

## Script generated by TTT

Title: Einf\_HF (12.05.2014)

Date: Mon May 12 14:14:44 CEST 2014

Duration: 88:09 min

Pages: 27

Fließband Bearbeitung (Pipelining) ↑

Bearbeitung jeden Befehls in mehreren Phasen. Überlappende Verarbeitung. Quasi-parallele Ausführung mehrerer Maschinenbefehle.

Zeit

Befehl holen → Befehl decodieren → Befehl ausführen

Befehl 1  
Befehl 2  
Befehl 3

Phase 1      Phase 2      Phase 3

**Pipelining Animation**

Generated by Targeteam



Interdarstellung von Information



Beispiele von Codierungen ↑

### [Codierung](#)

[Codierung ganzer Zahlen](#)

[Codierung von Text](#)

[Codierung von Bildern und Tönen](#)

### Komprimierung

Datenkompression: reduzierte Speicher- und Übertragungskosten.

#### Verlustfreie Kompression

Ausnutzung von Mustern und Redundanzen in den Daten; Ausnutzung der Häufigkeit von Symbolen durch Änderung der Codierung.

#### Verlustbehaftete Kompression

Ausnutzung von Medien- und Wahrnehmungseigenschaften, z.B. bei MP3.

Generated by Targeteam

### bekannte Codierungen aus der Praxis

Barcode

QR-Code



Barcode (Strichcode): befindet sich auf fast jedem Artikel.

wird nicht intern vom Rechner verwendet, kann jedoch vom Rechner dekodiert werden.

QR-Code ("Quick Response"): entwickelt von Denso Wave (1994)

quadratische Matrix aus schwarzen und weißen Punkten, die die kodierten Daten binär darstellen. Fehlerkorrektur bis zu 30%; Darstellung von alphanumerischen Zeichen oder Kanji/Kana Zeichen.

Viele Smartphones verfügen über eine eingebaute Kamera und eine Software, die das Interpretieren von QR-Codes ermöglicht.

Generated by Targeteam



Zuordnung (oder Abbildung) der Werte eines Zeichenvorrats auf Werte eines anderen Zeichenvorrats.

### Beispiele von Codierungen

Zeichen: Ausprägung (Form, Wert) eines Signals; auch: Symbole.

Zeichenvorrat: Menge der Zeichen (d.h. Formen, Werte), die ein bestimmtes Signal annehmen kann.

Codierung erfolgt für bestimmten Zweck:

- Speicherung
- Übertragung
- Komprimierung, z.B. von Bildern oder Video
- Verschlüsselung
- Veranschaulichung

Codierung z.B. notwendig um für Menschen verständliche Information auf für Rechner verständliche oder speicherbare Darstellung abzubilden. (Symbole auf Bitfolgen.)

Abbildung berechenbar, eindeutig und (in der Regel) umkehrbar.

Generated by Targeteam



Umwandlung einer Dezimalzahl  $w$  in eine Dualzahl  $z$

dividiere  $w$  durch 2: Ergebnis  $w_1$  und Rest  $r_0$

dividiere  $w_1$  durch 2: Ergebnis  $w_2$  und Rest  $r_1$

fahre fort, bis das Ergebnis der Division 0 und Rest  $r_k$  ist.

Die Dualzahl ist  $z = r_k r_{k-1} \dots r_1 r_0$

### Beispiel

Dezimalzahl  $w = 23$

$23 : 2 = 11$  mit Rest 1

$11 : 2 = 5$  mit Rest 1

$5 : 2 = 2$  mit Rest 1

$2 : 2 = 1$  mit Rest 0

$1 : 2 = 0$  mit Rest 1

Die Dualzahl lautet:  $z = 00010111$  (in 8-Bit Darstellung)



Generated by Targeteam



Codierung im Binärsystem. Zwei Ziffern 0,1 ("Bits") geben Anzahl von Zweierpotenzen an. Vgl. Dezimalsystem: Zehn Ziffern geben Anzahl von Zehnerpotenzen an.

### Beispiel

Dezimalsystem:  $148 = 1 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$

Binärsystem:  $1010 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$  (= 10 im Dezimalsystem)

### Formel für Wert einer Binärsystem-Zahl

$$W = \sum_{i=0}^{n-1} (b_i \times 2^{n-1-i})$$

mit den Binärziffern  $b_i \in \{0,1\}$  und  $n$  ist die Anzahl der verwendeten Bits (d.h. eine  $n$ -stellige Zahl). Beachte, es wird die Folge  $b_0 b_1 \dots b_{n-1}$  betrachtet.

### Beispiel

eine ganze Zahl sei als 8 bit lange Zahl zur Basis 2 dargestellt

$$W(00001101_2) = 0 \times 2^7 + \dots + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{10}$$

### Verfahren zur Umwandlung

#### Feste Zifferanzahl

Typisch: Feste Bitzahl, meist ebenfalls Zweierpotenz. Z.B. 4 Bit, 16 Bit, 32 Bit oder 64 Bit. Aktuell entweder 32 oder 64 Bit verwendet. Mit  $n$  Bit codierbar: Werte 0 bis  $2^n - 1$ .

#### Negative Zahlen



Codierung im Binärsystem. Zwei Ziffern 0,1 ("Bits") geben Anzahl von Zweierpotenzen an. Vgl. Dezimalsystem: Zehn Ziffern geben Anzahl von Zehnerpotenzen an.

### Beispiel

Dezimalsystem:  $148 = 1 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$

Binärsystem:  $1010 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$  (= 10 im Dezimalsystem)

### Formel für Wert einer Binärsystem-Zahl

$$W = \sum_{i=0}^{n-1} (b_i \times 2^{n-1-i})$$

mit den Binärziffern  $b_i \in \{0,1\}$  und  $n$  ist die Anzahl der verwendeten Bits (d.h. eine  $n$ -stellige Zahl). Beachte, es wird die Folge  $b_0 b_1 \dots b_{n-1}$  betrachtet.

### Beispiel

eine ganze Zahl sei als 8 bit lange Zahl zur Basis 2 dargestellt

$$W(00001101_2) = 0 \times 2^7 + \dots + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{10}$$

### Verfahren zur Umwandlung

#### Feste Zifferanzahl

Typisch: Feste Bitzahl, meist ebenfalls Zweierpotenz. Z.B. 4 Bit, 16 Bit, 32 Bit oder 64 Bit. Aktuell entweder 32 oder 64 Bit verwendet. Mit  $n$  Bit codierbar: Werte 0 bis  $2^n - 1$ .

#### Negative Zahlen



Zehn Ziffern geben Anzahl von Zehnerpotenzen an.

**Beispiel**

Dezimalsystem:  $148 = 1 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$

Binärsystem:  $1010 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$  (= 10 im Dezimalsystem)

**Formel für Wert einer Binärsystem-Zahl**

$$W = \sum_{i=0}^{n-1} (b_i \times 2^{n-1-i})$$

mit den Binärziffern  $b_i \in \{0,1\}$  und n ist die Anzahl der verwendeten Bits (d.h. eine n-stellige Zahl). Beachte, es wird die Folge  $b_0 b_1 \dots b_{n-1}$  betrachtet.

**Beispiel**

eine ganze Zahl sei als 8 bit lange Zahl zur Basis 2 dargestellt

$$W (00001101_2) = 0 \times 2^7 + \dots + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{10}$$

Verfahren zur Umwandlung

**Feste Zifferanzahl**

Typisch: Feste Bitzahl, meist ebenfalls Zweierpotenz. Z.B. 4 Bit, 16 Bit, 32 Bit oder 64 Bit. Aktuell entweder 32 oder 64 Bit verwendet. Mit n Bit codierbar: Werte 0 bis  $2^n - 1$ .

Negative Zahlen

Generated by Targeteam



Positive ganze Zahlen Darstellung im Binärsystem. Für negative ganze Zahlen mehrere Möglichkeiten.

**Vorzeichen-Darstellung**

Erstes Bit: Vorzeichen (0 = +, 1 = -), restliche Bits Absolutwert der Zahl im Binärsystem. Bei n Bits sind Zahlen von  $-2^{n-1} - 1$  bis  $2^{n-1} - 1$  codierbar. Zwei Nullen: +0 (000...00), -0 (100...00).

Beispiel

Zweierkomplement-Darstellung

Generated by Targeteam



Positive ganze Zahlen Darstellung im Binärsystem. Für negative ganze Zahlen mehrere Möglichkeiten.

**Vorzeichen-Darstellung**

Erstes Bit: Vorzeichen (0 = +, 1 = -), restliche Bits Absolutwert der Zahl im Binärsystem. Bei n Bits sind Zahlen von  $-2^{n-1} - 1$  bis  $2^{n-1} - 1$  codierbar. Zwei Nullen: +0 (000...00), -0 (100...00).

Beispiel

Zweierkomplement-Darstellung

Generated by Targeteam

für 4 bit Darstellung - Windows Internet Explorer

C:\www\leinfss14\whiteboard\leinf\_course4.4.2.2.2.1.html

Beispiel für 4 bit Darstellung

**Beispiel für 4 bit Darstellung**

Zweierkomplement-Darstellung mit 4 bit für eine ganze Zahl

$W_+ = \sum_{i=1}^3 (b_i \times 2^{3-i})$  (positive Zahl)  
 $W_- = -2^3 + \sum_{i=1}^3 (b_i \times 2^{3-i})$  (negative Zahl)  
 mit  $b_i \in \{0,1\}$

Windows taskbar: Start, 15:01

für 4 bit Darstellung - Windows Internet Explorer

C:\www\enf-ss14\whiteboard\enf\_course4.4.2.2.1.html

C:\Dokumente und Einstellungen\Schlifter\Startmenu

Beispiel für 4 bit Darstellung

Zweierkomplement-Darstellung

1111 0000

1110 -1 0

1101 -2 0

1100 -3 0

1011 -4 0

1010 -5 0

1001 -6 0

1000 -7 0

1000 -8 0

$W_+ = \sum_{i=1}^3 (b_i \times 2^{3-i})$  (positive Zahl)

$W_- = -2^3 + \sum_{i=1}^3 (b_i \times 2^{3-i})$  (negative Zahl)

mit  $b_i \in \{0,1\}$

für 4 bit Darstellung - Windows Internet Explorer

C:\www\enf-ss14\whiteboard\enf\_course4.4.2.2.1.html

C:\laufzeichnung

Beispiel für 4 bit Darstellung

Zweierkomplement-Darstellung

1111 0000

1110 -1 0

1101 -2 0

1100 -3 0

1011 -4 0

1010 -5 0

1001 -6 0

1000 -7 0

1000 -8 0

$W_+ = \sum_{i=1}^3 (b_i \times 2^{3-i})$  (positive Zahl)

$W_- = -2^3 + \sum_{i=1}^3 (b_i \times 2^{3-i})$  (negative Zahl)

mit  $b_i \in \{0,1\}$

Rechnen mit Zweierkomplement-Zahlen

Rechnen mit Zweierkomplement-Zahlen

Negativbildung und Grundrechenarten sind einfach durchführbar.

**Negativbildung einer Zahl**

Komplementbildung (Bits invertieren) und 1 addieren.

**Beispiel**

Zweierkomplement-Codierung mit 8 Bit für -14:

14 =	00001110
Komplement:	11110001
1 addiert:	11110010

**Addition von zwei Zahlen**

Stellenweise mit Übertrag, analog zum Dezimalsystem.

**Differenzbildung von zwei Zahlen**

Realisierbar durch Addition mit negativer Zahl.

**Beispiel**

Berechnung 17 - 14:

dezimal	dual
17	00010001
+(-14)	11110010
= 3	00000111

Komplementbildung (Bits invertieren) und 1 addieren.

**Beispiel**

Zweierkomplement-Codierung mit 8 Bit für -14:

14 =	00001110
Komplement:	11110001
1 addiert:	11110010

*Handwritten calculation:*

11110001  
 00000001  
 -----  
 11110010

**Addition von zwei Zahlen**

Stellenweise mit Übertrag, analog zum Dezimalsystem.

**Differenzbildung von zwei Zahlen**

Realisierbar durch Addition mit negativer Zahl.

**Beispiel**

Berechnung 17 - 14:

dezimal	dual
17	00010001
+(-14)	11110010
= 3	00000111



Eine negative Zahl mehr als positive Zahlen. Einfache Umsetzung von Addition und Subtraktion.

Beispiel für 4 bit Darstellung

Formel für Wert einer Zweierkomplement-Zahl

$$W = -b_0 \times 2^{n-1} + \sum_{i=1}^{n-1} (b_i \times 2^{n-1-i})$$

mit  $b_i \in \{0, 1\}$  . n ist hier die Anzahl der Bitstellen.

Beispiel

Wert der Zahl W: -1

Binärdarstellung mit 4 Bit: 1111

$$W = -2^3 + 2^2 + 2^1 + 2^0 = -8 + 7 = -1$$

Rechnen mit Zweierkomplement-Zahlen

Generated by Targemam



Darstellung von Buchstaben und Ziffern in einer 8-Bit Folge, d.h. wie Zahl zwischen 0 und 255. *2~*

ISO = International Standards Organisation

ASCII = American Standard Code for Information Interchange

Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)

Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)

Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)

Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzende), LF (Linefeed = Neuzeile)

Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001 <i>a &lt; b</i>
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111
CR	13	00001101

Bei Netzübertragung gelegentlich noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö sind nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").

Unicode



Alphanumerische Daten - ISO-ASCII 8-bit-Code

Darstellung von Buchstaben und Ziffern in einer 8-Bit Folge, d.h. wie Zahl zwischen 0 und 255.

ISO = International Standards Organisation

ASCII = American Standard Code for Information Interchange

Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)

Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)

Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)

Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzende), LF (Linefeed = Neuzeile)

Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111
CR	13	00001101

Bei Netzübertragung gelegentlich noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö sind nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").



ASCII = American Standard Code for Information Interchange

Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)

Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)

Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)

Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzende), LF (Linefeed = Neuzeile)

Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001 <i>a &lt; b</i>
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111
CR	13	00001101

*2^7 = 128*

Bei Netzübertragung gelegentlich noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö sind nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").

Unicode

Unicode codiert Zeichen mit zwei Bytes. 65536 Zeichen.

Generated by Targemam



nicht Zahl!

ASCII = American Standard Code for Information Interchange  
 Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)  
 Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)  
 Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)  
 Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzen), LF (Linefeed = Neuzeile)  
 Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001 $a < b$
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111 $2^7 \hat{=} 128$
CR	13	00001101

Bei Netzübertragung gelegentlich noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö sind nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").

Unicode

Unicode codiert Zeichen mit zwei Bytes. 65536 Zeichen.

Generated by Targeseam



Problem: Information gleichmäßig über Fläche verteilt.

- Auflösung in Rasterpunkte. Bildschirm: 60 bis 360 Bildelemente (Pixel) pro Zoll (2,54cm)
- Darstellung Eigenschaft eines Pixels (Grauwert, Farbe, Helligkeit): meist ein oder zwei Byte
- Darstellung Farbinformation: RGB (rot-grün-blau) oder andere Codierungen
- SVGA: 1024 \* 768 \* (8 bit pro Pixel / 8 bit pro Byte) = 786432 Byte
- Graphics Interchange Format (GIF): häufig vorkommende Folgen von Bytes werden in Tabelle eingetragen; im Bild Verweis auf Tabelleneintrag.

The rain in Spain falls mainly on the plain, while the rain in the Amazon just falls => 85 Zeichen

Abkürzungen: W = the, X = ain, Y = on, Z = falls

W rX in SpX Z mXly Y W plX, while W rX in W AmazY just Z => 57 Zeichen

Joint Photographic Expert Group (JPG): Farben des Bildes werden analysiert; weglassen von Information, die für menschliches Auge nicht wichtig erscheint (**Achtung: Verlust von Information**).

Generated by Targeseam



Graphiken

Unterscheidung zwischen Rastergrafik (Bilder) und Vektorgrafik

Eigenschaft	Rastergrafik	Vektorgrafik
Dokument besteht aus	Folge von Pixeln	Menge von geometrischen Objekten
Eignung	Fotos	Zeichnungen
Platzbedarf DIN A4, 16 Mio Farben, 600dpi	ca 95 MB	je nach Umfang ca 10 KB - 1 MB
Formate	BMP, GIF, JPG, PNG	WMF, VSD, CDR

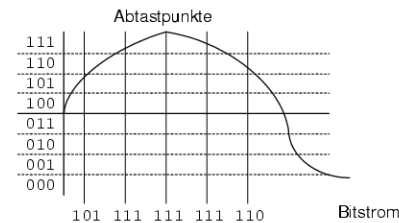
Rastergrafik - Bilder

Töne

Generated by Targeseam



Information gleichmäßig über Zeitdauer verteilt.



- Diskretisierung, Digitalisierung. 100, 1000 und mehr Werte pro Sekunde.
- Darstellung der Eigenschaften des Tonelements durch ein oder zwei Byte
- Sprache wird beim Telefon 8000 mal pro Sekunde (8kHz) abgetastet.

Generated by Targeseam



## Codierung

### [Codierung ganzer Zahlen](#)

### [Codierung von Text](#)

### [Codierung von Bildern und Tönen](#)

## Komprimierung

Datenkompression: reduzierte Speicher- und Übertragungskosten.

### Verlustfreie Kompression

Ausnutzung von Mustern und Redundanzen in den Daten; Ausnutzung der Häufigkeit von Symbolen durch Änderung der Codierung.

### Verlustbehaftete Kompression

Ausnutzung von Medien- und Wahrnehmungseigenschaften, z.B. bei MP3.

Generated by Targeteam



Ohne Programme ist Hardware nicht arbeitsfähig. Zwei Klassen: Anwendungsprogramme, Systemprogramme (insbes.: Betriebssystem; elementare Dienste).

### • Fragestellungen des Abschnitts:

- Was sind die Aufgaben eines Betriebssystems?
- Welche Dienste bietet ein Betriebssystem zur Arbeit mit Massenspeichern (Festplatte)?
- Was sind Prozesse (im Gegensatz zu Programmen)?
- Wie wird der Arbeitsspeicher verwaltet?

### [Einführung](#)

### [Dateiverwaltung](#)

### [Prozessorverwaltung / Prozessorzuteilung](#)

Generated by Targeteam



Betriebssysteme: verwalten Ressourcen eines Rechners (z.B. Arbeitsspeicher), schirmen Anwendungsprogramme von Eigenschaften der Hardware ab (z.B. Ausführung mehrerer Programme mit einem Prozessor).

### [Aufgaben des Betriebssystems](#)

### [Hardware-Software Struktur](#)

### [Betriebssysteme versus Benutzeroberflächen](#)

### [Wesentliche Komponenten eines Betriebssystems](#)

## Aktuelle Betriebssysteme

- DOS / Windows für Arbeitsplatzrechner/Clients und Arbeitsgruppen-Server, MacOS
- Unix (Linux, Solaris, HP-UX, Irix, ...) für Server (insbes. Web-Server) und Arbeitsplatzrechner
- MVS (IBM) für Großrechner/Server
- BS2000 (Siemens) für Großrechner/Server
- PalmOS, Windows Mobile für PDAs (Handhelds)

Generated by Targeteam



## Sichten

### 1. Top-Down-Sicht

Abschirmung von der Hardware. Betriebssystem als einfacher zu programmierende "virtuelle Maschine".

### 2. Bottom-Up-Sicht

Betriebssystem als "Ressourcenverwalter": steuert alle internen Abläufe, speziell den Mehrbenutzerbetrieb, koordiniert die Ressourcen des Rechners (Arbeitsspeicher, Platte, Terminal, Drucker, ...).

## Aufgaben

### Beispielabläufe

Benutzer schreibt Text (Textverarbeitungsprogramm), hört Musik (über den Computer), druckt Tabelle aus Tabellenkalkulation aus. Steuerung gleichzeitiger Nutzung mehrerer Geräte ist Aufgabe des Betriebssystems.

Anwender druckt Text. Betriebssystem muß sicherstellen, dass erst Tabelle zu Ende gedruckt wird.

Generated by Targeteam