

Übungen zu Systemnahe Programmierung in C

Abschnitt 3.2: Bitoperationen

11.05.2020

Tim Rheinfels

Benedict Herzog

Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT



- Zahlen können in unterschiedlichen Basen dargestellt werden
 - ⇒ Üblich: dezimal (10), hexadezimal (16), oktal (8) und binär (2)
- Nomenklatur:
 - Bits: Ziffern von Binärzahlen
 - Nibbles: Gruppen von 4 Bits
 - Bytes: Gruppen von 8 Bits

- Bitoperation: Bitweise logische Verknüpfung
- Mögliche Operationen:

~	
0	1
1	0

nicht

&	0	1
0	0	0
1	0	1

und

	0	1
0	0	1
1	1	1

oder

^	0	1
0	0	1
1	1	0

exklusives
oder

- Bitoperation: Bitweise logische Verknüpfung
- Mögliche Operationen:

~	
0	1
1	0

nicht

&	0	1
0	0	0
1	0	1

und

	0	1
0	0	1
1	1	1

oder

^	0	1
0	0	1
1	1	0

exklusives
oder

- Beispiel:

$$\begin{array}{r} \sim 1001_2 \\ \hline 0110_2 \end{array}$$

$$\begin{array}{r} 1100_2 \\ \& 1001_2 \\ \hline 1000_2 \end{array}$$

$$\begin{array}{r} 1100_2 \\ | 1001_2 \\ \hline 1101_2 \end{array}$$

$$\begin{array}{r} 1100_2 \\ \wedge 1001_2 \\ \hline 0101_2 \end{array}$$

■ Beispiel:

```
uint8_t x = 0x9d;
```

1	0	0	1	1	1	0	1
---	---	---	---	---	---	---	---

```
x = x << 2;
```

0	1	1	1	0	1	0	0
---	---	---	---	---	---	---	---

```
x = x >> 2;
```

0	0	0	1	1	1	0	1
---	---	---	---	---	---	---	---

■ Setzen von Bits:

```
(1 << 0)
```

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

```
(1 << 3)
```

0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

```
(1 << 3) | (1 << 0)
```

0	0	0	0	1	0	0	1
---	---	---	---	---	---	---	---

■ Achtung:

Bei signed-Variablen ist das Verhalten des >>-Operators nicht vollständig definiert. In der Regel werden bei negativen Werten 1er geshiftet.