1.Vorlesung Grundlagen der Informatik

Christian Baun

Hochschule Darmstadt
Fachbereich Informatik
christian.baun@h-da.de

13.10.2011

Heute

- Vorstellung
- Organisatorisches zur Vorlesung
- Literatur
- Grundlagen der Informatik
  - Definition der Informatik
  - Teildisziplinen der Informatik
  - Informationen und Daten
  - Repräsentation von Zahlen
  - Datei- und Speichergrößen
  - Informationsdarstellung

Christian Baun

- 2005 Diplom in Informatik an der FH-Mannheim
- 2006 Master of Science an der HS-Mannheim
- 2006 – 2008 Wissenschaftler im Institut für Wissenschaftliches Rechnen des Forschungszentrum Karlsruhe
  - Teil des D-Grid Integrationsprojekts (`http://www.d-grid.de`)
- 2008 – 2011 Wissenschaftler im Steinbuch Centre for Computing des Karlsruher Institut für Technologie
  - Schwerpunkte: Cloud-Computing, Virtualisierung, Verteilte Dateisysteme
  - Titel: Untersuchung und Entwicklung von Cloud Computing-Diensten als Grundlage zur Schaffung eines Marktplatzes
- 2011 Promotion an der Universität Hamburg
- Ab Oktober 2011 Vertretungsprofessur an der HS-Darmstadt

Organisatorisches zur Vorlesung und Übung

- **Homepage:** `https://www.fbi.h-da.de/organisation/personen/baun-christian.html`
- **E-Mail:** `christian.baun@h-da.de`
- **Skriptum:** Folienskript auf der Homepage
- **Vorlesung:**
  - Zug C + D: Donnerstags. 14:15-15:45 Uhr. Raum 17/119
  - Zug A + B: Prof. Dr. Hans-Peter Wiedling
- **Übungen:** Praktikum als Voraussetzung zur Klausurteilnahme!
  - Zug C: Donnerstags. 17:45-19:15 Uhr. Raum 16/012a
  - Zug D: Donnerstags. 16:00-17:30 Uhr. Raum 16/012a
  - Übungsblatt 1: Morgen online!

Organisatorisches

Bildquelle: Google

- Zitat von Mr. Miyagi:
  „*Nicht nur der Schüler lernt von seinem Meister; auch der Meister lernt von seinem Schüler.*“

- **Rege Teilnahme an Vorlesung und Übung ist sehr erwünscht!**

Inhalt der Vorlesung

- Organisatorisches
- Definition der Informatik und historische Entwicklung
- Informationsdarstellung
- Boolesche Algebra, Hardware-Komponenten
- Weg vom Programm zum Maschinenprogramm
- Betriebssysteme
- Formale Sprachen
- Datentypen und Kontrollstrukturen
- Grundlagen der Computervernetzung
- Client-Server
- Informationsdarstellung mit HTML und XML

Literatur

- **Grundlagen der Informatik**, *Helmut Herold, Bruno Lurz, Jürgen Wohlrab*, Pearson (2007)
- **Einführung in die Informatik**, *Heinz Peter Gumm, Manfred Sommer*, Oldenburg (2011)
- **Moderne Betriebssysteme**, *Andrew S. Tanenbaum*, Pearson Studium (2009)
- **Betriebssysteme**, *William Stallings*, Pearson Studium (2003)
- **Computernetzwerke**, *Andrew S. Tanenbaum*, Pearson (2000)
- **Verteilte Systeme: Prinzipien und Paradigmen**, *Andrew S. Tanenbaum, Maarten van Steen*, Pearson (2008)
- **IT Handbuch für Fachinformatiker**, *Sascha Kersken*, Galileo Computing (2009)
  `http://openbook.galileocomputing.de/it_handbuch/`

Informatik

- Informatik ist die **Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, besonders der automatischen Verarbeitung mit Hilfe von Digitalrechnern (Computern)**
  - Quelle: Duden (leicht abgewandelt)

- „In der Informatik geht es genau so wenig um Computer, wie in der Astronomie um Teleskope.“ (Edsger W. Dijkstra)
- Die englische Bezeichnung der akademischen Disziplin Informatik ist *Computer Science* und **nicht** Information Technology (IT) ($\Longrightarrow$ Informationstechnik)
- Bis in die 1960er Jahre war Informatik ein Spezialgebiet anderer wissenschaftlicher Disziplinen wie der Mathematik oder Elektrotechnik
- In vielen Teilbereichen ist Informatik heute eine Ingenieurwissenschaft

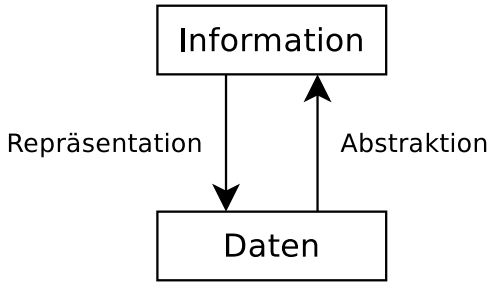
Teildisziplinen der Informatik

- Die Informatik unterteilt sich in 3 Teilgebiete
  - **Theoretische Informatik**
    - Logik, Automatentheorie, formale Sprachen, Berechenbarkeitstheorie, Komplexitätstheorie,...
  - **Praktische Informatik**
    - Algorithmen, Datenstrukturen, Programmiersprachen, Compiler, Interpreter, Softwareentwicklung, Betriebssysteme, Datenbanken,...
  - **Technische Informatik**
    - Hardwareentwicklung, Eingebettete Systeme, Systemnahe Softwareentwicklung, Elektrotechnik (insbes. Digitaltechnik), Echtzeitsysteme, Rechnernetze, Sensorik, Signal- und Bildverarbeitung,...
- Diese 3 Teilgebiete nennt man auch **Kerninformatik**
- Auf der Kerninformatik baut die **Angewandte Informatik** mit ihren verschiedenen Ausprägungen auf
  - Wirtschaftsinformatik, Medizinische Informatik, Bioinformatik, Geoinformatik, Umweltinformatik, Rechtsinformatik, Künstliche Intelligenz, Computerlinguistik,...

Informationen und Daten

- In Computersystemen (Rechnern) werden **Informationen** in Form von Zahlen (Nullen und Einsen) verarbeitet
- Es existieren verschiedene Wege, wie die Informationen mit Nullen und Einsen **repräsentiert** werden können

- Die so repräsentierten Informationen sind die **Daten**
- Die Repräsentation muss immer so gewählt sein, dass man aus den Daten wieder die Information zurückgewinnen kann



- Die Interpretation von Daten als Information heißt **Abstraktion**

Bit

- Kleinstmögliche Einheit der Information
- Jede Information ist an einen Informationsträger gebunden
- Ein Informationsträger, der sich in genau einem von 2 Zuständen befinden kann, kann die Datenmenge 1 Bit darstellen
  - In der Informatik nennt man den Wert eines oder mehrerer Bits **Zustand**
  - 1 Bit stellt 2 Zustände dar
- 1 Bit ist die Informationsmenge in einer Antwort auf eine Frage, die zwei mögliche Antworten zulässt
  - ja oder nein
  - wahr oder falsch
  - hell oder dunkel
  - ...
- Die Antwort wird mit Hilfe der beiden Zeichen 0 und 1 kodiert
- Diese Kodierung (binärer Code) stellt Informationen technisch dar
  - Elektrische Ladungen: 0 = ungeladen, 1 = geladen
  - Elektrische Spannungen: 0 = 0 Volt, 1 = 5 Volt
  - Magnetisierung: 0 = nicht magnetisiert, 1 = magnetisiert

Bitfolgen

- Mit n Bit kann man 2^n verschiedene Zustände darstellen
- Mit 2 Bit kann man $2^2 = 4$ verschiedene Zustände repräsentieren, nämlich 00, 01, 10 und 11
- Mit 3 Bit kann man $2^3 = 8$ verschiedene Zustände repräsentieren, nämlich 000, 001, 010, 011, 100, 101, 110 und 111
- Mit 4 Bit kann man schon $2^4 = 16$ verschiedene Zustände repräsentieren
- Jedes zusätzliche Bit verdoppelt die Anzahl der möglichen darstellbaren Zustände
  - Darum gilt: Es gibt genau 2^n verschiedene Bitfolgen der Länge n

Anzahl der Bit	Anzahl der Zustände	Anzahl der Bit	Anzahl der Zustände
1	$2^1 = 2$	9	$2^9 = 512$
2	$2^2 = 4$	10	$2^{10} = 1.024$
3	$2^3 = 8$	11	$2^{11} = 2.048$
4	$2^4 = 16$	12	$2^{12} = 4.096$
5	$2^5 = 32$	13	$2^{13} = 8.192$
6	$2^6 = 64$	14	$2^{14} = 16.384$
7	$2^7 = 128$	15	$2^{15} = 32.768$
8	$2^8 = 256$	16	$2^{16} = 65.536$

Repräsentation von Zahlen

- Zahlen kann man auf unterschiedliche Arten darstellen
  - Aufgabe: Zahlen aus der *realen Welt* im Computer abbilden
- Wichtig ist die Unterscheidung zwischen **Wert** und **Darstellung**
- In der Mathematik unterscheidet man Zahlen als Elemente verschiedener Wertemengen
  - Natürliche Zahlen, ganze Zahlen, reelle Zahlen, komplexe Zahlen,...
  - Beispiel: 2 ist ein Element der natürlichen Zahlen
- **Konkrete Zahlen** werden durch Zeichen repräsentiert
  - Im täglichen Leben handelt es sich üblicherweise um Dezimalzahlen
- Neben der Darstellung interessiert häufig auf der Wert einer Zahl
  - Den Wert nennt man **abstrakte Zahl**
  - Der Wert ist unabhängig von der Darstellung (z.B. $0,5 = 1/2$)

Repräsentation von Zahlen

- Operationen eines Rechners werden auf Zeichen (Bitmuster) und nicht auf Werten ausgeführt
  - Darum ist für die Informatik besonders die Darstellung der Zahlen interessant
- Um Bitmuster korrekt zu interpretieren, muss eine **Abbildung zwischen der Darstellung und dem Wert** einer (binär) kodierten Zahl vorliegen

Als nächstes werden die für die Informatik wichtigsten Zahlendarstellungen vorgestellt

Positive ganze Zahlen

- Wir rechnen im **Dezimalsystem**
  - Wird auch als Zehnersystem System bezeichnet
  - Darstellung von natürlichen Zahlen mit den Symbolen $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, +\}$
    - Beispiel: 0, $+123$, -987
  - Die Zahlen beginnen mit einem Vorzeichensymbol aus der Menge $\{+, -\}$
  - Bei positiven Zahlen und der Null kann das Vorzeichensymbol weggelassen werden
  - Danach kommen Ziffernsymbole aus der Menge $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  - Das Dezimalsystem verwendet als Basis die Zahl 10
  - Jede Ziffer D an der Stelle i hat den Wert $D * 10^i$
    - Beispiel: $2011 = 2 * 10^3 + 0 * 10^2 + 1 * 10^1 + 1 * 10^0$
- Rechner unterscheiden zwischen 2 elektrischen Werten
  - Darum ist als Basis die Zahl 2 optimal geeignet

Dualzahlen

- Zahlen werden nur mit den Ziffern des Wertes Null und Eins dargestellt
- Die Zahldarstellungen im Dualsystem werden auch **Dualzahlen** oder **Binärzahlen** genannt
- Positive natürliche Zahlen inklusive der Null können mit den Symbolen $B = \{1, 0\}$ repräsentiert werden
  - Somit werden dann alle positiven natürliche Zahlen inklusive der Null durch Folgen von Symbolen aus der Menge B abgebildet
  - Beispiel: 1011011011

LSB, MSB

- x_0, das niederwertigste Bit, heißt **LSB** (Least Significant Bit)
- x_{n-1}, das höchstwertigste Bit, heißt **MSB** (Most Significant Bit)

Umrechnung: Dualzahlen $\leftrightarrow$ Dezimalzahl

- Dualzahl $\Longrightarrow$ Dezimalzahl
  - Beispiel: $100100100 = 2^8 + 2^5 + 2^2 = 292$
- Dezimalzahl $\Longrightarrow$ Dualzahl

- Vorgehensweise:

  $i \leftarrow 0;$
  $X_0 \leftarrow 0;$
  `WHILE` $k \neq 0$ `DO`
  $X_i \leftarrow k$ `MODULO 2`
  $k \leftarrow k$ `DIV 2`
  $i \leftarrow i + 1$

- Beispiel: 292

k	k MODULO 2	k DIV 2	i
292	$x_0 : 0$	146	0
146	$x_1 : 0$	73	1
73	$x_2 : 1$	36	2
36	$x_3 : 0$	18	3
18	$x_4 : 0$	9	4
9	$x_5 : 1$	4	5
4	$x_6 : 0$	2	6
2	$x_7 : 0$	1	7
1	$x_8 : 1$	0	8

- $292_{10} = 100100100_2$

Oktal- und Hexadezimalzahlen

- Bits werden zur besseren Lesbarkeit zu **Bitgruppen** zusammengefasst
- **Oktaden** (3 Bit) $\Longrightarrow$ Kodierung im **Oktalsystem**
  - Das Oktalsystem ist ein Zahlensystem mit der Basis 8
  - Kennt 8 Ziffern zur Darstellung einer Zahl: $0, 1, 2, 3, 4, 5, 6$ und 7
- **Tetraden** (4 Bit) $\Longrightarrow$ Kodierung im **Hexadezimalsystem**
  - Das Hexadezimalsystem ist ein Zahlensystem mit der Basis 16
  - Kennt 16 Ziffern zur Darstellung einer Zahl: $0, 1, \ldots 8, 9, A, B, C, D, E, F$
  - 1 Byte (8 Bit) kann mit 2 Hexadezimalziffern dargestellt werden

dezimal	binär	oktal	hexadezimal	dezimal	binär	oktal	hexadezimal
0	0000	0	0	8	1000	—	8
1	0001	1	1	9	1001	—	9
2	0010	2	2	10	1010	—	A
3	0011	3	3	11	1011	—	B
4	0100	4	4	12	1100	—	C
5	0101	5	5	13	1101	—	D
6	0110	6	6	14	1110	—	E
7	0111	7	7	15	1111	—	F

- $327_{10} = 101|000|111_2 = 507_8$
- $327_{10} = 1|0100|0111_2 = 147_{16}$

Datei- und Speichergrößen

- Unter der Größe einer Datei versteht man die Anzahl der Bytes dieser Datei
- Da sich die Größenordnungen der allermeisten Dateien im Bereich mehrerer Tausend oder Millionen Bytes befinden, müssen die entsprechenden Längenmaße bekannt sein
- Für Datenspeicher mit binärer Adressierung ergeben sich Speicherkapazitäten von 2^n Byte, also **Zweierpotenzen**

Name (Symbol)	**binär**
Kilobyte (kB)	2^{10} Byte = 1.024 Byte
Megabyte (MB)	2^{20} Byte = 1.048.576 Byte
Gigabyte (GB)	2^{30} Byte = 1.073.741.824 Byte
Terabyte (TB)	2^{40} Byte = 1.099.511.627.776 Byte
Petabyte (PB)	2^{50} Byte = 1.125.899.906.842.624 Byte
Exabyte (EB)	2^{60} Byte = 1.152.921.504.606.846.976 Byte
Zettabyte (ZB)	2^{70} Byte = 1.180.591.620.717.411.303.424 Byte
Yottabyte (YB)	2^{80} Byte = 1.208.925.819.614.629.174.706.176 Byte

- Diese Maßeinheiten haben sich für die Größenangabe von Hauptspeicher und anderen Speichermedien eingebürgert

Datei- und Speichergrößen

- Die Hersteller von Festplatten, CD/DVDs und USB-Speichermedien verwenden für die Berechnung lieber die Dezimal-Präfixe, also den Faktor 10^9 anstatt 2^{30} für Gigabyte und 10^{12} anstatt 2^{40} für Terabyte
  - Darum wird z.B. bei einem DVD-Rohling mit einer angegebenen Kapazität von 4,7 GB in einigen Anwendungen nur eine Kapazität von 4,38 GB angezeigt
    - $10^9 = 1.000.000.000$, $2^{30} = 1.073.741.824$
    - Unterschied: ca. $7,37$ %
  - Ein Rechner mit einer angeblich 1 TB großen Festplatte bekommt nur eine Kapazität von ca. 930 GB angezeigt
    - $10^{12} = 1.000.000.000.000$, $2^{40} = 1.099.511.627.776$
    - Unterschied: ca. $9,95$ %

Kibibyte (KiB), Mebibyte (MiB), Gibibyte (GiB), Tebibyte (TiB), Pebibyte (PiB), Exbibyte (EiB), Zebibyte (ZiB),...

- Die International Electrotechnical Commissio (IEC) schlug 1996 vor, die populären Größenfaktoren, die auf den Zweierpotenzen basieren, mit einem kleinen „i“ zu kennzeichnen
- Die etablierten Bezeichnungen der Maßeinheiten wären dann für die Dezimal-Präfixe reserviert
- Dieser Vorschlag konnte sich bislang nicht durchsetzen

Informationsdarstellung

- Daten sind Folgen von Nullen und Einsen, die beliebige Informationen repräsentierten
- Wie erfolgt aber die Darstellung von Texten und Zahlen durch Daten?
- Um Texte darzustellen, kodiert man die Zeichen des Alphabets (Groß- und Kleinschreibung), die Satzzeichen wie Punkt, Komma und Semikolon, sowie einige Spezialzeichen wie +, %, & und $ in Bitfolgen
- Zudem benötigt man noch Sonderzeichen zur Steuerung wie das Leerzeichen (`SP`) den Wagenrücklauf (`CR`) und das Tabulatorzeichen (`TAB`)
- Die populärste Form der Kodierung ist der **American Standard Code for Information Interchange (ASCII)**

ASCII-Kodierung

- 7-Bit-Zeichenkodierung
  - Jedem Zeichen wird ein Bitmuster aus 7 Bit zugeordnet
  - Es gibt $2^7 = 128$ verschiedene Bitmuster
- Standard seit 1963
  - Letzte Aktualisierung 1968
- Die Zeichenkodierung definiert 128 Zeichen
  - 33 nicht druckbare Zeichen
  - 95 druckbare Zeichen
- Die druckbaren Zeichen sind (beginnend mit dem Leerzeichen):

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

- Das 8. Bit ermöglichte als Paritätsbit die Fehlererkennung
  - Es wurde auf 0 gesetzt, wenn die Anzahl der Einsen an den übrigen 7 Bitpositionen gerade ist und ansonsten auf 1 gesetzt
- Heute wird das 8. Bit fast immer als Erweiterung von ASCII auf einen 8-Bit-Code verwendet

ASCII-Tabelle

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ASCII-Erweiterungen

- Die vergangen Jahre und Jahrzehnte wurde fast immer zur Erweiterung von ASCII ein 8 Bit-Code verwendet
  - Wird jedem Zeichen ein Bitmuster aus 8 Bit zugeordnet, sind $2^8 = 256$ verschiedene Bitmuster verfügbar
- Die Erweiterungen sind mit dem ursprünglichen ASCII weitgehend kompatibel, so dass alle im ASCII definierten Zeichen auch in den verschiedenen Erweiterungen durch die gleichen Bitmuster kodiert werden
- Die Erweiterungen wie z.B. `ISO 8859-1` (ISO Latin 1) enthalten sprachspezifische Zeichen (z.B. Umlaute) und Sonderzeichen, die nicht im lateinischen Grundalphabet enthalten sind

ASCII-Erweiterung ISO 8859-1 (ISO Latin 1)

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Bildquelle: `http://www.wikiservice.at/upload/duser/iso-8859-1.png`

UNICODE

- Zeichenkodierung, die jede menschliche Schriftzeichen oder Textelement aller bekannten Schriftkulturen und Zeichensysteme enthalten soll
- Ziel: Vermeidung der Verwendung unterschiedlicher und inkompatibler Kodierungen in verschiedenen Ländern oder Kulturkreisen
- Es existieren verschiedene UNICODE-Standards (UTF-2, UTF-7, UTF-8, UTF-16)
- Am populärsten ist UTF-8
  - UTF-8 ist eine Mehrbyte-Kodierung
    - Die ersten 128 Zeichen werden mit einem Byte codiert und sind mit der ASCII-Kodierung identisch
    - Die Kodierungen aller anderen Zeichen verwenden zwischen 2 und 6 Byte
  - 1.114.112 Codepunkte (Zeichen) sind möglich
  - Enthält bereits über 100.000 Zeichen und wird ständig erweitert
- Unter der ISO-Norm 10636 wurde UNICODE international standardisiert

Zeichenkodierung bei UTF-8 (UNICODE)

- Jedes mit 0 beginnende Byte ist ein 7-Bit ASCII-Zeichen
- Jedes mit 1 beginnende Byte gehört zu einem aus mehreren Bytes bestehenden UTF-8 Code
  - Besteht ein UTF-8 Code aus $n \geq 2$ Byte, beginnt das erste Byte mit n vielen Einsen und jedes der $n-1$ folgenden Byte mit der Bitfolge 10

Code-Länge	**Bit zur Zeichen-Codierung**	**Format**
1 Byte	7 Bit	`0xxxxxxx`
2 Byte	11 Bit	`110xxxxx 10xxxxxx`
3 Byte	16 Bit	`1110xxxx 10xxxxxx 10xxxxxx`
4 Byte	21 Bit	`11110xxx 10xxxxxx 10xxxxxx 10xxxxxx`
5 Byte	26 Bit	`111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx`
6 Byte	31 Bit	`1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx`

Darstellung von Zeichenketten

- Um einen fortlaufenden Text zu kodieren, fügt man die einzelnen Zeichen zu einer Zeichenkette (*String*) aneinander
- Der Text `Vorlesung Netzwerke.` wird zur folgenden Zeichenfolge
  `V, o, r, l, e, s, u, n, g,  , N, e, t, z, w, e, r, k, e, .`
- Alle Zeichen (auch das Leerzeichen) werden durch die Nummer in der ASCII-Tabelle ersetzt
  `086 111 114 108 101 115 117 110 103 032`
  `078 101 116 122 119 101 114 107 101 046`
- Alternativ kann man die ASCII-Nummern auch hexadezimal schreiben
  `56 6F 72 6C 65 73 75 6E 67 20 4E 65 74 7A 77 65 72 6B 65 2E`
- Damit kann man die Repräsentation als Bitfolge ermitteln
  `01010110 01101111 01110010 01101100 01100101`
  `01110011 01110101 01101110 01100111 00100000`
  `01001110 01100101 01110100 01111010 01110111`
  `01100101 01110010 01101011 01100101 00101110`

Nächste Vorlesung

Nächste Vorlesung:

20.10.2011