

Standardisierte Codes

Der ASCII-Code

Damit Daten zwischen verschiedenen Programmen und zwischen verschiedenen Rechnern problemlos ausgetauscht werden können, hat man sich bereits 1963 auf einen Standard für die binäre Codierung gängiger Zeichen, wie z. B. der lateinischen Klein- und Großbuchstaben geeinigt. Dieser Code nennt sich **ASCII-Code**. Das steht für *American Standard Code for Information Interchange* und bedeutet auf deutsch *Amerikanischer Standard-Code für den Informationsaustausch*. Tabelle 1 zeigt die Codierung der Großbuchstaben und des Leerzeichens gemäß des ASCII-Codes.

Buchstabe	ASCII-Code	Buchstabe	ASCII-Code	Buchstabe	ASCII-Code
A	100 0001	J	100 1010	S	101 0011
B	100 0010	K	100 1011	T	101 0100
C	100 0011	L	100 1100	U	101 0101
D	100 0100	M	100 1101	V	101 0110
E	100 0101	N	100 1110	W	101 0111
F	100 0110	O	100 1111	X	101 1000
G	100 0111	P	101 0000	Y	101 1001
H	100 1000	Q	101 0001	Z	101 1010
I	100 1001	R	101 0010	Leerzeichen	010 0000

Tabelle 1: ASCII-Code für die Großbuchstaben und das Leerzeichen

Aufgabe 1:

- Beschreibe den Aufbau des ASCII-Codes.
- Vergleiche den Aufbau des ASCII-Codes mit euren eigenen Codetabellen. Welche Gemeinsamkeiten und welche Unterschiede stellt ihr fest?

Der ASCII-Code ist ein binärer Code, da jede Stelle in einem Codewort nur die Werte 0 oder 1 annehmen kann. Eine solche Stelle bezeichnen Informatiker als Bit. Die Codierung eines Zeichens im ASCII-Code besteht somit aus 7 Bit.

Aufgabe 2:

- Berechne, wie viele Zeichen sich mit 7 Bit codieren lassen.
- Eine vollständige Codetabelle des ASCII-Codes findest du in der Datei *ASCII-Code.pdf*. Schau in der Codetabelle nach, welche der Zeichen aus Tabelle 2 im ASCII-Code berücksichtigt wurden. Wenn du eine binäre Codierung findest, trage sie in die Tabelle 2 ein.

Zeichen	ASCII-Code	Zeichen	ASCII-Code	Zeichen	ASCII-Code
!		ß		@	
Ä		€		\$	

Tabelle 2: Für welche Zeichen gibt es einen ASCII-Code?

- Stelle eine Vermutung auf, warum der ASCII-Code für einige der Zeichen in Tabelle 2 keine Codierung vorsieht.

Erweiterungen des ASCII-Codes

Fast jede Sprache, die das lateinische Alphabet verwendet, verfügt über einige Sonderzeichen. Im Deutschen verwenden wir z. B. die Umlaute Ä, Ö und Ü. Andere Sprachen haben ein eigenes Alphabet. Um auch sprachspezifische Zeichen im Umgang mit Computern verwenden zu können, gibt es für verschiedene Sprachfamilien Erweiterungen des ASCII-Codes, die 8 statt 7 Bits verwenden. Der binären Codierung der Zeichen, die im ASCII-Code bereits enthalten sind, wird dabei eine 0 vorangestellt. Für die westeuropäischen Sprachen sind aus dem ASCII-Code zwei verschiedene 8-Bit-Codes entstanden. Der eine ist unter der Bezeichnung *ISO 8859-1* bzw. *Latin-1* zu finden ist. Der andere unter der Bezeichnung *ISO 8859-9* bzw. *Latin-9*. Ein weiterer 8-Bit-Code, der z. B. die kyrillischen Zeichen¹ abdeckt, trägt die Bezeichnung *ISO 8859-5* bzw. *Latin-5*.

Aufgabe 3:

- Berechne wie viele zusätzliche Zeichen sich mit einem weiteren Bit codieren lassen.
- Welche Probleme könnten unter Verwendung der verschiedenen Erweiterungen des ASCII-Codes auftreten, wenn ein Text sowohl Umlaute der deutschen Sprache als auch kyrillische Schriftzeichen enthalten soll?

Um den Datenaustausch weltweit zu vereinfachen, wurde in den 1990er Jahren mit der Entwicklung des Unicodes begonnen. Der **Unicode** soll für jedes weltweit bekannte Schriftzeichen eine einzigartige binäre Codierung enthalten. Er wird daher ständig erweitert. Für die Codierung der Zeichen wurden zunächst 16 Bit verwendet. Damit lassen sich $2^{16} = 65\,536$ Zeichen codieren. Da man jedoch noch mehr Zeichen codieren wollte, hat man die Anzahl der Bits, mit denen ein Zeichen codiert werden kann, noch einmal verdoppelt. Damit lassen sich theoretisch über 4 Milliarden Zeichen unterscheiden.

Interpretation des ASCII-Codes als Zahlen

Wenn man sich Codetabellen des ASCII-Codes anschaut, enthalten diese häufig nicht nur einen binären Code, sondern auch einen dezimalen Zahlencode. So wird der Großbuchstabe A beispielsweise mit der Zahl 65 codiert. Das hängt damit zusammen, dass die Bitfolge aus Nullen und Einsen, die den Zeichen im ASCII-Code zugeordnet wird, nicht völlig willkürlich ist. Die Bitfolgen können als Zahlen im binären Zahlensystem interpretiert werden. Um diesen Zusammenhang zu verstehen, müssen wir uns unser dezimales und das binäre Zahlensystem etwas genauer anschauen.

Interpretation von Zahlen im Binärsystem

Wir sind es gewohnt im **Dezimalsystem** (Zehnersystem) zu rechnen. Dabei hängt der Wert einer Ziffer davon ab, an welcher Stelle die Ziffer in einer Zahl steht. In der Grundschule haben wir gelernt, dass die letzte Ziffer ganz rechts die Einer sind, dann kommen links daneben die Zehner, die Hunderter usw. Die Zahl 3278 wurde in eine solche Stellenwerttabelle eingetragen. Der Wert der Zahl 3278 ergibt sich also daraus, dass wir sie so interpretieren, dass wir 3 Tausender, 2 Hunderter, 7 Zehner und 8 Einer addieren: $3 \cdot 1000 + 2 \cdot 100 + 7 \cdot 10 + 8 \cdot 1 = 3000 + 200 + 70 + 8 = 3278$

Stellen	T(ausender)	H(underter)	Z(ehner)	E(einer)
Wert	$1000 = 10^3$	$100 = 10^2$	$10 = 10^1$	$1 = 10^0$
Beispiel	3	2	7	8

¹ Eine Sprache, die kyrillische Zeichen verwendet, ist z. B. Russisch.

Da uns die Ziffern 0 bis 9 zur Verfügung stehen, ergibt sich für die 10 ein Übertrag zur nächsten Stelle. Die Werte der einzelnen Stellen ergeben sich aus den 10er Potenzen.

Im binären Zahlensystem stehen uns nur die Ziffern 0 und 1 zur Verfügung. Ein Übertrag zur nächsten Stelle ergibt sich daher schon für die Zahl 2. Die Werte der einzelnen Stellen ergeben sich aus den 2er Potenzen. Wenn wir die 2er Potenzen in unserem dezimalen Zahlensystem darstellen, können wir eine binäre Zahl in eine dezimale Zahl umrechnen: $1101 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 8 + 4 + 1 = 13$

Stellen	Achter	Vierer	Zweier	Einer
Wert	$8 = 2^3$	$4 = 2^2$	$2 = 2^1$	$1 = 2^0$
Beispiel	1	1	0	1

Damit man weiß, dass ob es sich um eine Zahl im Binärsystem oder im Dezimalsystem handelt, schreibt man eine kleine 2 bzw. 10 als Index an die Zahl: $(1101)_2 = (13)_{10}$

Aufgabe 4:

- Trage den ASCII-Code für die Großbuchstaben C, K und Z und für das Leerzeichen in eine binäre Stellenwerttabelle ein. Berechne anschließend welcher Zahl der Code im Dezimalsystem entspricht, wenn man ihn als binäre Zahl interpretiert.
- Wie kann man den 7-Bit ASCII-Code zu einem 8-Bit Code erweitern, ohne den Wert der Zahl zu verändern?
- Auch im Unicode soll sich der Zahlenwert der binären Codierung der Zeichen, die bereits im ASCII-Code enthalten sind, nicht verändern. Gib an, wie der 16-Bit Unicode des Großbuchstabens A lauten muss.

Dezimalzahlen als Binärzahlen darstellen

In manchen Codetabellen für den ASCII-Code werden nur die Dezimalzahlen angegeben. Es ist auch möglich zu einer Dezimalzahl die entsprechende Binärzahl zu bestimmen, um so den passenden binären Code herauszufinden. Dafür gibt es zwei Möglichkeiten.

Verfahren 1: Dezimalzahl in Summe aus 2er-Potenzen zerlegen

Bei dem ersten Verfahren suchen wir zunächst nach der größten Zweierpotenz, die in unsere Dezimalzahl passt. Bei der 78 (ASCII-Code für das große N) wäre das z. B. $2^6 = 64$. Somit wissen wir, dass die Ziffer an der Position für 2^6 eine 1 sein muss und dies die vorderste Stelle ist:

Zer-Potenz	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	64	32	16	8	4	2	1
Beispiel	1						

Außerdem erhalten wir einen Rest von $78 - 64 = 14$, der mithilfe der restlichen Stellen dargestellt werden muss. Wir füllen nun die Stellenwerttabelle beginnend bei der 64 von links nach rechts auf. Da jede Stelle nur die Ziffer 1 oder 0 haben kann, müssen wir für jede Stelle entscheiden, ob ihr Wert in den verbleibenden Rest passt oder nicht.

32 und 16 passen nicht in den Rest 14, daher tragen wir bei beiden Stellen eine 0 ein:

Zer-Potenz	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	64	32	16	8	4	2	1
Beispiel	1	0	0				

Die 8 passt in den Rest 14, daher tragen wir hier eine 1 ein und erhalten den neuen Rest $14 - 8 = 6$:

Zer-Potenz	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	64	32	16	8	4	2	1
Beispiel	1	0	0	1			

Die 4 passt in den Rest 6, daher tragen wir hier eine 1 ein und erhalten den neuen Rest $6 - 4 = 2$:

Zer-Potenz	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	64	32	16	8	4	2	1
Beispiel	1	0	0	1	1		

Die 2 passt in den Rest 2, daher tragen wir hier eine 1 ein und erhalten den neuen Rest $2 - 2 = 0$:

Zer-Potenz	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	64	32	16	8	4	2	1
Beispiel	1	0	0	1	1	1	

Anhand des Restes 0 erkennen wir, dass wir die Zahl vollständig in Summanden zerlegt haben, die den Stellen im Binärsystem entsprechen: $78 = 64 + 8 + 4 + 2$

Für die verbleibenden Stellen, hier die letzte Stelle, können wir daher eine 0 eintragen:

Zer-Potenz	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	64	32	16	8	4	2	1
Beispiel	1	0	0	1	1	1	0

Die Zahl 78 entspricht im Binärsystem also der Zahl 100 1110: $(78)_{10} = (1001110)_2$

Verfahren 2: Fortlaufende Division durch 2

Bei dem zweiten Verfahren teilen wir die Zahl aus dem Dezimalsystem durch 2 und schreiben uns sowohl das ganzzahlige Ergebnis als auch den Rest auf. Das ganzzahlige Ergebnis teilen wir wieder durch 2. Dies setzen wir fort, bis bei der Division 0 herauskommt. Aus den Resten der Divisionen setzt sich nun unsere Binärzahl zusammen, wobei der Rest der **letzten Division** die **erste Stelle** ganz links unserer Binärzahl ist.

Beispiel: $78 : 2 = 39 \text{ R } 0$

$39 : 2 = 19 \text{ R } 1$

$19 : 2 = 9 \text{ R } 1$

$9 : 2 = 4 \text{ R } 1$

$4 : 2 = 2 \text{ R } 0$

$2 : 2 = 1 \text{ R } 0$

$1 : 2 = 0 \text{ R } 1$

$(78)_{10} = (1001110)_2$

Aufgabe 5:

- Bestimme mit einem der beiden Verfahren die Darstellung der Zahlen 99 und 50. Welche Zeichen codieren diese Zahlen gemäß des ASCII-Codes?
- Justus behauptet, ich muss mir nur merken, dass der Großbuchstabe A im ASCII-Code mit der Zahl 65 und der Kleinbuchstabe a mit der Zahl 97 codiert wird, dann kann ich mir die binäre Codierung aller Groß- und Kleinbuchstaben herleiten. Nimm Stellung.



Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

