
- <http://www.asphelper.de/referenz/ASCII.asp>
- <http://www.itwissen.info/definition/lexikon/ANSI-Code-ANSI-code.html>
- <http://de.wikipedia.org/wiki/Zahlensystem>
- <http://de.wikipedia.org/wiki/Dualsystem>
- <http://de.wikipedia.org/wiki/Hexadezimal>

Alle Links: Stand 04.10.2008