

Grundlagen der Betriebssysteme

Blatt 01

Gruppe 055

Marco Deuscher
Ibrahim Hasan

Mai 2019

1 Festkomma Darstellung

(a)

$$7,75_{10} = 4 + 2 + 1 + \frac{1}{2} + \frac{1}{4} = 00111110_2$$

Da die Zahl aus zweier Potenzen zusammengesetzt werden kann, ist sie exakt darstellbar.

(b)

$$2,71_{10} \approx 2 + \frac{1}{2} + \frac{1}{4} = 00010110_2 = 2.75_{10}$$

Die Zahl ist nicht exakt darstellbar. $|Z_{original} - Z_{Umwandlung}| = 0,04$

(c)

$$5,375_{10} = 4 + 1 + \frac{1}{4} + \frac{1}{8} = 00101011_2$$

Da die Zahl aus zweier Potenzen zusammengesetzt werden kann, ist sie exakt darstellbar.

(d)

$$9,12_{10} \approx 01001001_2 = 9,125_{10}$$

Die Zahl ist nicht exakt darstellbar. $|Z_{original} - Z_{Umwandlung}| = 0,005$

2 Gleitkomma Darstellung

Ergebnisse werden im folgenden in drei Gruppen unterteilt

1. Vorzeichen

2. Exponent

3. Mantisse

Außerdem ist der Bias mit $B = 127$ gegeben. Es wird ein Vorzeichenbit verwendet, der Exponent hat eine Länge von 8bit und die Mantisse eine Länge von 23bit.

(a)

$$x = (-1)^v \cdot m \cdot 2^{e-B} \Rightarrow m = \frac{x}{2^{e-B}} = 1,109375$$

Der Exponent ist gegeben durch $127 + 4 = 131$ was in binär Darstellung wiederum 10000011 entspricht. Durch Umwandeln Nachkommastellen der Mantisse in binäre Darstellung erhält man dann die folgende Darstellung

$$0 \quad 10000011 \quad 00011100 \dots 0$$

(b) Analoges Vorgehen wie in der (a) liefert dann

$$m = \frac{x}{2} = 1,8125$$

Für den Exponenten erhält man $127 + 1 = 128$. Durch Umwandeln in die binäre Darstellung erhält man dann

$$0 \quad 10000000 \quad 110100 \dots 0$$

3 Gleitkomma Operationen

Noch zu bearbeiten

(a)

Addition:

$$\begin{aligned} & 2^{128-127} \cdot 1,0010111 + 2^{128-127} \cdot 1,1001011 \\ &= 2^{128-127} (1,0010111 + 1,1001011) \\ &= 2^{129-127} \cdot 1,01100010 \end{aligned}$$

Multiplikation

$$\begin{aligned} & 2^{128-127} \cdot 1,0010111 \cdot 2^{128-127} \cdot 1,1001011 \\ &= 2^{129-127} \cdot (1,0010111 \cdot 1,1001011) \\ &= 2^{129-127} \cdot 1,11011110111101 \end{aligned}$$

(b)

Addition Nutze zweier Komplement um die beiden Zahlen zu addieren

$$\begin{aligned} & (-1)^1 \cdot 2^{131-127} \cdot 1,1011 + 2^{127-127} \cdot 1,000011 \\ &= 2^{131-127} \cdot (-1,1011 + 0,0001000011) \\ &= (-1)^1 \cdot 2^{131-127} \cdot 1,1001111101 \end{aligned}$$

Multiplikation

$$\begin{aligned} & (-1)^1 \cdot 2^{131-127} \cdot 1,1011 \cdot 2^{127-127} \cdot 1,000011 \\ &= (-1)^1 \cdot 2^{131-127} \cdot (1,1011 \cdot 1,000011) \\ &= (-1)^1 \cdot 2^{131-127} \cdot 1,1100010001 \end{aligned}$$

(c)

Addition

$$\begin{aligned} & (-1)^1 \cdot 2^{128-127} \cdot 1,1010011 + (-1)^1 \cdot 2^{133-127} \cdot 1,0101011 \\ &= (-1)^1 \cdot 2^{133-127} \cdot (0,000011010011 + 1,0101011) \\ &= (-1)^1 \cdot 2^{133-127} \cdot 1,011000110011 \end{aligned}$$

Multiplikation

$$\begin{aligned} & (-1)^1 \cdot 2^{128-127} \cdot (-1)^1 \cdot 2^{133-127} \cdot 1,1010011 \cdot 1,0101011 \\ &= 2^{135-127} \cdot 1,000110011110001 \end{aligned}$$

4 UTF8 Darstellung

Hierbei ist die **Startsequenz**, **Data**, **Beginn eines neues Bytes** jeweils markiert.

(a)

$$\begin{aligned} 202e &= 0010\ 0000\ 0010\ 1110 \\ \textcolor{red}{1110}\ \textcolor{blue}{0010}\ \textcolor{green}{1000}\ 0000\ \textcolor{blue}{1010}\ 1110 \end{aligned}$$

(b)

$$\textcolor{red}{11110}\ 000\ \textcolor{green}{10}\ \textcolor{blue}{011111}\ \textcolor{green}{10}\ \textcolor{blue}{011000}\ \textcolor{green}{10}\ 001000$$

Wandelt man die **cyan** markierten Daten in Hex. Darstellung um, erhält man den Unicode Character U+1F608.

5 Bitinterpretation

(a) Umwandeln von `0x447b7d00` in die binäre Darstellung. Hierbei markiert sind **Vorzeichen**, **Exponent**, **Mantisse**.

0100 0100 **0111 1011 0111 1101 0000 0000**

Vorzeichen lässt sich direkt ablesen \rightarrow Zahl ist positiv

Exponent ist gegeben durch `0b10001000` $= 128 + 8 = 136$. Mit dem Bias ergibt sich dann $e = 9$.

Für die Mantisse erhält man durch die Stufenzahlendarstellung den folgenden Wert

$$m_2 = 1111\ 0110\ 1111\ 1010_2 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{64} + \frac{1}{128} + \frac{1}{512} + \frac{1}{1024} + \frac{1}{2048} + \frac{1}{4096} + \frac{1}{8192} + \frac{1}{32768} = 0.9647521973_{10}$$

Dann erhält man für die Gleitkommazahl den folgenden Wert

$$z_{IEEE754} = (-1)^0 \cdot 1.9647521973_{10} * 2^{136-127} = 1005,953125_{10}$$

(b) Die hexadezimal Darstellung der ersten 16bit Integer ist `0x447b`. Umwandlung in binäre liefert dann `0b0100 0100 0111 1011`. Mit dem Stufenzahlenverfahren erhält man dann einen Wert von $z_{16bit,1} = 17531$.

Für die zweite Zahl ergibt sich dann `0x7d00` $= 0b0111\ 1101\ 0000\ 0000 = 32000_{10}$

(c) Interpretiert man `00447b7d00` als ASCII bzw. UTF-8 erhält man die folgende Darstellung.

In ASCII entspricht `0x44` = `D`, `0x7b` = `{`, `0x7d` = `}` und `0x00` = `\0`.

Interpretiert man `00447b7d00` in UTF-8 erhält man das gleiche Ergebnis wie in ASCII, da UTF-8 die ersten 128 Zeichen aus dem ASCII-Code übernommen hat.