Das Binärsystem

Jacques Mock Schindler

26.02.2025

Wir sind es gehwont uns im Dezimalsystem zu bewegen. Das bedeutet, dass wir Zahlen in der Basis 10 darstellen. Wir verwenden dazu 10 verschiedene Ziffern, nämlich die Ziffern 0 bis 9.
Um Zahlen darzustellen, welche grösser sind als 9 darzustellen, verwednen wir die Zehnerpotenzen. Das heisst, dass die Ziffern von rechts nach links gelesen, die Zehnerpotenzen von 0 an aufsteigend sind. Die Zahl 123 entspricht also der Rechnung $1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$.

Das Binärsystem funktioniert nach dem gleichen Prinzip, nur dass wir nur die Ziffern 0 und 1 verwenden. Das bedeutet, dass wir Zahlen in der Basis 2 darstellen. Sobald eine Zahl grösser als 1 dargestellt werden soll, schreiben wir die Zahl als Summe von Zweierpotenzen. Die Zahl 101 entspricht also der Rechnung $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$.

Umrechnung von Dezimalzahlen in Binärzahlen

Um eine Dezimalzahl in eine Binärzahl umzurechnen, gibt es eine systematische Methode, die auf wiederholter Division durch 2 basiert:

1. Teile die Dezimalzahl durch 2 und notiere den Rest (0 oder 1).
2. Teile den Quotienten (Ergebnis der Division) erneut durch 2 und notiere wieder den Rest.
3. Fahre damit fort, bis der Quotient 0 ist.
4. Die Binärzahl wird nun gebildet, indem man die Reste von unten nach oben (vom letzten zum ersten) liest.

Beispiel: Umrechnung von 42 ins Binärsystem

Lassen Sie uns die Dezimalzahl 42 in eine Binärzahl umrechnen:

Division	Rechnung	Quotient	Rest
42 ÷ 2	42 = 21 · 2 + 0	21	0
21 ÷ 2	21 = 10 · 2 + 1	10	1

Division	Rechnung	Quotient	Rest
10 ÷ 2	10 = 5 · 2 + 0	5	0
5 ÷ 2	5 = 2 · 2 + 1	2	1
2 ÷ 2	2 = 1 · 2 + 0	1	0
1 ÷ 2	1 = 0 · 2 + 1	0	1

Nun lesen wir die Reste von unten nach oben: 101010

Also ist 42 im Dezimalsystem gleich 101010 im Binärsystem.

Überprüfung:

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 32 + 0 + 8 + 0 + 2 + 0 = 42$$

Eine alternative Methode zur Umrechnung ist die Verwendung der Zweierpotenzen. Man zerlegt die Dezimalzahl in eine Summe von Zweierpotenzen und schreibt dann eine 1 an den Stellen der verwendeten Potenzen und eine 0 an allen anderen Stellen.

Umrechnung von Binärzahlen in Dezimalzahlen

Die Umrechnung von Binärzahlen in Dezimalzahlen ist vergleichsweise einfach und basiert auf der Berechnung der Stellenwerte im Binärsystem.

Methode:

1. Identifiziere jede Stelle in der Binärzahl und ihre Position (von rechts nach links beginnend bei 0).
2. Multipliziere jede Binärziffer (0 oder 1) mit dem Wert von 2 hoch der Stellenposition.
3. Addiere alle resultierenden Werte zusammen.

Beispiel: Umrechnung von 10110 ins Dezimalsystem

Lassen Sie uns die Binärzahl 10110 in eine Dezimalzahl umrechnen:

Position (von rechts)	4	3	2	1	0
Binäre Ziffer	1	0	1	1	0
Berechnung	$1 \cdot 2^4$	$0 \cdot 2^3$	$1 \cdot 2^2$	$1 \cdot 2^1$	$0 \cdot 2^0$
Dezimalwert	16	0	4	2	0

Nun addieren wir alle Dezimalwerte: 16 + 0 + 4 + 2 + 0 = 22

Also ist 10110□ im Binärsystem gleich 22 im Dezimalsystem.

Formel:

Für eine Binärzahl mit n Stellen (b□, b□, …, b□□□) gilt:

Dezimalzahl = b□ · 2^0 + b□ · 2^1 + b□ · 2^2 + … + b□□□ · $2^{n□1}$

wobei b□ die erste Stelle von rechts ist.

Praktischer Trick:

Bei der Umrechnung kann man auch eine Tabelle mit den Stellenwerten erstellen:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Man schreibt die Binärzahl unter die Tabelle und addiert nur die Werte, an deren Position eine 1 steht.

Binärcodierung von Strings

In der Informatik werden Textzeichen (Strings) durch binäre Codes repräsentiert. Bei Unicode-Zeichen wird eine fortschrittliche Codierung verwendet, um sowohl einfache als auch komplexe Schriftzeichen aus allen Sprachen der Welt darstellen zu können.

Unicode: Ein universeller Zeichencode

Unicode ist ein internationaler Standard zur Darstellung von Textzeichen aller Schriftsysteme. Jedes Zeichen erhält eine eindeutige Nummer (Codepoint), z.B. hat der Buchstabe “A” den Codepoint U+0041 (dezimal: 65).

UTF-8: Die häufigste Unicode-Codierung

UTF-8 ist die am weitesten verbreitete Codierungsform für Unicode. Sie hat folgende Eigenschaften:

1. **Variable Länge**: Ein Zeichen wird mit 1 bis 4 Bytes codiert
2. **Abwärtskompatibilität**: ASCII-Zeichen (0-127) werden mit nur 1 Byte dargestellt
3. **Selbstsynchronisierend**: Der Anfang eines Zeichens ist eindeutig erkennbar

UTF-8 Codierungsschema:

Anzahl Bytes	Binärmuster	Codepoint-Bereich
1	`0xxxxxxx`	U+0000 bis U+007F
2	`110xxxxx 10xxxxxx`	U+0080 bis U+07FF
3	`1110xxxx 10xxxxxx 10xxxxxx`	U+0800 bis U+FFFF
4	`11110xxx 10xxxxxx 10xxxxxx 10xxxxxx`	U+10000 bis U+10FFFF

Die mit `x` markierten Stellen werden mit den Bits des Unicode-Codepoints gefüllt.

Beispiel: Codierung deutscher Umlaute

Nehmen wir als Beispiel den Buchstaben “ü” (U+00FC):

1. Der Codepoint von “ü” ist U+00FC (dezimal: 252)
2. Als Binärzahl: 11111100
3. Da der Wert > 127 ist, benötigen wir 2 Bytes
4. Nach dem 2-Byte-Schema: `110xxxxx 10xxxxxx`
5. Wir füllen den Unicode-Wert ein: `110(00000) 10(111100)` → `11000011 10111100`
6. Somit wird “ü” als die zwei Bytes `11000011 10111100` codiert

Weiteres Beispiel: Das Pi-Symbol π

1. Der Codepoint von “π” ist U+03C0 (dezimal: 960)
2. Als Binärzahl: 0000001111000000
3. Da der Wert > 127 und < 2048 ist, benötigen wir 2 Bytes
4. Nach dem 2-Byte-Schema: `110xxxxx 10xxxxxx`
5. Wir füllen den Unicode-Wert ein: `110(00111) 10(100000)` → `11000111 10100000`
6. Somit wird “π” als die zwei Bytes `11000111 10100000` codiert (hexadezimal: CE A0)

Beispiel: ‘Informatik ist interessant.’

Lassen Sie uns den Satz “Informatik ist interessant.” in die binäre UTF-8-Kodierung umwandeln:

Zeichen	Unicode-Codepoint	Dezimal	Binär	UTF-8 Kodierung
I	U+0049	73	1001001	01001001
n	U+006E	110	1101110	01101110

Zeichen	Unicode-Codepoint	Dezimal	Binär	UTF-8 Kodierung
f	U+0066	102	1100110	01100110
o	U+006F	111	1101111	01101111
r	U+0072	114	1110010	01110010
m	U+006D	109	1101101	01101101
a	U+0061	97	1100001	01100001
t	U+0074	116	1110100	01110100
i	U+0069	105	1101001	01101001
k	U+006B	107	1101011	01101011
□	U+0020	32	100000	00100000
i	U+0069	105	1101001	01101001
s	U+0073	115	1110011	01110011
t	U+0074	116	1110100	01110100
□	U+0020	32	100000	00100000
i	U+0069	105	1101001	01101001
n	U+006E	110	1101110	01101110
t	U+0074	116	1110100	01110100
e	U+0065	101	1100101	01100101
r	U+0072	114	1110010	01110010
e	U+0065	101	1100101	01100101
s	U+0073	115	1110011	01110011
s	U+0073	115	1110011	01110011
a	U+0061	97	1100001	01100001
n	U+006E	110	1101110	01101110
t	U+0074	116	1110100	01110100
.	U+002E	46	101110	00101110

Da alle Zeichen in diesem Beispiel im ASCII-Bereich liegen (0-127), wird jedes Zeichen mit genau einem Byte codiert. Der vollständige Text benötigt also 27 Bytes im UTF-8 Format.

Die komplette binäre Darstellung des Textes ist:

```
01001001 01101110 01100110 01101111 01110010 01101101 01100001 01110100
01101001 01101011 00100000 01101001 01110011 01110100 00100000 01101001
01101110 01110100 01100101 01110010 01100101 01110011 01110011 01100001
01101110 01110100 00101110
```

In hexadezimaler Schreibweise:

```
49 6E 66 6F 72 6D 61 74 69 6B 20 69 73 74 20 69 6E 74 65 72 65 73 73 61 6E 74 2E
```

Dieser String kann direkt in einem Computer gespeichert und verarbeitet werden. In Textdateien, bei der Übertragung im Internet oder in Datenbanken sind Strings immer in solchen binären Formaten gespeichert.

Vorteile der Unicode-Codierung:

1. **Universalität**: Alle Schriftsysteme und Sonderzeichen können dargestellt werden
2. **Effizienz**: Häufig verwendete Zeichen benötigen weniger Speicherplatz
3. **Kompatibilität**: Rückwärtskompatibel mit ASCII, was die Integration erleichtert

Die Umwandlung zwischen Textzeichen und ihrer binären Repräsentation wird in Computersystemen automatisch durch Codierungs- und Decodierungsprozesse abgewickelt, sodass Benutzer sich in der Regel nicht mit den Details befassen müssen.